

AI Literacy und Computational Thinking im Mathematikunterricht – Eine empirische Untersuchung einer Projektarbeit am Beispiel Q-Learning

ALISSA FOCK, WÜRZBURG; NORBERT NOSTER, WÜRZBURG & HANS-STEFAN SILLER, WÜRZBURG

Zusammenfassung: Im Beitrag wird untersucht, wie KI-Algorithmen – im Kontext von AI Literacy und Computational Thinking – aus mathematischer Perspektive mithilfe fiktiver Settings thematisiert werden können. In der Untersuchung wurden Schüler:innen der Oberstufe bei der Erschließung des Q-Learning-Algorithmus videographiert und die Transkripte mittels qualitativer Inhaltsanalyse analysiert. Die Ergebnisse weisen darauf hin, dass Lernende das fiktive Setting zielführend nutzen und dass durch die Erarbeitung Komponenten des Computational Thinking – insbesondere Abstraktionsprozesse – angeregt werden, was auf Potenziale von KI für die mathematische Bildung über AI Literacy hinaus hindeutet.

Abstract: The article examines how AI algorithms can be addressed from a mathematical perspective with the help of fictitious settings, in the context of AI Literacy and Computational Thinking. In the study, high school students were videotaped as they explored the Q-learning algorithm, and the transcripts were analyzed using qualitative content analysis. The results indicate that learners use the setting successfully and that components of computational thinking – especially abstraction processes – are stimulated during the work sessions. This points to the potential of AI topics for mathematics education beyond AI Literacy.

1. Einleitung

Die zunehmende Allgegenwärtigkeit von KI-Technologien im Alltag hat in den vergangenen Jahren vermehrt die Forderung aufkommen lassen, KI und ihre Nutzung in schulische Bildungsprozesse zu integrieren (z. B. KMK, 2024). Insbesondere soll die Förderung von AI Literacy fokussiert werden (Tan & Tang, 2025). KI-Systeme basieren häufig auf mathematischen Modellen und Algorithmen, sodass der Betrachtung von KI auch im Mathematikunterricht eine wachsende Bedeutung zukommen muss. Wird dabei insbesondere der Umgang mit (KI-)Algorithmen fokussiert, zeigt sich, dass auch Computational Thinking im Zusammenhang mit AI Literacy einen Stellenwert einnimmt (Yang et al., 2025).

Im vorliegenden Beitrag wird daher ein didaktischer Ansatz vorgestellt und empirisch evaluiert, der es Schüler:innen ermöglicht, einen Algorithmus aus

dem Themenfeld der KI aus mathematischer Perspektive zu erkunden. Im Zentrum steht hierbei der Q-Learning-Algorithmus als realer KI-Algorithmus, welcher mit Blick auf die mathematische Komplexität für die Thematisierung mit Schüler:innen angemessen erscheint und sich zugleich mit deren Lebens- und Erfahrungswelt verknüpfen lässt. Hierbei stellt sich zum einen die Frage, inwieweit diese Verknüpfung für die Erschließung des Algorithmus zuträglich ist. Zum anderen ergibt sich die Frage nach der Beziehung zu Computational Thinking als Konstrukt, das sich auf den Umgang mit Algorithmen im Allgemeinen und somit möglicherweise auch auf KI bezieht. Im Beitrag werden beide Fragestellungen empirisch beleuchtet und mit Blick auf die Relevanz für die Ausbildung von AI Literacy im Mathematikunterricht eingeordnet.

2. Theoretischer Hintergrund

Die Konzepte der AI Literacy und des Computational Thinking einerseits, sowie der fachlich-mathematische Hintergrund andererseits dienen als Grundlage für die Aufbereitung einer Lerngelegenheit zu Q-Learning – einem KI-Algorithmus – und der Analyse der Implementierung sowie Vorgehensweise von Lernenden bei der Erarbeitung des Algorithmus.

2.1 AI Literacy

Die vielfältige Verwendung des Begriffs AI Literacy in unterschiedlichen Disziplinen und mit unterschiedlichen Schwerpunkten lässt eine einheitliche Definition des Konzepts kaum möglich erscheinen (Almatrafi et al., 2024). In einem vielfach zitierten Beitrag zu AI Literacy definieren Long und Magerko (2020) AI Literacy als eine Sammlung von Kompetenzen, die es dem Einzelnen ermöglichen, KI-Technologien kritisch zu bewerten, effektiv mit KI zu „kommunizieren“ und „zusammenzuarbeiten“ sowie KI als Werkzeug online, zu Hause und am Arbeitsplatz einzusetzen. Die genauen Fähigkeiten bzw. Komponenten die unter AI Literacy gefasst werden, variieren zwischen verschiedenen Konzeptualisierungen (z. B. Kong et al., 2024; Laupichler et al., 2023; Ng et al., 2021; Ng, Wu, et al., 2024). Insgesamt lassen sich aber technisches Verständnis, Wissen um den gesellschaftlichen Einfluss von KI sowie KI-Ethik als zentrale Komponenten in verschiedenen Konzeptualisierungen von AI Literacy identifizieren (Lintner, 2024).

Der konkrete Fokus einzelner Konzeptualisierungen ist jedoch von der Zielgruppe abhängig (Chee et al., 2025).

Spezifisch für den Schulkontext kritisieren beispielsweise Chiu et al. (2024) die Definition von Long und Magerko (2020) als zu breit, komplex und ungeeignet. Sie schlagen einen stärkeren Fokus auf selbstreflexive Fähigkeiten vor, um den ständig veränderten Anforderungen durch neue KI-Technologien gerecht zu werden. Touretzky et al. (2019) unterbreiten fünf übergreifende Ideen bezüglich KI, die Schüler:innen verstehen sollten: 1) Computer nehmen die Welt durch Sensoren wahr; 2) Agenten bilden Modelle bzw. Repräsentationen der Welt und nutzen sie für logische Argumentationen („reasoning“); 3) Computer lernen aus Daten; 4) Es ist eine große Herausforderung für Entwickler, Agenten zu erstellen, die mit Menschen interagieren und 5) KI kann die Gesellschaft sowohl positiv als auch negativ beeinflussen.

Insgesamt scheint – sowohl allgemein als auch für den Schulkontext – weitestgehend Einigkeit darüber zu bestehen, dass das Verständnis darüber, wie KI-Technologien funktionieren, eine wichtige Komponente von AI Literacy darstellt. Für den Schulkontext stellt sich daher die Frage, wie AI Literacy und insbesondere ein Verständnis der Funktionsweise von KI bei Lernenden zielgruppenorientiert gefördert werden können.

2.2 Förderung von AI Literacy im Schulkontext

In verschiedenen Studien wurden unterschiedliche Interventionen zur Förderung von AI Literacy untersucht, welche verschiedene pädagogische Ansätze, wie projektbasiertes Lernen (Kong et al., 2024) oder spielerische Elemente (Ali et al., 2021; Ng, Xinyu, et al., 2024), nutzen. Dabei wird jeweils das Ziel verfolgt, dass die Lernenden Wissen zu KI nicht nur verinnerlichen, sondern die Fähigkeit entwickeln, Algorithmen zum Lösen von Problemen zu konstruieren (Lin et al., 2024). Fleischer und Biehler (2025) beispielsweise führen eine Unterrichtseinheit zu Entscheidungsbäumen durch, um Lernende so aus einer mathematischen Perspektive an die Erstellung und Evaluation von KI-Algorithmen heranzuführen. Ng et al. (2024) lassen Schüler:innen selbst KI-getriebene Recyclingtonnen konzipieren und Ng, Xinyu, et al. (2024) implementieren und evaluieren eine Intervention zur Förderung von AI Literacy mittels eines Online-Spiels. Ein spielerischer Ansatz wird auch von Ali et al. (2021) verfolgt, bei dem die Lernenden in die Rolle von KI-Modellen schlüpfen und anhand konkreter, alltäglicher Repräsentationen Wissen zu

generativer KI erwerben. Solche spielerischen, interdisziplinären Zugänge, die an alltägliche Erfahrungen der Lernenden anknüpfen, können eine Möglichkeit darstellen, das Erlernen komplexer Konzepte zu erleichtern (Ng et al., 2024). Dies kann beispielsweise auch durch die Verwendung von Simulationen, Erklärungen der Entscheidungsprozesse von KI-Agenten und interaktive Darstellungen erfolgen (Long & Magerko, 2020).

Allen zuvor beschriebenen Aktivitäten – mit Ausnahme von Fleischer und Biehler (2025) – ist gemein, dass sie grundlegende Konzepte der Funktionsweise von KI vermitteln, ohne dabei die mathematischen oder informatischen Hintergründe der Technologien genauer zu beleuchten. Ali et al. (2021) machen deutlich, dass sie bewusst einen niedrigschwelligen Zugang ohne tiefergehende mathematische Konzepte schaffen möchten. Dies führt unter Umständen bei Lernenden zu dem Gefühl, dass die genaue Funktionsweise von KI trotz der Intervention weiterhin unbekannt bleibt (Ng, Xinyu, et al., 2024).

Soll dieser Herausforderung begegnet werden, so gilt es, nicht nur grundlegende Konzepte von KI zu vermitteln, sondern die dahinterliegenden Algorithmen aus mathematischer Perspektive zu erkunden. Gerade bei komplexeren Algorithmen stellt sich bei der Entwicklung konkreter Lerngelegenheiten dann die Frage, wie es gelingt, mathematische Algorithmen in ihrer Abstraktion zu thematisieren und gleichzeitig der Forderung nach einem spielerischen Zugang nachzukommen, der an alltägliche Erfahrungen der Lernenden anknüpft, um das Verständnis der mathematischen Inhalte zu erleichtern. Projekte, deren Schwerpunkt auf der Erarbeitung einer KI-basierten Lösung für ein reales Problem liegt (wie etwa bei Ng, Su et al., 2024), erscheinen dabei weniger geeignet, wenn die Erarbeitung der mathematischen Inhalte und weniger deren reale Anwendung im Fokus stehen soll. Naheliegender scheint es, künstlich eine Situation zu konstruieren – etwa in Form eines Spiels (Ali et al., 2021; Ng, Xinyu, et al., 2024) – die Analogien zum zu erlernenden Inhalt aufweist und so möglicherweise zu einem verbesserten Verständnis bei Lernenden beiträgt (Ali et al., 2021). Bei einem solchem Ansatz steht die Erarbeitung eines KI-Algorithmus aus mathematischer Perspektive im Fokus – ein Schwerpunkt, der sich zumindest in Teilen als Umgang mit Daten und Algorithmen in Beiträgen zu AI Literacy wiederfinden lässt (Chee et al., 2025; Long & Magerko, 2020; Touretzky et al., 2019). Die Relevanz von Algorithmen für das Verständnis von KI legt nahe, auch das Konzept des Compu-

tational Thinking mit Bezug zu AI Literacy genauer zu diskutieren.

2.3 Computational Thinking

Die Definition von Computational Thinking ist nicht eindeutig (Kadijevich, 2023; Shute et al., 2017; Weintrop et al., 2016). Cuny et al. (2010) beispielsweise definieren Computational Thinking als Denkprozesse beim Formulieren von Problemen und deren Lösungen, sodass die Lösungen durch einen „information-processing agent“ (zitiert in Wing, 2010, S. 1), in der Regel einen Computer, ausgeführt werden können. Shute et al. (2017) beziehen sich ebenfalls auf das Lösen von Problemen und betrachten Computational Thinking als Grundlage dafür, Probleme effizient und effektiv algorithmisch zu lösen und dabei Lösungen zu erzeugen, die in unterschiedlichen Kontexten wiederverwendbar sind.

Während einige Konzeptualisierungen den Bezug von Computational Thinking insbesondere zum Programmieren verdeutlichen (z. B. Brennan & Resnick, 2012), führte die zunehmende Relevanz von digitalen Technologien aber zu Anwendungen, die darüber hinaus gehen (Weintrop et al., 2016). Hierfür wurden eigene Konzeptualisierungen von Computational Thinking erarbeitet, z. B. von Shute et al. (2017), Kadijevich (2018) und Lehmann (2023). Sie zeigen verschiedene, aber sich überschneidende Komponenten von Computational Thinking. Wir führen die zentralen Elemente dieser Konzep-

tualisierungen in Tabelle 1 zusammen, die in dieser Studie zur Analyse des Umgangs mit KI-Algorithmen dienen sollen. Viele der in Tabelle 1 genannten Aspekte (z. B. Zerlegen in Teilschritte oder Abstraktion) sind anschlussfähig an etablierte Konzepte der Mathematikdidaktik, etwa das Problemlösen nach Polya (1945) oder algebraisches Denken (z. B. Kieran et al., 2016).

Da also sowohl AI Literacy als auch Computational Thinking Algorithmen in den Blick nehmen, betrachten wir deren Zusammenhänge im Folgenden.

2.4 Zusammenhang von Computational Thinking und AI Literacy

Ein Zusammenhang zwischen AI Literacy und Computational Thinking fällt bereits bei Betrachtung ihrer Konzeptualisierungen auf. So identifizieren Chee et al. (2025) das Wissen darüber, wie Algorithmen funktionieren, wie sie Informationen repräsentieren und wie sie angewendet werden können, als zentrale Komponente verschiedener Konzeptualisierungen von AI Literacy. Ng et al. (2021) beschreiben Parallelen zwischen Brennan und Resnicks' (2012) Konzeptualisierung von Computational Thinking und AI Literacy, indem sie *AI concepts*, *AI practices* und *AI perspectives* darstellen (in Analogie zu Brennan und Resnicks' *CT concepts*, *CT practices* und *CT perspectives*).

Komponente von CT	Beschreibung	Referenzen
Abstraction	Extrahieren eines zentralen Kerns aus einem komplexen System durch das Sammeln von Daten und deren Analyse, das Erkennen von Mustern und das Erstellen von Modellen oder Simulationen der Funktionsweise eines Systems	Kadijevich, 2018, 2023; Kadijevich et al., 2023; Lehmann, 2023; Shute et al., 2017
Algorithms	Erstellen logischer und geordneter Schritte, um eine Lösung für ein Problem zu finden	Kadijevich, 2018, 2023; Kadijevich et al., 2023; Shute et al., 2017
Decomposition	Zerlegen eines komplexen Problems oder Systems in Teilprobleme	Kadijevich, 2018, 2023; Kadijevich et al., 2023; Lehmann, 2023; Shute et al., 2017
Debugging	Finden und Beheben von Fehlern	Lehmann, 2023; Shute et al., 2017
Automation	Auslagern von Berechnungen an Computer	Kadijevich, 2018, 2023; Kadijevich et al., 2023

Tab. 1: Betrachtete Komponenten des Computational Thinking (CT)

Auf Grund dieses Zusammenspiels von AI Literacy und Computational Thinking auf konzeptueller Ebene, ist es nicht verwunderlich, dass Computational Thinking als relevant für die Bearbeitung von Aktivitäten zu KI angesehen wird (Ng, Su, et al., 2024) und sich ein positiver statistischer Zusammenhang zwischen dem Computational Thinking und AI Literacy identifizieren lässt (Celik, 2023). Zhong & Liu (2025) betrachten Computational Thinking sogar als Subkomponente von AI Literacy.

Auf Basis der Literatur lässt sich festhalten, dass ein – konzeptueller wie statistischer – Zusammenhang zwischen den beiden Konzepten besteht. Um aufzuzeigen, ob sich Komponenten des Computational Thinking auch bei der Erarbeitung von KI-Algorithmen aus mathematischer Perspektive identifizieren lassen, bedarf es zunächst einer geeigneten Lerngelegenheit. Dafür gilt es, die mathematischen Hintergründe eines KI-Algorithmus aufzugreifen, die später die Lernenden (selbst) erarbeiten.

2.5 Reinforcement Learning

Reinforcement Learning (auch *Verstärkendes Lernen* genannt) stellt eine Variante des Machine Learning dar. Machine Learning wiederum ist ein wichtiges Teilgebiet von KI, welches sich mathematischer Verfahren und Modelle bedient. Beim Machine Learning identifiziert ein Algorithmus Muster in Daten, ermittelt daraus Regeln und „lernt“ diese (Ris-Ala, 2023). Dies kann beispielsweise auf Basis der Idee des *Reinforcement Learning* erfolgen. Es kommt dann zum Einsatz, wenn dem Algorithmus zu Beginn noch keine Daten zur Verfügung stehen, diese aber durch „Interaktion“ mit einer Umgebung Schritt für Schritt gewonnen werden können (Ris-Ala, 2023).



Abb. 1: Schematischer Ablauf des Reinforcement Learnings: Ein Agent führt eine Aktion in einer Umgebung durch, wodurch er eine Belohnung erhält und in einen neuen Zustand übergeht. Anschließend, kann eine weitere Aktion durchgeführt werden.

Schematisch lässt sich das wie folgt beschreiben: Beim *Reinforcement Learning* interagiert ein sogenannter Agent (engl. *agent*) – beispielsweise ein

Roboter – mit seiner Umgebung (engl. *environment*), in der er ein bestimmtes Ziel erreichen muss. Innerhalb der Umgebung kann sich der Agent in verschiedenen Zuständen s (engl. *states*) befinden und verschiedene Aktionen a (engl. *actions*) ausführen, für die unterschiedliche Belohnungen oder Bestrafungen R (engl. *rewards*) vergeben werden. Zu Beginn befindet sich der Agent in einem Startzustand und „erkundet“ anschließend mittels Trial-and-Error die Umgebung. Durch jede Aktion geht er von seinem derzeitigen Zustand s in einen neuen Zustand s' über und erhält eine bestimmte Belohnung bzw. Bestrafung (Ris-Ala, 2023) (vgl. Abb. 1). Der Agent wird belohnt, wenn sein Verhalten dazu beiträgt, das Ziel zu erreichen. Die Belohnungen werden also je nach Ziel der Aufgabe unterschiedlich festgelegt (Sutton & Barto, 2018). In diesem Beitrag interpretieren wir unter Belohnungen auch Bestrafungen, also „negative“ Belohnungen.

Der Zweck des Reinforcement Learnings ist es, eine Verhaltensweise zu finden, für die die langfristigen, kumulierten Belohnungen bis zum Erreichen des Ziels maximal werden (Ris-Ala, 2023). Durch das Reinforcement Learnings soll eine solche „optimale“ Verhaltensweise entwickelt werden.

Reinforcement Learning eignet sich für eine Vielzahl von realen Anwendungen (für einen Überblick, siehe Naeem et al., 2020), beispielsweise in der Robotik (Singh et al., 2022), im Bereich von Videospielen (Shao et al., 2019) oder bei der intelligenten Steuerung von Verkehrsflüssen (Khamis & Gomaa, 2014). Auch beim Training von Large Language Models, welche auf Basis des Feedbacks von Nutzer:innen verbessert werden, wird Reinforcement Learning genutzt (Havrilla et al., 2024).

2.6 Der Q-Learning-Algorithmus als Beispiel für Reinforcement Learning

Eine Möglichkeit, eine optimale Verhaltensweise zu finden, stellt der Q-Learning-Algorithmus (Watkins, 1989) dar. Mithilfe dieses Algorithmus soll für jeden Zustand (mindestens) eine optimale Aktion gefunden werden, um ein Ziel zu erreichen. Im Folgenden betrachten wir zuerst das Ergebnis des Q-Learning-Algorithmus – aus dem die optimale Verhaltensweise, das heißt eine Verkettung von Handlungen, die zu einer maximalen Belohnung führen, direkt abgelesen werden kann – und anschließend den Algorithmus selbst.

```

1 Setze  $Q(s, a) = 0$  für alle Zustände  $s$  und alle Aktionen  $a$ 
2 Wiederhole:
3   Setze als Zustand den Startzustand
4   Wiederhole:
5     Wähle zufällig Aktion  $a$ 
6     Führe Aktion  $a$  aus, erhalte Belohnung  $R$  und neuen Zustand  $s'$ 
7     Berechne  $Q_{neu}(s, a)$  als  $Q_{neu}(s, a) = Q_{alt}(s, a) + \alpha \cdot [(R + \gamma \cdot \max_{a' \in A} Q_{alt}(s', a')) - Q_{alt}(s, a)]$ 
8     Setze  $s'$  als neuen Zustand  $s$ 
9   bis Zielzustand oder max. Anzahl Schritte erreicht

```

Abb. 2: (Adaptierter) Q-Learning-Algorithmus

Das Ergebnis des Q-Learning-Algorithmus ist eine *finale Q-Tabelle*. Darin sind alle möglichen Zustände und Aktionen gegeneinander aufgetragen (vgl. Tab. 2). Ein Eintrag repräsentiert die Qualität der jeweiligen Aktion im entsprechenden Zustand. Höhere Zahlen drücken eine „bessere“ Zustands-Aktions-Kombination aus, also „bessere Verhaltensweisen“ (Ris-Ala, 2023). In Tabelle 2 ist beispielsweise in Zustand s_1 die Aktion a_2 am „besten“ und die Aktion a_3 am „schlechtesten“. In Zustand s_4 hingegen sind keine Informationen über die beste Aktion vorhanden. Dies spricht dafür, dass es sich bei s_4 um einen Zustand handelt, der vom Agenten nicht mehr verlassen werden kann oder nie erreicht wurde. Aus der dargestellten Q-Tabelle kann also eine optimale Verhaltensweise direkt abgelesen werden (Ris-Ala, 2023).

	a_1	a_2	a_3	a_4
s_1	5	10	1	5
s_2	1	5	1	5
s_3	1	5	5	10
s_4	0	0	0	0

Tab. 2: Beispiel einer finalen Q-Tabelle mit Zuständen (s_i) und Aktionen (a_i) am Ende eines Q-Learning-Prozesses; höhere Werte repräsentieren zielführendere Zustands-Aktions-Kombinationen

Die exakten Zahlen in der finalen Q-Tabelle – die sogenannten Q-Werte – lassen sich berechnen, wenn man bereits eine optimale Verhaltensweise kennt. Angenommen, der Agent befindet sich im Zustand s und vollzieht von dort aus Aktion a . Von dem Zustand aus, der nach Aktion a erlangt wird, werden bei optimaler Verhaltensweise n Aktionen bis zum Zielzustand benötigt. Dann ist

$$(1) \quad Q(s, a) = R_0 + \gamma \cdot R_1 + \gamma^2 \cdot R_2 + \dots + \gamma^n \cdot R_T.$$

R_0 ist dabei die Belohnung nach Aktion a und R_t für $t \geq 1$ drückt die Belohnung aus, die der Agent nach der t -ten Aktion nach Aktion a bekommt, wenn er sich immer optimal verhält. R_T ist die Belohnung

nach der letzten Aktion (Sutton & Barto, 2018), also entweder bei Erreichen des Ziels oder nach einer festgelegten Anzahl maximaler Aktionen, wenn das Ziel bis dahin nicht erreicht wurde. Durch den Parameter γ , der immer zwischen 0 und 1 liegt, erfolgt eine Gewichtung der Belohnungen.

Gleichung (1) beschreibt somit die finalen Q-Werte in Abhängigkeit der verschiedenen Parameter und einer optimalen Verhaltensweise. Wenn eine optimale Verhaltensweise bereits bekannt ist, handelt es sich jedoch um ein gelöstes Problem. Folglich gäbe es keinen Anlass, auf Reinforcement Learning zurückzugreifen. Der Q-Learning-Algorithmus muss deshalb die finale Q-Tabelle anderweitig produzieren, um die optimale Verhaltensweise ableiten zu können. Abbildung 2 zeigt eine Variante des entsprechenden Q-Learning-Algorithmus.

Zu Beginn des Algorithmus können in der Q-Tabelle zunächst alle Q-Werte auf null gesetzt werden (vgl. Abb. 2, Zeile 1) (Sutton & Barto, 2018). Während des Q-Learning-Prozesses führt der Agent – wie beim Reinforcement Learning allgemein (vgl. Abb. 1) – Aktionen aus und erhält dafür neue Zustände sowie Belohnungen (vgl. Abb. 2, Zeile 5, 6 und 8). Bei jeder Aktion wird eine Veränderung im entsprechenden Eintrag der Q-Tabelle iterativ mittels der Formel in Zeile 7 (vgl. Abb. 2) vorgenommen. Dabei gibt der Parameter α die Lernrate an, welche bestimmt, wie stark sich Q-Werte ändern (Sutton & Barto, 2018). Unter bestimmten Voraussetzungen nähern sich die Werte in der Q-Tabelle nach hinreichend vielen Iterationen den finalen Q-Werten an, aus denen dann eine optimale Verhaltensweise abgelesen werden kann. Ein anschauliches Beispiel für die Anwendung dieses Algorithmus wird in Abschnitt 3 beschrieben.

3. Didaktische Aufbereitung des Q-Learnings

Um Q-Learning mit Schüler:innen zu thematisieren, bedarf es einer entsprechenden didaktischen Aufbereitung. Wir stellen diesbezüglich zunächst dar, wie Q-Learning bzw. Reinforcement Learning bisher in

Bildungskontexten thematisiert wurde, um anschließend eine mögliche didaktische Aufbereitung für das Q-Learning aus mathematikdidaktischer Perspektive vorzustellen.

3.1 Thematisierung von Q-Learning und Reinforcement Learning in Bildungskontexten

Q-Learning wurde bereits in universitären Bildungskontexten eingesetzt (Martínez-Tenor et al., 2019; Podobnik et al., 2024). Für den Schulkontext unterbreitet beispielsweise Dauscher (2019) einen Vorschlag, um Q-Learning mithilfe eines Spiels im Informatikunterricht der Mittelstufe zu thematisieren. Insgesamt spielt Reinforcement Learning im schulischen Kontext bei der Vermittlung von Konzepten des maschinellen Lernens allerdings eine untergeordnete Rolle (Marques et al., 2020). Das spiegelt sich auch aktuell in Lehrplänen wider (z. B. ISB, 2025). All den genannten Beispielen ist gemein, dass sie primär die technische Komponente und Implementierung fokussieren und weniger die mathematischen Hintergründe des Q-Learnings. Biehler & Fleischer (2024) nehmen eine mathematische Perspektive zu Reinforcement Learning ein und fokussieren dabei die Thematisierung grundlegender Strategien des Reinforcement Learning. Die Erarbeitung konkreter (etablierter) Algorithmen aus mathematischer Perspektive, wie dem Q-Learning, findet hierbei aber keine explizite Erwähnung.

Dabei lässt sich Q-Learning durchaus tiefgehend aus mathematischer Perspektive erörtern. In Sinne bisheriger Forschung zur Förderung von AI Literacy (vgl. Abschnitt 2.2) liegt es nahe, als Unterstützung dafür ein fiktives Setting – ähnlich zu einem Spiel – zu wählen, das Parallelen zum zu erlernenden (mathematischen) Inhalt aufweist, sodass die Lernenden bei der Erarbeitung auf ihr Weltwissen über diese Situation zurückgreifen können. In diesem Sinne verstehen wir unter einem *Setting* eine exemplarische Interpretation des mathematischen Inhaltes, derart dass Rückschlüsse von der Interpretation (und assoziierten Eigenschaften) auf den Inhalt unter Zuhilfenahme von Weltwissen möglich sind. Hier stellt das Setting somit eine exemplarische Interpretation der allgemeinen Komponenten des Q-Learnings dar (z. B. des Agenten, der Umgebung und der Belohnungen). Es kommt zu einem Perspektivwechsel: Die Lernenden sollen *nicht* ein (reales oder fiktives) Problem mithilfe eines (geeigneten) KI-Algorithmus lösen. Vielmehr bearbeiten sie eine Art „Rekonstruktionsproblem“, in dem sie aus der Lösung eines (realen oder fiktiven) Problems in einem bestimmten Setting den Algorithmus erarbeiten bzw.

rekonstruieren, der das Problem gelöst hat. Das ursprüngliche (reale oder fiktive) Problem und das Setting dienen dabei als Unterstützung bei der mathematischen Erarbeitung. Für das Q-Learning wählen wir für diesen Beitrag als fiktives, spielerisches Setting die sogenannte Gridworld (z. B. Lorenz, 2024).

3.2 Beschreibung der Gridworld

Die Gridworld bildet eine Rahmung des mathematischen Q-Learning-Algorithmus, über deren Funktionsweise und Eigenschaften die Lernenden ein gewisses Weltwissen (z. B. aus spielerischen Kontexten) besitzen. Eine Gridworld besteht aus einem fiktiven Labyrinth, in dem sich eine Figur – in unserem Fall ein fiktiver Roboter – bewegen kann. Diese Bewegungen kann er allerdings nur nach oben, unten, rechts und links durchführen. Er soll den bestmöglichen Weg zu einem Ziel finden. Behindert wird der Roboter dabei beispielsweise von Wänden und Feldern mit speziellen Eigenschaften (Lorenz, 2024; Sutton & Barto, 2018).

Abbildung 3 zeigt ein mögliches Labyrinth mit Startfeld (1,A), Zielfeld (5,E) und verschiedenen Sonderfeldern (grün, blau, blau mit Floß). Das Betreten der Sonderfelder ist günstiger bzw. ungünstiger als das Betreten normaler Felder. Es kann auch Felder geben, die gar nicht betreten werden können (in Abbildung 3 grün dargestellt).

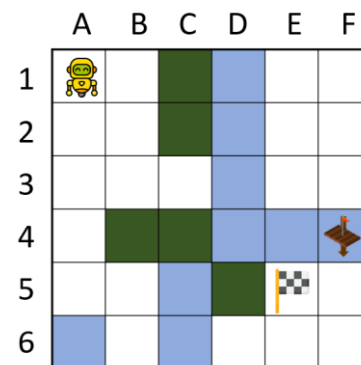


Abb. 3: Beispiel einer 6x6-Gridworld mit Sonderfeldern. Feld (1,A) stellt das Startfeld und (5,E) das Zielfeld dar.

Es lässt sich vermuten, dass die Gridworld an Erfahrungen der Lernenden anknüpft (z. B. aus Videospielen) oder zumindest intuitiv nachvollzogen werden kann. Mithilfe der Gridworld soll daher – durch die Möglichkeit eines Rückgriffs auf Weltwissen – das Verständnis des abstrakten Q-Learning-Algorithmus erleichtert werden. Hierzu wird im Folgenden zunächst eine Beziehung zwischen der Gridworld und dem Reinforcement Learning und anschließend zwischen der Gridworld und Q-Learning hergestellt. Dabei ergänzen wir jeweils weitere Details in der

Gridworld, um die Beziehungen zwischen der Gridworld als Setting und dem abstrakten Reinforcement Learning bzw. Q-Learning deutlicher zu machen.

3.3 Navigieren in der Gridworld mittels Reinforcement Learning

Übertragen wir zunächst die Terminologie des Reinforcement Learnings (vgl. Abschnitt 2.5) auf die Gridworld, so stellt der Roboter den Agenten und die Gridworld die Umgebung dar. Die Felder der Gridworld, auf denen sich der Agent befinden kann, stellen die möglichen Zustände der Gridworld dar (Sutton & Barto, 2018). Auf jedem Feld sind die vier oben genannten Bewegungsaktionen möglich. Für jeden Schritt, den der Roboter ausführt, erhält er einen neuen Zustand – er steht dann auf einem neuen Feld oder bleibt auf dem Ausgangsfeld, wenn er beispielsweise versucht, einen Schritt in Richtung Wand zu machen. Zudem erhält der Roboter für jeden Schritt eine Belohnung. In unserem Fall wird diese *nur dann* positiv gewählt, wenn der Roboter das Ziel erreicht. Die Belohnung für alle andere Schritte (einschließlich solcher gegen die Wand) ist negativ – selbst, wenn sie den Roboter näher zum Ziel bringen! Dies mag zunächst kontraintuitiv erscheinen, da ein Schritt in Richtung Ziel eine vermeintlich „gute“ Aktion darstellt. Es wird hier jedoch nur ein tatsächlicher abschließender „Erfolg“ positiv belohnt, nämlich das Erreichen des Ziels. Nach dem Erreichen des Ziels – oder nach einer festgelegten Anzahl an Schritten, falls das Ziel bis dahin nicht erreicht wurde – beginnt der Roboter wieder auf dem Startfeld. Dabei ist die Anzahl der maximalen Schrittzahl pro Durchlauf fest und unabhängig davon, welche Belohnungen während des Durchlaufs gesammelt wurden.

Um die negativen Belohnungen für jeden Schritt außer den Schritt auf das Zielfeld nachvollziehbarer zu machen, knüpfen wir an die Vorstellung eines Akkus an: Der Akku des Roboters ist auf dem Startfeld voll aufgeladen. In der 6x6-Gridworld (vgl. Abb. 3) verliert der Roboter mit jedem Schritt fünf Prozent der Akkuladung. Am Zielfeld wird sein Akku wieder auf hundert Prozent aufgeladen. Die Analogie des Akkus zielt also darauf ab, einen möglicherweise irritierend wirkenden Sachverhalt (negative Belohnungen auch bei Bewegung in Richtung Ziel) intuitiv, d. h. mittels Weltwissen über das Setting, nachvollziehbar zu machen. Wir wählen konkret auf dem Zielfeld die Belohnung +100 und für alle anderen Felder -5. Die Belohnung von +100 wurde in Anlehnung an das vollständige Laden eines leeren Akkus gewählt. Ist der

Akku des Roboters leer, bevor dieser das Ziel erreicht, muss er wieder auf dem Startfeld beginnen. Er hat insgesamt beliebig viele Versuche. Für Sonderfelder, auf die ein Schritt besonders „gut“ bzw. „schlecht“ ist, kann ebenfalls die Analogie des Akkustands verwendet werden: Angenommen, im Beispiel in Abbildung 3 stellen die blauen Felder besonders schlechte Felder dar. In der Gridworld könnten diese als Wasserfelder interpretiert werden, auf denen besonders viel Energie des Akkus verbraucht wird, weil der Widerstand im Wasser größer ist. Zum Schluss des Lernprozesses ergibt sich der Weg mit geringstem Energieverbrauch, welcher als optimal betrachtet werden kann. Die optimale Verhaltensweise in der Gridworld gibt also für jedes Feld eine oder mehrere Richtungen an, die Teil eines optimalen Weges zum Ziel sind.

Dabei ist einschränkend anzumerken, dass die Belohnung am Ziel unabhängig von der aktuellen „Akkuladung“ erhalten wird, so dass der Vergleich zum Ladeprozess nur bedingt gegeben ist. Erreicht der Roboter nämlich das Ziel beispielsweise mit einem Restakkustand von 20 Prozent, so beträgt die Belohnung trotzdem +100 und nicht etwa +80. Zudem würde bei der Reduktion des Akkustandes pro Schritt die maximal mögliche Anzahl an Schritten von den erhaltenen Belohnungen (also der Art und Anzahl der überquerten Felder) während des Durchlaufs abhängen, was beim Reinforcement Learning nicht der Fall ist. Hier wird deutlich, dass die Problemsituation zwar eine Unterstützung zum Begreifen des (mathematischen) Inhalts sein kann, jedoch nicht in allen Details eine exakte Entsprechung darstellt.

3.4 Veranschaulichung des Q-Learnings anhand der Gridworld

Zur Veranschaulichung des Q-Learning wird der Einfachheit halber zunächst eine 2x2-Gridworld ohne Sonderfelder (vgl. Abb. 4) betrachtet. Da nun weniger Schritte nötig sind, um zum Ziel zu kommen, setzen wir den Akkuverlust für jeden Schritt auf -10.

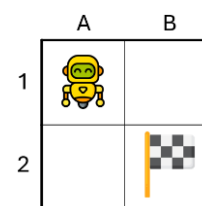


Abb. 4: Beispiel einer 2x2-Gridworld ohne Hindernisse. Feld (1, A) stellt das Startfeld und (2, B) das Zielfeld dar.

Zunächst werden beim Q-Learning in der Gridworld die vier Felder sowie die vier Aktionen gegen-

einander in einer Q-Tabelle aufgetragen und alle Q-Werte auf 0 gesetzt (vgl. Abb. 5, Schritt 0). Anschließend wählt der Roboter zufällig eine Laufrichtung und macht einen Schritt. Bei jedem Schritt wird eine Veränderung im entsprechenden Feld der Q-Tabelle auf Basis der Q-Lernregel vorgenommen (Sutton & Barto, 2018). In der Spielsituation interpretiert die Q-Lernregel an, wie der Roboter neue Informationen mit alten verrechnet bzw. wie er sich neue Informationen „merkt“. Notwendig für die Lernregel sind dabei die beiden Parameter α und γ . Der Parameter α gibt anschaulich an, wie sehr der Roboter alte Informationen über die Qualität von Feldern mit neuen überschreibt. Der Parameter γ drückt aus, wie stark Akkuverluste bzw. -gewinne gewichtet werden. Abbildung 5 zeigt die ersten vier Q-Tabellen in der 2x2-Gridworld.

Schritt 0				
	up	down	left	right
(1,A)	0.0	0.0	0.0	0.0
(1,B)	0.0	0.0	0.0	0.0
(2,A)	0.0	0.0	0.0	0.0
(2,B)	0.0	0.0	0.0	0.0

Schritt 1				
	up	down	left	right
(1,A)	0.0	0.0	-8.0	0.0
(1,B)	0.0	0.0	0.0	0.0
(2,A)	0.0	0.0	0.0	0.0
(2,B)	0.0	0.0	0.0	0.0

Schritt 2				
	up	down	left	right
(1,A)	0.0	0.0	-8.0	-8.0
(1,B)	0.0	0.0	0.0	0.0
(2,A)	0.0	0.0	0.0	0.0
(2,B)	0.0	0.0	0.0	0.0

Schritt 3				
	up	down	left	right
(1,A)	0.0	0.0	-8.0	-8.0
(1,B)	0.0	80.0	0.0	0.0
(2,A)	0.0	0.0	0.0	0.0
(2,B)	0.0	0.0	0.0	0.0

Abb. 5: Vier exemplarische Q-Tabellen bei einem Q-Learning-Prozess in einer 2x2-Gridworld mit $R_{\text{Ziel}} = -10$ bzw. $R_{\text{Ziel}} = 100$, $\gamma = 0,8$, $\alpha = 0,8$

Die Tabellen können folgendermaßen interpretiert werden: Der Roboter läuft zufällig zunächst nach links, stößt auf eine Wand und bleibt auf dem Feld (1,A), da er auf kein neues gelangt. Anschließend läuft er zufällig nach rechts und steht auf (1,B). Von dort aus läuft er zufällig nach unten auf das Zielfeld. Danach beginnt er wieder auf dem Startfeld, kann aber bereits gelernte Informationen aus der Q-Tabelle mithilfe der Q-Lernregel weiterverwenden (Ris-Ala, 2023). Die Intuition, dass der Roboter, wie in einem Spiel, wieder von vorne anfangen muss, wenn sein Akku leer ist, stimmt somit mit der tatsächlichen Funktionsweise des Algorithmus überein. Die Q-Tabelle lässt sich als das „Gedächtnis“ des Roboters interpretieren, in dem Informationen über die Qualität bereits erkundeter Wege gespeichert werden.

Im Laufe des Q-Learning-Prozesses nähern sich die Werte in der Q-Tabelle iterativ den finalen Q-Werten an, während sich der Roboter in der Gridworld bewegt. Die finale Q-Tabelle ist in Abbildung 6 dargestellt.

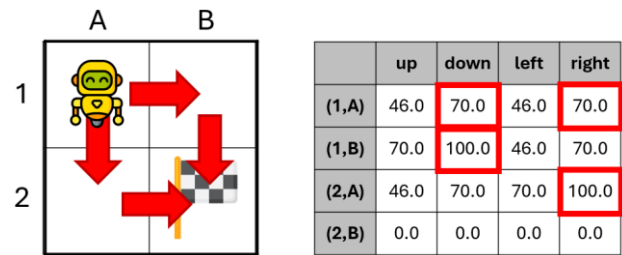


Abb. 6: Links: Optimale Wege zum Ziel in einer 2x2-Gridworld; Rechts: Finale Q-Tabelle der 2x2-Gridworld. Die jeweils höchsten Werte geben die „besten“ Schritte an.

Exemplarische Berechnung der finalen Q-Werte

Die Zahlen aus der Q-Tabelle in Abbildung 6 lassen sich mithilfe der Gleichung

$$Q(s, a) = R_0 + \gamma \cdot R_1 + \gamma^2 \cdot R_2 + \dots + \gamma^n \cdot R_T$$

nachrechnen. Es wurde hier $\gamma = 0,8$ gewählt, sowie die Belohnungen +100 für das Zielfeld und -10 für alle anderen Felder. Steht der Roboter beispielsweise auf Feld (1,A) und läuft nach unten, so erhält er eine Belohnung von -10 und braucht danach noch einen Schritt zum Ziel ($n = 1$). Im Ziel erhält er eine Belohnung von +100. Als Q-Wert ergibt sich $Q((1,A), \text{"down"}) = -10 + 0,8 \cdot 100 = 70$.

4. Forschungsfragen

Prozesse des Computational Thinking stehen im Zusammenhang mit AI Literacy bzw. dem Thema KI (vgl. Abschnitt 2.4). Vor diesem Hintergrund wird in der vorliegenden Studie geprüft, ob sich Komponenten des Computational Thinking auch im Rahmen der mathematischen Erarbeitung eines Reinforcement-Learning-Algorithmus identifizieren lassen.

(FF 1) *Lassen sich Komponenten des Computational Thinking beim Erarbeiten a) der finalen Q-Tabelle des Q-Learning-Algorithmus und b) der Q-Lernregel identifizieren?*

Die spielerische Aufbereitung und der Rückgriff auf Weltwissen der Lernenden bei dieser Erarbeitung knüpft an die Empfehlungen zur Gestaltung von Lernumgebung zur Förderung von AI Literacy an (vgl. Abschnitt 2.2). Anders als in vielen bisherigen Untersuchungen hierzu liegt der Fokus dieser Studie auf den mathematischen Grundlagen des KI-Algorithmus zwecks Aufbaus eines Verständnisses. Es stellt sich die Frage, ob es gelingt, auch hier ein fiktives Setting für den Q-Learning-Algorithmus zu gestalten, bei dem die Lernenden auf ihr Weltwissen zurückgreifen:

(FF 2) *Inwieweit greifen die Lernenden beim Erarbeiten a) der finalen Q-Tabelle des Q-Learning-Algorithmus und b) der Q-Lernregel auf das*

fiktive Setting der Gridworld zurück (oder beschränken sich auf die mathematische Ebene)?

5. Methodik

5.1 Datenerhebung

Die Daten wurden bei den jährlich stattfindenden Schülerprojekttagen der Universität Würzburg erhoben. Hier wird motivierten Schüler:innen der 10. und 11. Klasse an Gymnasien ermöglicht, vier Tage lang in verschiedenen Kleingruppen an mathematikbezogenen Projekten zu arbeiten. Ihre Arbeit legen die Schüler:innen zum Abschluss in einem Projektbericht dar und präsentieren ihr Projekt vor Publikum.

Im Rahmen der Schülerprojekttag 2024 beschäftigte sich eine Gruppe anhand der entwickelten Lerneinheit zu Q-Learning eingehend mit dem Thema KI und Reinforcement Learning. Die gesamte Bearbeitung wurde videographiert. Abbildung 7 zeigt den Aufbau der Lerneinheit zum Q-Learning.

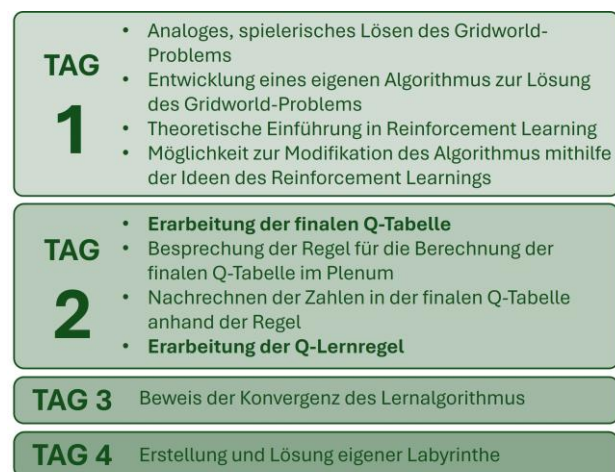


Abb. 7: Aufbau der Schülerprojekttag. Für die Studie wurde die Erarbeitung der Bedeutung der finalen Q-Tabelle und die Erarbeitung der Q-Lernregel ausgewertet.

Am ersten Tag sollten sich die Lernenden zunächst mit dem Spiel bzw. Setting selbst und den grundlegenden Prinzipien des Reinforcement Learnings vertraut machen. Hierbei kamen noch keine digitalen Werkzeuge zum Einsatz. Zunächst sollten die Lernenden in Partnerarbeit den schnellsten Weg durch zwei Labyrinth der Größe 6x6 – eines ohne und eines mit Sonderfeldern (vgl. Abb. 3) – finden. Sie durften dabei verschiedene Strategien und Vorgehensweisen eigenständig erproben. Danach fassten die Schüler:innen ihre Erkenntnisse – immer noch ohne digitale Werkzeuge – in einem eigenen Algorithmus zusammen. Nach einer darauffolgenden theoretischen Einführung in das Thema Reinforcement Learning sollten sie ihre Algorithmen mit den Prinzipien

dieser Variante des Machine Learning vergleichen und hatten die Gelegenheit, ihre Algorithmen gegebenenfalls zu optimieren. Nachdem sich die Lernenden am ersten Tag intensiv mit der Gridworld beschäftigt hatten, stand im Fokus des zweiten Tages die Erschließung des mathematischen Inhalts des Q-Learnings. Dazu erhielten die Lernenden zwei digitale Simulationen bzw. Lernumgebungen: eine zur finalen Q-Tabelle (vgl. Abb. 8) und eine zu der Abfolge der Q-Tabellen während des Lernprozesses mit der Q-Lernregel (vgl. Abb. 9).

Die finale Q-Tabelle bezog sich auf dasselbe Labyrinth (vgl. Abb. 3), welches am Tag zuvor analog betrachtet worden war. Die Lernenden kannten also die Beschaffenheit des Labyrinths sowie den optimalen Weg. Sie sollten nun erklären, was die Zahlen in der finalen Q-Tabelle bedeuten bzw. diese mithilfe ihres Wissens über das Setting (z. B. über den optimalen Weg) interpretieren.

Q = training(env, episodes=1000, steps=10)

	up	down	left	right
(1, A)	-6.715456	-2.14432	-6.715456	-2.14432
(1, B)	-2.144320	3.56960	-6.715456	-2.14432
(1, C)	0.000000	0.00000	0.000000	0.00000
(1, D)	8.912000	18.64000	8.912000	23.64000
(1, E)	23.640000	35.80000	8.912000	17.24000
(1, F)	17.240000	27.80000	23.640000	17.24000
(2, A)	-6.715456	3.56960	-2.144320	3.56960
(2, B)	-2.144320	10.71200	-2.144320	3.56960
(2, C)	0.000000	0.00000	0.000000	0.00000
(2, D)	8.912000	30.80000	18.640000	35.80000
(2, E)	23.640000	51.00000	18.640000	27.80000
(2, F)	17.240000	41.00000	35.800000	27.80000
(3, A)	-2.144320	-0.83360	3.569600	10.71200
(3, B)	3.569600	10.71200	3.569600	19.64000
(3, C)	19.640000	19.64000	10.712000	30.80000
(3, D)	18.640000	46.00000	19.640000	51.00000
(3, E)	35.800000	70.00000	30.800000	41.00000
(3, F)	27.800000	57.50000	51.000000	41.00000
(4, A)	3.569600	5.20800	-0.833600	-0.83360
(4, B)	0.000000	0.00000	0.000000	0.00000
(4, C)	0.000000	0.00000	0.000000	0.00000
(4, D)	30.800000	46.00000	46.000000	70.00000
(4, E)	51.000000	100.00000	46.000000	57.50000
(4, F)	41.000000	75.00000	70.000000	57.50000
(5, A)	-0.833600	7.76000	5.208000	12.76000
(5, B)	12.760000	22.20000	5.208000	17.20000
(5, C)	17.200000	34.00000	12.760000	17.20000
(5, D)	0.000000	0.00000	0.000000	0.00000
(5, E)	0.000000	0.00000	0.000000	0.00000
(5, F)	57.500000	55.00000	100.00000	75.00000
(6, A)	5.208000	7.76000	7.760000	22.20000
(6, B)	12.760000	22.20000	7.760000	34.00000
(6, C)	17.200000	34.00000	22.200000	55.00000
(6, D)	55.000000	55.00000	34.000000	75.00000
(6, E)	100.00000	75.00000	55.000000	55.00000
(6, F)	75.000000	55.00000	75.000000	55.00000

Abb. 8: Aufbau der Simulation zur finalen Q-Tabelle für eine 6x6-Gridworld. Die Anzahl der Episoden und Schritte kann verändert werden. In der Abbildung ist $\gamma = 0,8$.

Konkret erhielten sie die folgende Frage:

Welche Erkenntnisse könnt ihr anhand der Q-Tabelle gewinnen?

Um diese zu beantworten, hatten sie die Möglichkeit, die Anzahl der Schritte (*steps*) und Episoden (*episodes*) abzuändern, um etwaige Vermutungen zu bestätigen oder zu verwerfen.

Im zweiten Schritt erhielten die Lernenden eine interaktive Simulation, in der sie sich selbstständig verschiedene Abfolgen von Q-Tabellen anzeigen lassen konnten. Die Frage an die Lernenden lautete:

Welche Erkenntnisse könnt ihr anhand der Q-Tabellen der einzelnen Schritte gewinnen?

Die Simulation ermöglichte es, verschiedene Parameter zu verändern: Episoden und Schritte wie bei der finalen Q-Tabelle, Alpha und Gamma aus der Q-Lernregel und drei weitere Parameter für Anzeigeeoptionen der Q-Tabellen. Um die Situation zu vereinfachen, wurde die Größe des Labyrinths auf 2×2 (vgl. Abb. 9) reduziert.

Q = training(env, episodes=1000, steps=10, alpha=0.8, gamma=0.8, show_Q=True, Q_step=1, max_Q=10)

	up	down	left	right
(1,A)	0.0	0.0	0.0	0.0
(1,B)	0.0	0.0	0.0	0.0
(2,A)	0.0	0.0	0.0	0.0
(2,B)	0.0	0.0	0.0	0.0

	up	down	left	right
(1,A)	0.0	0.0	-8.0	0.0
(1,B)	0.0	0.0	0.0	0.0
(2,A)	0.0	0.0	0.0	0.0
(2,B)	0.0	0.0	0.0	0.0

...

Abb. 9: Aufbau der Simulation zur Q-Lernregel für eine 2×2-Gridworld. Verschiedene Parameter können verändert werden.

Aufgabe der Lernenden war es nun, sich zu überlegen, wie das Computerprogramm schrittweise die finale Q-Tabelle errechnet. Im Fokus stand dabei die Erschließung der Q-Lernregel. Die Ergebnisse stellten dennoch lediglich Vermutungen der Lernenden dar, welche durch die Simulation gestützt wurden. Am dritten Tag folgte deshalb noch ein mathematischer Beweis, um die Konvergenz der Q-Lernregel zu bestätigen. Die Lerneinheit schloss am vierten Tag mit der Erstellung eigener Labyrinth und eines Projektberichts ab.

5.2 Teilnehmende

Die Gruppe des Projekts zu Q-Learning bestand aus sechs Teilnehmenden (vier männlich, zwei weiblich). Fünf Schüler:innen besuchten zum Zeitpunkt der Erhebung die 11. Klasse des Gymnasiums und eine Schülerin besuchte die 10. Klasse. Der organisatorische Rahmen der Schülerprojekttage legt nahe, dass es sich bei der Stichprobe um eine Positivauswahl

handelt. Während der Schülerprojekttage arbeiteten die Schüler:innen in drei Zweiergruppen (Gruppe 1, 2 und 3) zusammen.

5.3 Datenauswertung

Um die erste Forschungsfrage zu beantworten, wurden zunächst die Teile der Videos, in denen die Schüler:innen die finale Q-Tabelle und die Q-Lernregel erarbeiten, transkribiert. Anschließend wurden Fallzusammenfassungen der insgesamt sechs Erarbeitungsphasen erstellt. Kuckartz (2018) beschreibt Fallzusammenfassungen als eine „systematisch ordnende, zusammenfassende Darstellung der Charakteristika dieses Einzelfalls“, welche vor dem „Hintergrund der Forschungsfrage zentrale Charakterisierungen des jeweiligen Einzelfalls“ aufzeigt (S. 58). Im Sinne dieser Zielsetzung wurden die Fallzusammenfassungen mit Blick auf die fünf Kategorien des Computational Thinking (vgl. Abschnitt 2.3) erstellt. Hierbei sollen lediglich Existenzaussagen über die verschiedenen Komponenten des Computational Thinking getroffen werden (siehe Forschungsfrage), weshalb wir von einer systematischen Kodierung der Transkripte mit Überprüfung der Inter-coder-Reliabilität absehen. Stattdessen wurden von Erst- und Zweitautor:in konsensuell Fallzusammenfassungen erstellt, anhand derer und ausgewählter Zitate wir exemplarisch aufzeigen, ob die fünf Komponenten in den Erarbeitungsprozessen eine Rolle spielen.

Zur Beantwortung von Forschungsfrage 2 wurden die Transkripte der sechs Erarbeitungsphasen im Rahmen einer qualitativen Inhaltsanalyse nach Kuckartz (2019) analysiert. Im Gegensatz zu FF1 ist hier das Ziel, den Arbeitsprozess reliabel und objektiv, d.h., reproduzierbar, in abstrahierter Form abzubilden, weshalb eine Kodierung des gesamten Datenmaterials vorgenommen wurde. Als Kodiereinheit betrachten wir jeweils einen zusammenhängenden Redebeitrag einer sprechenden Person bis zum nächsten Sprecherwechsel. Für die qualitative Inhaltsanalyse wurde ein konzeptgeleitetes Kategoriensystem verwendet, mit dem erfasst werden soll, inwieweit die Lernenden bei der Erarbeitung auf das fiktive Setting eingehen (z. B. auf das Verhalten eines Roboters in der Gridworld). Es ist möglich, dass sich Lernende in einigen Aussagen nur auf das Setting beziehen (z. B., wenn sie beschreiben, wie der Akkustand des Roboters beim Laufen durch das Labyrinth sinkt) und keinerlei mathematische Überlegungen verbalisieren. Andererseits können Aussagen auch nur auf mathematische Beschreibungen des Q-Learnings zurückgreifen und überhaupt nicht auf das Setting. Denkbar sind zudem Kombinationen

der beiden Pole, bei denen die Lernenden das Setting mit dem mathematischen Algorithmus des Q-Learnings in Beziehung setzen. Wir betrachten drei Kategorien: *Setting*, *mathematischer Inhalt* sowie *Setting & mathematischer Inhalt*. Als Unterstützung für die Kodierung ziehen wir die Begrifflichkeiten für die beiden Kategorien aus Tabelle 3 heran.

Terminologie im abstrakten Algorithmus (mathematischer Inhalt)	Beispiele für Terminologie im Setting der Gridworld
Belohnungen / Bestrafung / Reward	Akkuverluste
Action / Aktionen	Schritt; Laufen / hochgehen / runtergehen / Richtungen
States / Zustände	Felder
Q-Lernregel	Merken von Informationen
Algorithmus	„Entdeckungsstrategie“ / Vorgehen / was der Roboter tut
Q-Wert	Bewertung eines Schrittes auf einem Feld hinsichtlich der Erreichung des Ziels
Q-Tabelle	Gedächtnis
Episode	Einmal durchlaufen / Durchgang
Alpha	Wie stark alte Informationen über Akkustände mit neuen überschrieben werden
Gamma	Wie stark spätere Akkuverluste weniger gewichtet werden

Tab. 3: Parallelen zwischen Setting (Gridworld) und (mathematischem) Inhalt (Q-Learning)

Tabelle 4 zeigt beispielhafte Tätigkeiten der Schüler:innen für die drei Kategorien. Die Objektivität der Anwendung des Kategoriensystems erfolgt durch zwei unabhängige Kodierungen. Die Übereinstimmung wurde mittels Brennan und Predigers (1981) Kappa ermittelt. Mit $\kappa = 0,69$ liegt eine gute Übereinstimmung (Kuckartz, 2016; Landis & Koch, 1977) vor. Nichtübereinstimmende Fälle wurden konsens-kodiert.

Kategorie	Beispielhafte Tätigkeiten: Die Lernenden...
Setting	... verwenden Begrifflichkeiten aus der Gridworld ...machen allgemeine Aussagen über Eigenschaften der Gridworld ... interpretieren Zahlen in der Q-Tabelle mithilfe der Gridworld ... beschreiben, für welche Zahlen in den Q-Tabellen ihre Interpretationen in der Gridworld (nicht) gelten
Mathematischer Inhalt	... verwenden Begrifflichkeiten des mathematischen Inhalts ... stellen eine allgemeine Gleichung/Regel auf oder verändern sie ... führen Berechnungen mithilfe einer aufgestellten Gleichung/Regel aus ... überlegen, wann eine Gleichung/Regel gilt und wann nicht
Setting & mathematischer Inhalt	... verwenden Begrifflichkeiten der Gridworld und des mathematischen Inhalts ... beschreiben anhand der Gridworld, wann oder warum ihre Gleichung/Regel (nicht) gilt ... erklären die Wahl ihrer Gleichung/Regel oder die Wahl von Werten darin mithilfe der Gridworld

Tab. 4: Kategorien und beispielhafte Tätigkeiten

6. Ergebnisse

6.1 Computational Thinking in den Erarbeitungsprozessen

Zur Beantwortung von FF 1 betrachten wir die Fallzusammenfassung der drei Gruppen für die Erarbeitung der finalen Q-Tabelle und anschließend für die Q-Lernregel (siehe Kästen) und beschreiben anhand der Fallzusammenfassungen Prozesse, die auf Computational Thinking bei den Lernenden hindeuten. Um einzelne Aspekte der Fallzusammenfassungen zu referenzieren, nummerieren wir die Sätze.

6.1.1 Erarbeitungen der finalen Q-Tabelle

Gruppe 1 schließt zunächst aus der Betrachtung mehrerer Felder und Q-Werte, dass die Q-Werte nichtbegehrter Felder 0 sind und der höchste Q-Wert die optimale Richtung zum Ziel angibt [1a]. Anschließend stellen sie anhand verschiedener Q-Werte Vermutungen auf, wofür die konkreten Q-Werte stehen könnten [1b]. Sie stellen die Anzahl der Episoden auf 1 und folgern daraus anhand mehrerer Durchläufe, dass die Schrittwahl zufällig erfolgt [1c]. Sie erhöhen die Anzahl der Episoden wieder und diskutieren daran über Werte in der Tabelle, die sie mit bisherigen Vermutungen nicht erklären können [1d].

Gruppe 2 schließt aus der Betrachtung mehrerer Felder und Q-Werte, wann Q-Werte die Zahlen 0 bzw. 100 annehmen [2a]. Ebenfalls durch Betrachtung verschiedener Felder folgern sie, dass der Roboter immer der Richtung des höchsten Q-Wertes folgt und dass Q-Werte näher der Ladestation größer sind [2b]. Sie stellen anhand verschiedener Felder Vermutungen für die Bedeutung konkreter Q-Werte auf [2c]. Sie äußern zudem Vermutungen darüber, wie die Q-Werte bei 0 beginnen und iterativ verändert werden [2d]. Sie überprüfen ihre Vermutung, indem sie die Anzahl der Episoden gezielt reduzieren [2e].

Gruppe 3 äußert anhand verschiedener Felder Vermutungen, wofür die Q-Werte konkret stehen könnten [3a]. Anschließend reduzieren sie die Anzahl der Episoden und folgern damit anhand konkreter Felder, dass Q-Werte näher der Ladestation größer sind [3b]. Sie merken, dass sie anhand der Q-Tabellen einige ihrer vorherigen Vermutungen nicht verifizieren können und erhöhen daraufhin die Anzahl der Episoden wieder [3c]. Anhand einiger Felder nahe des Ziels erkennen sie, dass Werte niedriger sind für Aktionen, mit denen sich der Roboter von der Ladestation wegbewegt und Schritte den gleichen Q-Wert haben, wenn sie auf dem gleichen Feld enden [3d]. Sie finden jedoch Fälle, in denen ihre Vermutung nicht stimmt [3e].

Zwei der drei Gruppen beginnen mit der Schlussfolgerung allgemeiner Regeln aus einzelnen Q-Werten und den entsprechenden Feldern für markante Q-Werte, also 0 und 100 (1a, 2a). Derartige Hinweise auf Abstraktionsprozesse bzw. Mustererkennungen im Sinne des Computational Thinking (**abstraction**) finden sich in allen Gruppen während der Erarbeitungen. Dabei wird jeweils anhand (systematisch) ausgewählter Felder auf verschiedenste Aspekte der allgemeinen Funktionsweise des Q-Learnings

geschlossen, etwa auf das Zustandekommen konkreter Q-Werte (1b, 2c), die relative Änderung der Q-Werte in Abhängigkeit der Position (2b, 3d) und deren Bedeutung für den optimalen Weg (1a, 2b), sowie die Wahl der Schritte (1c).

Das Erstellen von Algorithmen (**algorithms**) lässt sich bei den Erarbeitungen der finalen Q-Tabelle eher selten identifizieren. Lediglich Gruppe 2 trifft Aussagen über das Vorgehen des Roboters während des Lernprozesses (2d), was als (Teil der) Algorithmusentwicklung verstanden werden könnte. Dabei beschreiben die Lernenden logische und geordnete Schritte, von denen sie glauben, dass diese während des Q-Learnings ablaufen, um die Q-Tabelle erklären zu können: alle Q-Werte starten bei 0, der Roboter läuft anschließend einen Weg ab und notiert danach für jedes überquerte Feld, ob das Ziel bei diesem Durchgang erreicht wurde. Dies wird wiederholt und – wie die Gruppe an anderer Stelle erwähnt – die Q-Werte ergeben sich als durchschnittlicher Anteil, wie oft das Ziel von einem Feld aus erreicht wurde.

Ein Zerlegen in Teilprobleme (**decomposition**) ist erkennbar, wenn die Lernenden die Q-Werte auf unterschiedlichen Ebenen untersuchen bzw. auf unterschiedlichen Ebenen Muster beschreiben: zunächst werden nur „auffällige“ Zahlen betrachtet (1a, 2a), später die relativen Größen der Zahlen durch Vergleiche oder die Betrachtung von Vorzeichen (1a, 2b, 3b, 3d) und die konkreten Werte (1b, 2c). Das gesamte Rekonstruktionsproblem – das Erklären der Werte in der Q-Tabelle – wird auf Teilprobleme unterschiedlicher Komplexität reduziert.

Gruppe 1 und Gruppe 3 zeigen zudem längere Phasen, in denen sie Fehler in ihren eigenen Ansätzen (i. d. R. Verallgemeinerungen bzw. Abstraktionen) suchen, finden und teilweise auch versuchen, diese zu beheben (**debugging**) (1d, 3c, 3e), beispielsweise indem sie die Episodenanzahl erhöhen.

Während sämtlicher Phasen nutzen die Lernenden in allen Gruppen die Simulation. Inwieweit es sich dabei um eine Automatisierung im Sinne des Computational Thinking (**automation**) handelt, ist jedoch fraglich. Die Lernenden lagern damit zwar Berechnungen der Q-Werte an den Computer aus, diese Auslagerung wird allerdings durch die Lernumgebung nahegelegt und erfordert daher vermutlich wenig aktives Zutun der Lernenden.

6.1.2 Erarbeitungen der Q-Lernregel

Gruppe 1 beginnt damit, durch Modifikation von Alpha und Gamma Vermutungen über den Einfluss von Alpha auf die Q-Werte aufzustellen [1a]. Sie belassen schließlich Alpha bei 1 und finden durch Nachrechnen heraus, dass die Formel für die finale Q-Tabelle die Q-Werte in der ersten Episode erklären kann [1b]. Sie betrachten daraufhin die Vorgehensweise in der zweiten Episode, insbesondere inwiefern die Schrittwahl zufällig erfolgt [1c]. Anschließend beschreiben sie anhand der dritten Episode, wie der Roboter während des Lernprozesses vorgeht [1d]. Sie variieren dann mehrfach den Wert für Alpha und versuchen, ihre Formel so anzupassen, dass sie die entstehenden Q-Werte erklären können, es treten jedoch immer wieder Unstimmigkeiten auf [1e]. Um diese zu lösen, stellen sie schließlich einen Teil der Formel als Funktion in GeoGebra dar [1f].

Gruppe 2 stellt ebenfalls zu Beginn durch Modifikation von Alpha und Gamma Vermutungen über die Nutzung von Alpha und Gamma innerhalb der Formel auf [2a]. Sie beschreiben zudem die Vorgehensweise des Roboters während des Lernprozesses [2b]. Sie merken anhand konkreter Werte, dass diese noch nicht stimmen, und schlagen eine Alternative vor [2c]. Sie suchen nun den Punkt in der Folge von Q-Tabellen, ab dem ihre Formel nicht mehr funktioniert [2d]. Sie betrachten daraufhin gezielt mehrere Fälle für Alpha und Gamma und passen ihren Term an [2e].

Gruppe 3 beginnt nach einer Exploration der Simulation damit, durch Modifikation von Alpha die Vermutung aufzustellen, dass die Q-Werte durch Multiplikation entstehen, sofern der Roboter noch keinen Weg zum Ziel kennt [3a]. Sie vermuten zusätzlich, dass die Formel für die finale Q-Tabelle verwendet wird, sobald der Roboter einmal das Ziel gefunden hat [3b]. Sie merken jedoch, dass ihre Idee nicht mit den Werten aus der Simulation zusammenpasst [3c]. Sie suchen daraufhin nach einer konkreten Zahl in den Q-Tabellen, die sie mit ihrer Formel nicht erklären können und modifizieren daraufhin ihre Formel und rechnen nach [3d]. Dieser Prozess wiederholt sich mehrfach, teils mit auch durch Hinweise der Lehrkraft [3e].

Alle drei Gruppen beginnen damit, den Parameter Alpha mehrfach zu variieren und daraus Schlussfolgerungen über den Einfluss von Alpha auf die Q-Werte zu ziehen (1a, 2a, 3a). Ihre Erkenntnisse diesbezüglich formalisieren sie – auch unter Zuhilfenahme von Variablen – indem sie aus beobachteten

Einzelfällen eine allgemeine Regel aufstellen (z. B. Gruppe 3: „Was ich mir auch vorstellen könnte, das ist Alpha hoch, sozusagen 0,8 hoch Gamma mal der Zahl. 0,8 hoch 1“) und somit Abstraktionsprozesse im Sinne des Computational Thinking (**abstraction**) vornehmen. Abstraktionsprozesse zu Funktionsweisen des Q-Learnings finden sich auch immer wieder im Laufe des Erarbeitungsprozesses, wenn die Lernenden mittels unterschiedlicher Parametereinstellungen verschiedene Ausgaben analysieren, um diese dann zu einer Regel zu verallgemeinern bzw. ihre bisherige Regel zu modifizieren. Beispielsweise stellt Gruppe 1 in (1e) eine Formel unter Verwendung von Variablen auf: „Dann wenn man das nochmal macht, wird das ja zu Alpha mal, Alpha mal das Q davor, alt, plus das q dann alt, dann ist das die gleiche Länge“.

Gruppe 1 und Gruppe 2 diskutieren zudem beide, wie der Roboter im Labyrinth vorgeht, insbesondere wie er die nächsten Schritte auswählt (1d, 2b). Hier sind Ansätze einer Beschreibung von Algorithmen erkennbar, indem die Lernenden logische, aufeinander aufbauende Schritte des Q-Learning-Algorithmus beschreiben (**algorithms**) – wenn auch teilweise sehr nah am Setting der Gridworld, recht umgangssprachlich und mit wenig mathematischer Terminologie: „Also sagen wir, wenn das jetzt hier 14 wäre [...]. Und dann merkt er aber, huh, huh, da komme ich ja auch lang. Und dann haben wir hier halt 46 von dem [...]“ (Gruppe 1).

Neben der Variation von Parametern mit dem Ergebnis einer allgemeinen Formel bzw. Regel (vgl. **abstraction**), verändern die Lernenden auch Parameter, um einzelne (Spezial-)Fälle zu untersuchen (1b, 2e). So verändert zum Beispiel Gruppe 2 (2e) gezielt Alpha zu Beginn der Modifikation einer Formel, um Veränderungen sichtbar zu machen – „Aber wenn wir Alpha auf 1 setzen, wird es vielleicht deutlicher“ – sie betrachten also zunächst einen Spezialfall und nehmen damit eine Zerlegung des Problems in Teilprobleme vor (**decomposition**). Gruppe 1 (1b) betrachtet nacheinander verschiedene Episoden und verändert so die Formel sukzessive. Zudem setzen sie Alpha zunächst auf 1, während sie es später variieren und damit weitere Fälle erschließen. In Gruppe 3 zeigt sich eine Zerlegung in Teilprobleme „im Nachhinein“, wenn die Lernenden merken, dass eine Formel für bestimmte Fälle gilt und für andere nicht (3b).

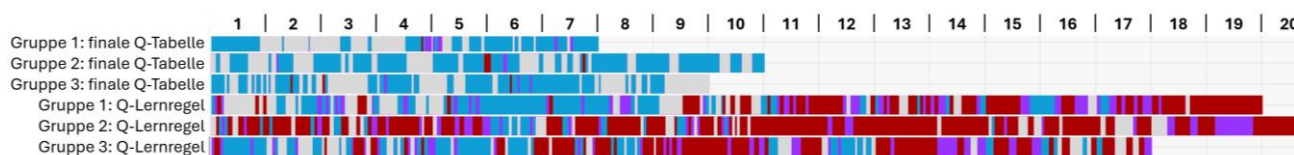


Abb. 10: Erarbeitungsprozess der drei Gruppen bei der finalen Q-Tabelle (obere drei Streifen) und bei der Q-Lernregel (untere drei Streifen); Blau: Setting, Rot: mathematischer Inhalt, Lila: Setting und mathematischer Inhalt, Grau: Sonstige Tätigkeiten; Abschnitte (1 bis 20) geben Seitenumbrüche im Transkript an.

Bei ihren abstrahierten Formeln oder Algorithmen stoßen die Lernenden immer wieder auf Fehler bzw. Grenzen und versuchen diese durch Modifikationen ihrer Ansätze zu beheben (**debugging**). Solche Prozesse treten in allen Gruppen auf (1e, 2c, 2d, 3c, 3d, 3e). Interessant ist insbesondere, dass Gruppe 2 und Gruppe 3 jeweils aktiv nach Punkten suchen, in denen Fehler auftreten könnten, um ihre Formel zu „testen“ (S3: „[...] Aber für den Anfang passt ja alles. Nur am Ende machst du dann irgendwas.“ S4: „Ja, ich möchte mal rausfinden, ab welchem Punkt es nicht mehr stimmt“). Gruppe 1 zieht zum Beheben von Fehlern GeoGebra als externes digitales Werkzeug heran (1f). Die Lernenden lagern damit Arbeitsprozesse (in diesem Fall eine Visualisierung) bewusst an den Computer aus (**automation**). Die anderen Gruppen lagern ebenfalls Berechnungen an den Computer aus, indem sie die Simulation bedienen und sich damit die Q-Tabellen ausgeben lassen. Allerdings sind diese Automatisierungsprozesse eher durch die Lernumgebung bedingt als durch Initiative der Lernenden.

6.2 Zusammenspiel zwischen Setting und mathematischem Inhalt

Zur Diskussion von Forschungsfrage 2 wurden die Kodierungen der Aussagen in einem Diagramm visualisiert (vgl. Abb. 10), um den Verlauf zu veranschaulichen. Die oberen drei Streifen zeigen die Erarbeitung der finalen Q-Tabelle der drei Zweiergruppen, die unteren drei Streifen die Erarbeitung der Q-Lernregel. Die vertikalen Unterteilungen (nummeriert mit 1 bis 20) signalisieren einen Seitenumbruch im Transkript.

Zunächst lässt sich feststellen, dass alle vier Kategorien in den Bearbeitungen der Lernenden auftreten. Zudem sind Wechsel zwischen den Kategorien erkennbar. Bei der **Erarbeitung der finalen Q-Tabelle** fällt zunächst auf, dass alle drei Gruppen ein ähnliches Bild zeigen: Es wird vorwiegend auf das Setting zurückgegriffen. Ein Beispiel dafür zeigt folgende Konversation:

Transkript 1 – Beispiel: Setting

S1: [...] Welche Richtung ist das hier? Da hat er ja 100. Vielleicht nach links?

S2: Ja, weil das ist ja der optimalste Schritt. Das ist wahrscheinlich dann von E5.

S1: Je höher die Zahl, desto besser der Track?

(Gruppe 1, finale Q-Tabelle, S. 1)

S1 interpretiert hier die Größe der Zahl in der Tabelle als Qualität der jeweiligen Richtung bzw. des jeweiligen Weges („Track“), interpretiert also Zahlen in der Q-Tabelle mithilfe der Gridworld (vgl. „Setting“ in Tab. 4). Insbesondere nutzt S1 mit „Richtung“ und „Schritt“ Begrifflichkeiten aus der Gridworld (vgl. Tab. 3). Derartige Konversationen treten bei der Erarbeitung der finalen Q-Tabelle häufiger auf, resultieren aber selten in einem Bezug auf den mathematischen Inhalt, etwa dem Versuch, eine allgemeine Rechenvorschrift aufzustellen. Bezüge zum mathematischen Inhalt (*Mathematischer Inhalt* oder *Setting & mathematischer Inhalt*) hingegen treten nur punktuell und vereinzelt, je nach Gruppe zu unterschiedlichen Zeitpunkten, auf. Die Kombination aus *Setting & mathematischer Inhalt* kommt beispielsweise dann vor, wenn beim Arbeiten mit dem Setting punktuell auf Begrifflichkeiten des mathematischen Inhalts zurückgegriffen wird:

Transkript 2 – Beispiel: Setting & mathematischer Inhalt

S2: Weil da weiß er [der Roboter] ja dann schon, was da ist, wenn er runtergeht. Vielleicht kriegt er einen Reward, wenn er ein Feld aufdeckt, wo er noch nicht war. Unter anderem. Weil der will ja auch nicht wieder zurückgehen.

S1: Ja, wenn er zurück geht wird er wahrscheinlich härter bestraft. [...]

(Gruppe 1, finale Q-Tabelle, S. 4-5)

Der Rückgriff auf die Begriffe „Reward“ bzw. „bestraft“ zeigt, dass hier Terminologie des Q-Learnings verwendet wird (vgl. Tab. 3). Sie wird zur Beschreibung eines im Setting verankerten Sachverhalts herangezogen.

In den Phasen, in denen weder auf das Setting noch auf den mathematischen Inhalt zurückgegriffen wird, arbeiten die Lernenden hauptsächlich mit der Simulation selbst, indem Sie beispielsweise explorativ Parameter in der Simulation ändern oder Zahlen ablesen.

Bei der **Erarbeitung der Q-Lernregel** treten mehr Phasen mit Bezug zum mathematischen Inhalt auf. Diese werden tendenziell gegen Ende der Bearbeitung häufiger. Es lassen sich hier jedoch auf deskriptiver Ebene Gruppenunterschiede identifizieren: Gruppe 2 arbeitet bereits zu Beginn vorwiegend mit Aussagen zum mathematischen Inhalt, bezieht sich dann kurzzeitig mehr auf das Setting (S. 6) und anschließend wieder auf den mathematischen Inhalt. Bei den beiden anderen Gruppen ist eine Tendenz vom Setting zum mathematischen Inhalt erkennbar. Dabei fällt insbesondere Gruppe 3 durch ihren frühen Bezug zum Setting auf. Die Lernenden aus Gruppe 3 beginnen – wie Gruppe 2, was ihren starken Bezug zum mathematischen Inhalt erklärt – mit der Gleichung für die finale Q-Tabelle, verwerfen diese aber recht schnell, sodass sie anschließend mit dem Setting weiterarbeiten.

Betrachtet man die Gruppen jeweils über den gesamten Erarbeitungsprozess hinweg (d. h. finale Q-Tabelle und Q-Lernregel), so zeichnet sich in allen Gruppen eine tendenzielle Entwicklung vom Rückgriff auf das Setting hin zum mathematischen Inhalt ab. Dabei tritt die Kategorie *Setting & mathematischer Inhalt* zwar immer wieder auf, allerdings seltener als die anderen Kategorien. Ein Beispiel für das Auftreten dieser Kategorie bei der Erarbeitung der Q-Lernregel findet sich an Übergängen vom Setting zum mathematischen Inhalt wie in folgendem Beispiel:

Transkript 3 – Beispiel: Übergang vom Setting zum mathematischen Inhalt

S4: Am Anfang geht der nach unten runter. Am Anfang geht er auf dem unten und bleibt am selben.

S3: Minus 8 mal 0 ist gleich 0. Ypsilon ist 0,8. Reward ist minus 10 mal Energieverlust.

S4: Hier geht er nochmal rauf. Dummes Teil.

S3: Äh 9,6, Der erste war quasi... R0 ist minus 8 in dem Fall. [...] R0 ist ja der Reward von dem Action-Dings vorher. Heißt, wenn das Reward war, müsste R0 minus 8 sein. Plus 0,8 hoch 2-mal minus 10.

(Gruppe 2, Q-Lernregel, S. 1)

S4 vollzieht zunächst das Setting nach, woraufhin S3 einen Term unter Verwendung von Begrifflichkeiten der Gridworld („Energieverlust“) aufstellt (*Setting & mathematischer Inhalt*). S4 greift dann nochmal auf das Setting zurück. S3 jedoch bezieht sich nun stärker auf den mathematischen Inhalt und stellt einen Term unter Verwendung von mathematischen Begrifflichkeiten („Reward“, „Action“) ohne verbalisierten Bezug zur Gridworld auf, sodass sich in diesem Transkriptausschnitt alle drei Kategorien wiederfinden.

In den Phasen, in denen weder auf das Setting noch auf den mathematischen Inhalt zurückgegriffen wird, ist wie bereits bei der finalen Q-Tabelle ein starker Fokus auf die Simulation erkennbar.

7. Diskussion

Wir diskutieren zunächst beide Forschungsfragen mit Bezug zu Computational Thinking und der Nutzung des Settings, um diese anschließend hinsichtlich ihrer Bedeutsamkeit für AI Literacy und für die mathematische Bildung zu erörtern.

Computational Thinking im Erarbeitungsprozess

In den Fallzusammenfassungen finden sich Hinweise auf alle fünf Komponenten des Computational Thinking. Dabei ist allerdings anzumerken, dass die Komponenten nicht notwendigerweise vollständig abgebildet werden.

Deutlich wird, dass **Abstraction** bei den hier dargestellten Bearbeitungsprozessen eine wichtige Rolle spielt. Die Lernenden analysieren die ausgegebenen Daten des KI-Algorithmus (vgl. Abb. 8, Abb. 9) und suchen diese systematisch nach Mustern ab. Diese resultieren in Formeln bzw. Regeln, welche die Funktionsweise des Q-Learnings abbilden. Diese und ähnliche mit *Abstraction* verbundene Prozesse ließen sich in beiden Arbeitsphasen aller Gruppen erkennen. Dies lässt sich unter Umständen damit erklären, dass die Fragestellungen an die Schüler:innen „*Welche Erkenntnisse könnt ihr anhand der Q-Tabelle bzw. Q-Tabellen der einzelnen Schritte gewinnen?*“ eine gewisse Abstraktion bzw. Verallgemeinerung nahelegen, um diese als „Erkenntnisse“ zu präsentieren. Zudem werden Abstraktionen durch die

Gestaltung der Lernumgebung nahegelegt, welche eine Variation mehrerer Parameter ermöglicht. Die Lernenden können somit schnell viele Input-Output-Paare erzeugen, um anhand derer Muster zu erkennen (vgl. auch Beispiele in Shute et al., 2017).

In Bezug auf die Komponente **Algorithms** ist anzumerken, dass bei der finalen Q-Tabelle lediglich das Ergebnis bzw. *Produkt* des Q-Learning-Algorithmus ersichtlich ist, was das Erschließen eines Algorithmus, also eines *Prozesses*, nicht nahelegt. Lediglich eine Gruppe geht trotzdem ausführlicher auf die Schritte während des Lernprozesses ein, indem sie die Anzahl der Episoden auf 1 stellt, was Aussagen über Algorithmen ermöglicht. Bei der Arbeit mit der Folge an Q-Tabellen ist zu bedenken, dass die Lernenden in den Erarbeitungen nicht in erster Linie Algorithmen entwickeln, um einen optimalen Weg des Roboters zum Ziel zu finden; vielmehr rekonstruieren sie den Algorithmus, der eine Folge von Q-Tabellen „erzeugt“. Wenn die Lernenden nun einen Teil dieses Algorithmus (z. B. eine Beschreibung der Wahl des nächsten Feldes durch den Roboter) beschreiben, so geben sie logische, geordnete Schritte zur Lösung dieses Rekonstruktionsproblems wieder, insofern als dass sie einen Teil der Lösung darlegen. Der Begriff des Algorithmus ließe sich auch anders interpretieren, nämlich als logische, geordnete Schritte, die die Lernenden ausführen, um zu einer Aussage über den Q-Learning-Algorithmus zu kommen (also ein Algorithmus für das Vorgehen der Lernenden, vgl. Beispiel in Shute et al., 2017). Während wir hierzu keine explizite Aussage treffen können, zeigen sich solche Zerlegungen in Teilschritte mitunter in der Komponente *Decomposition*. Des Weiteren lässt sich festhalten, dass die Entwicklung von Algorithmen durch die Lernenden teils Folge einer Abstraktion aus mehreren beobachteten Vorgängen ist. Ein einfaches Beispiel hierfür ist das mehrfache Durchspielen der Simulation mit Betrachtung verschiedener Episoden, um daraus zu schließen, dass die Schrittwahl zufällig erfolgt. Hier stellt sich somit auch die Frage nach der Überlappung von *Abstraction* und *Algorithms*.

Auch das **Debugging** steht im Zusammenhang mit den seitens der Lernenden entwickelten abstrahierten Algorithmen und Formeln bzw. Regeln. Hier findet ein Abgleich der durch den Q-Learning-Algorithmus erzeugten Daten mit den Werten, die laut des eigenen Algorithmus oder der eigenen Hypothesen entstehen müssten, statt. Inwieweit Fehler, die in diesem Fall Abweichungen in den Daten darstellen, auch behoben werden können, ist unterschiedlich und wurde nicht näher untersucht. Es ist jedoch

ersichtlich, dass die Lernenden gezielt versuchen, Fehler zu identifizieren, zu isolieren, zu reproduzieren und ihre Vermutungen systematisch zu testen (Weintrop et al., 2016). Hier zeigt sich auch ein Zusammenspiel mit der Komponente *Abstraction*: Abstraktionsprozesse werden auf Basis von Datenpunkten vorgenommen, dann anhand neuer Datenpunkte evaluiert und im Falle eines Fehlers wird die abstrahierte Formel bzw. Regel modifiziert oder verworfen.

Hinsichtlich der **Decomposition** findet zwar eine Zerlegung eines Problems statt, allerdings handelt es sich bei dem Problem nicht um eine authentische, realweltliche Problemstellung wie etwa bei Ng, Su et al. (2024). Eine Zerlegung erfolgt daher nicht, indem etwa bestimmte Teilbereiche eines realweltlichen Problems getrennt voneinander betrachtet werden, sondern indem bestimmte (Spezial-)Fälle von Parametern separat untersucht werden. Dieses Vorgehen wird durch die Gestaltung der Lernumgebung in gewisser Weise nahegelegt, welche eine Variation mehrerer Parameter ermöglicht. Auf Basis solcher gezielter Zerlegungen in „functional elements that collectively comprise the whole system/problem“ (Shute et al., 2017, S. 153) können dann wieder abstrahierte Aussagen über das System getroffen werden.

Die digitale Umgebung, in die der Algorithmus eingebettet ist, um Daten zu erzeugen, legt die **Automation** unweigerlich nahe. Es ergeben sich darüber hinaus einzelne Situationen, in denen die Lernenden auf zusätzliche digitale Hilfsmittel zurückgriffen, um Visualisierungen anfertigen und somit implizit Berechnungen durchführen zu lassen. Das Fehlen einer eigenen Implementierung des Algorithmus (anders als z. B. bei Brennan & Resnick, 2012) schmälert hier die bewusste Anwendung von Automation und wirft die Frage auf, inwieweit hier Computational Thinking oder doch besser Algorithmic Thinking – welches von Kadijevich (2018) als Computational Thinking ohne *Automation* konzeptualisiert wird – betrachtet werden sollte.

Insgesamt ließen sich bei der Erarbeitung des Q-Learning-Algorithmus Hinweise auf alle Komponenten des Computational Thinking finden, wobei diese mit unterschiedlichen Einschränkungen bzw. spezifischen Nuancen einhergehen. Dabei ist zu beachten, dass die Analyse auf den (auf Aussagen basierenden) Fallanalysen beruhen, die uns nur bedingt Einblicke in Denkprozesse geben. Daher ist es denkbar, dass einzelne Komponenten gewichtigere Anteile einnehmen, als es uns möglich ist, zu erfassen. Am

bedeutsamsten erscheint jedoch die Komponente *Abstraction*. Wir können zwar nur bedingt quantitative Aussagen über die Häufigkeiten der Komponenten treffen, jedoch lassen sich Abstraktionsprozesse – wie bereits erläutert – auch bei der Beschreibung von Algorithmen identifizieren. Zudem bedarf es Prozessen der *Decomposition*, um verschiedene Fälle für darauffolgende Abstraktionsprozesse zu generieren. Dies wirft die Frage nach der Trennschärfe bzw. Abhängigkeit der verschiedenen Komponenten des Computational Thinking bei der Bearbeitung des betrachteten Rekonstruktionsproblems auf. Die Bedeutsamkeit von *Abstraktion* könnte daher rühren, dass der Fokus auf der Erarbeitung eines bestimmten mathematischen Algorithmus liegt. Dieser soll anhand des konkreten Vorgehens des Roboters in der Gridworld – repräsentiert durch die Ausgaben der Simulation – erschlossen werden. Welche Rolle die Gridworld als Setting dabei spielt, wird im Folgenden eingehender diskutiert.

Rolle des Settings im Erarbeitungsprozess

Für die Umgebung wurde ein fiktives Setting gewählt, das zum einen eine Art spielerischen Zugang ermöglicht und zum anderen ein Erschließen durch Weltwissen und Vorerfahrungen erlauben soll. In den Erarbeitungsprozessen lässt sich eine Tendenz dahingehend erkennen, dass zunächst vom Setting ausgegangen und an späteren Stellen eher auf mathematischer Ebene gearbeitet wird. Der zunächst verstärkte Fokus auf das Setting bei der Erarbeitung der finalen Q-Tabelle ist insofern interessant, als dass sich die Lernenden bereits einen Tag vor den hier untersuchten Arbeitsphasen mit dem Setting vertraut gemacht haben und auch das Konzept sowie die zugehörigen Begrifflichkeiten des Reinforcement Learning kennengelernt haben. Dennoch greifen sie verstärkt auf das Setting zurück, wenn sie versuchen, mathematische Zusammenhänge zu erschließen. Das deutet auf das Potenzial hin, mathematische Zusammenhänge auf Erfahrungen und Wissen über das Setting zu erschließen.

Anders verhält es sich mit der Erarbeitungsphase zur Q-Lernregel. Hier beginnen die Lernenden eher mathematisch bzw. inhaltlich zu arbeiten. Eine Erklärung könnte die Präsentation der Q-Lernregel nach der ersten Erarbeitung liefern. Dies liegt insbesondere nahe, da alle Gruppen zunächst mit dieser Formel die zweite Erarbeitungsphase beginnen und sie dann modifizieren oder verwerfen. Die Präsentation eines (ähnlichen) mathematischen Inhalts könnte somit hilfreich sein, um schnell mathematische Überlegungen bezüglich eines anderen Inhalts

anzustoßen. Hierbei deutet sich an, dass die Nutzung des Settings bzw. der Fokus auf den mathematischen Inhalt in den eher offenen Arbeitsphasen zu einem gewissen Grad durch eingängliche Impulse beeinflusst werden kann. Es zeigt sich zudem, dass in der Erarbeitung der Q-Lernregel häufiger Prozesse des Debuggings auftreten, was gegebenenfalls durch den stärkeren Fokus auf mathematische Inhalte bedingt sein könnte.

Bemerkenswert ist, dass auch bei der Tendenz vom Setting zum mathematischen Inhalt hin verhältnismäßig wenige Phasen erkennbar sind, in denen die beiden Bereiche explizit verknüpft werden. Dies könnte einerseits dafürsprechen, dass die beiden Bereiche von den Lernenden tatsächlich nur selten verbunden wurden. Andererseits ist es denkbar, dass die Verknüpfung von den Lernenden zwar durchgeführt, allerdings nicht verbalisiert wurde.

Es zeigt sich somit insgesamt, dass das Setting von den Lernenden genutzt wird, aber gleichzeitig vielfältige Abstraktionen vorgenommen werden, sowohl direkt anhand des Settings (etwa zur Beschreibung von Algorithmen) als auch mithilfe von Variablen auf einer eher formal-mathematischen Ebene. Wenngleich wir keinen Vergleich zu einer Erarbeitung ohne konkretes Setting (z. B. nur mit der Simulation) ziehen können, so scheint das Vorliegen eines Settings die Lernenden zumindest nicht daran zu hindern, zu abstrahieren und mathematische Strukturen (hier Formeln) zu entwickeln. Auf Basis der Ergebnisse lässt sich daher vermuten, dass es auch bei einer mathematischen Aufarbeitung von KI-Algorithmen – welche über ein grundlegendes Verständnis der Funktionsweise von KI deutlich hinaus geht – ein Setting gewählt werden kann, das von den Lernenden als Unterstützung auch tatsächlich herangezogen wird. Die zunehmende Abstraktion durch die Lernenden könnte den Ansatz eines fiktiven Settings bekräftigen, bei der das Ziel nicht die Lösung eines realweltlichen Problems ist (vgl. Ng et al., 2024), sondern gewissermaßen die Loslösung von diesem Setting. Hierzu ist sind jedoch gegebenenfalls Impulse hinsichtlich ähnlicher mathematischer Inhalte hilfreich, wie sich in den Daten zwischen der finalen Q-Tabelle und der Q-Lernregel andeutet.

Bedeutung der Lerneinheit hinsichtlich AI Literacy

Bei der Erarbeitung des Q-Learning-Algorithmus aus mathematischer Perspektive kann erfahren werden, wie „Lernprozesse“ einer KI durch Daten und eine Modellbildung durch den Algorithmus (Abbildung der Gridworld in einer Q-Tabelle) erfolgen kann. Diese Prozesse werden von Touretzky et al. (2019)

angesprochenen und für Schüler:innen als relevante und zentrale Ideen hinsichtlich KI erachtet. Allerdings ist hierbei anzumerken, dass die Prüfung des Verständnisses der Funktionsweise nicht expliziter Teil der Untersuchung war. Vielmehr wurde untersucht, inwieweit beim Erarbeitungsprozess auf das Setting zurückgegriffen wurde.

Es ist anzumerken, dass der Fokus der Erarbeitungen ausschließlich auf dem Verständnis der Funktionsweise von KI lag, wobei ethische und gesellschaftliche Fragestellungen bzw. Auswirkungen von KI (Ali et al., 2021; Lintner, 2024; Ng et al., 2021) nicht thematisiert wurden. Eine Diskussion solcher Aspekte wäre im Anschluss an die vorliegende Lerneinheit aber denkbar, z. B. durch die Thematisierung der Folgen von Reinforcement Learning in großen Sprachmodellen (Havrilla et al., 2024) wie etwa eine mögliche Verstärkung von Bias. Es lässt sich durchaus kritisch hinterfragen, inwieweit dazu ein tiefgehendes, mathematisches Verständnis konkreter Algorithmen notwendig ist oder ob ein Verständnis grundlegender Funktionsweisen des Reinforcement Learning ausreichend ist. Hierzu sei aber erwähnt, dass die Facetten von AI Literacy zielgruppenspezifisch unterschiedlich sind (Chee et al., 2025). Bei den Teilnehmenden handelte es sich um mathematisch interessierte Schüler:innen, für die durchaus ein tieferes Verständnis möglich und hilfreich sein könnte. Ein mathematisches Verständnis von KI-Algorithmen kann zudem dazu beitragen, nicht das Gefühl zu hinterlassen, dass diese im Grunde weiterhin eine „Blackbox“ darstellen (Ng, Xinyu, et al., 2024). Das Verständnis dafür, dass die „Blackbox“ lediglich aus mathematischen Berechnungen besteht, kann dann wiederum eine kritisch-sachliche Evaluation von KI unterstützen, welche zentral für AI Literacy ist (Lau-pichler et al., 2023; Long & Magerko, 2020).

Implikationen des Lernens über KI für die mathematische Bildung

Im vorangegangenen Abschnitt wurde aufgezeigt, inwieweit die Thematisierung des Q-Learning-Algorithmus einen Beitrag zu AI Literacy leisten kann. Nun kann jedoch die Frage gestellt werden, inwieweit sich die hier vorgestellte Einheit tatsächlich im Kontext des Mathematikunterrichts bzw. der Mathematikdidaktik verorten lässt – oder ob diese eher aus dem Bereich der Informatik stammt – und inwieweit die Thematik einen Mehrwert für die mathematische Bildung aufweist.

Konzeptuell weisen die Komponenten des Computational Thinking Parallelen zu mathematischen Konzepten wie dem Problemlösen (Polyá, 1945) oder

dem algebraischen Denken (z. B. Kieran et al., 2016) auf. In unserer Untersuchung konnte festgestellt werden, dass sich gerade *Abstraction*, die schwerpunktmäßig identifizierte Komponente des Computational Thinking, in den Erarbeitungsprozessen der Lernenden wiederfindet. Daher könnte das Erarbeiten von Algorithmen in der Form, wie es in dieser Studie erfolgte, auch einen Beitrag dazu leisten, das algebraische Denken zu fördern. Darüber hinaus stellen die Optimierung und die Approximation fundamentale Ideen der Mathematik dar (vgl. Humenberger & Reichel, 1995), die beim Q-Learning eine zentrale Rolle spielen, indem iterativ Q-Werte angepasst und dem optimalen Wert angenähert werden.

Daher ist festzuhalten, dass die Thematisierung von KI im Mathematikunterricht nicht nur auf ein Verständnis von KI zielt, sondern auch der mathematischen Bildung dienen kann. Gleichzeitig kann ein Beitrag zur allgemeinen Bildung der Lernenden geleistet werden.

8. Limitationen

Die Studie und deren Erkenntnisse sind mit einigen Einschränkungen verbunden. Erstens ermöglichen die Daten zwar die Erfassung von verbalen Äußerungen, jedoch nicht von Denkprozessen. Es lassen sich deshalb nur Hinweise auf Computational Thinking bzw. einen Bezug zum Setting identifizieren, nicht aber Belege dafür. Das seltene Auftreten der Kategorie *Setting & mathematischer Inhalt* resultiert daher nicht notwendigerweise daraus, dass die Lernenden tatsächlich keine Vernetzung zwischen den beiden Bereichen herstellen – sie verbalisieren eine solche Verbindung nur nicht. Eine methodische Einschränkung zeigt sich darin, dass für die Komponenten des Computational Thinking keine systematische Kodierung vorgenommen wurde, sodass eine subjektive Färbung der Ergebnisse möglich ist. Eine solche Kodierung könnte zudem Einblick in die Häufigkeit der einzelnen Komponenten sowie in Zusammenhänge zwischen dem Bezug auf das Setting und den Komponenten des Computational Thinking bieten. Eine weitere Limitation stellt die Abhängigkeit der Ergebnisse von der konkreten Lerneinheit dar. So wird beispielsweise bei der Betrachtung von Prozessen des Computational Thinking der mögliche Einfluss der Gestaltung der Simulationen auf die Erarbeitungsprozesse bereits deutlich. Die sonstigen Arbeitsphasen wurden in unserer Analyse – abgesehen davon, dass vorwiegend mit der Simulation gearbeitet wird – nicht näher betrachtet. Eine solche Betrachtung erscheint insbesondere mit Blick auf die finale Q-Tabelle lohnenswert, bei deren Erarbeitung

viele Wechsel zwischen Setting und sonstigen Arbeitsphasen auftreten. Zudem sind die Erkenntnisse der Untersuchung aufgrund der kleinen Stichprobe und der Beschränkung auf ein konkretes Thema (Q-Learning) nicht verallgemeinerbar. Für andere Themen der KI müsste überprüft werden, inwiefern Computational Thinking bzw. der Bezug zum Setting in ähnlicher Weise auftreten. Zuletzt sollte einschränkend erwähnt werden, dass kein Vergleich zwischen der Erarbeitung mit dem Setting und einer rein innermathematischen Erarbeitung vorliegt. Es ist deshalb nicht klar, welchen Einfluss das Setting auf die Erarbeitung der Lernenden hatte und inwieweit er tatsächlich zu einem Verständnis des mathematischen Inhalts beigetragen hat.

9. Zusammenfassung

AI Literacy und dem damit verbundenen Computational Thinking kommt in einer zunehmend von KI beeinflussten Welt eine große Bedeutung zu. Dabei spielt unter anderem das Verständnis der Funktionsweisen von KI eine wichtige Rolle. In diesem Beitrag wurde der Frage nachgegangen, wie KI-Algorithmen aus mathematischer Perspektive im Sinne eines Beitrags zu AI Literacy mit Schüler:innen thematisiert werden können. Besonderer Wert in diesem Beitrag wurde auf die Rahmung durch ein fiktives Setting gelegt, sodass eine Möglichkeit zur (mathematischen) Erschließung des Algorithmus angeboten wird. Die Erarbeitungsprozesse der Lernenden wurden hinsichtlich des Bezugs auf das Setting und möglicher Hinweise auf Prozesse des Computational Thinking untersucht.

Unsere Ergebnisse geben Hinweise darauf, dass Lernende auch bei einer Thematisierung des KI-Algorithmus aus mathematischer Perspektive das fiktive Setting bei der Erarbeitung nutzen. Zudem zeigen sich während der Erarbeitung des KI-Algorithmus *Abstraction*, *Algorithms*, *Decomposition*, *Debugging* und *Automation* als typische Komponenten des Computational Thinking, wenn auch teils in spezifischen Nuancen. Hierbei scheint insbesondere die Komponente *Abstraction* zentral zu sein, was auf Potentiale des Lernens über KI für die mathematische Bildung über AI Literacy hinaus hinweist. Zudem deutet die zunehmend seltenere Nutzung des Settings über die Erarbeitungsprozesse hinweg darauf hin, dass es den Lernenden gelingt, trotz des anschaulichen Settings abstrakte mathematische Sachverhalte zu thematisieren, was durch die Bedeutung von *Abstraction* unterstrichen wird.

Insgesamt kann festgehalten werden, dass durch das Aufgreifen des Themas KI im Mathematikunterricht

nicht nur ein allgemeiner gesellschaftlich hoch relevanter Beitrag zur Bildung geleistet wird, sondern dadurch auch mathematische Denkprozesse gefördert werden können.

Literatur

- Ali, S., DiPaola, D., Lee, I., Hong, J. & Breazeal, C. (2021). Exploring Generative Models with Middle School Students. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–13. <https://doi.org/10.1145/3411764.3445226>
- Almatrafi, O., Johri, A. & Lee, H. (2024). A systematic review of AI literacy conceptualization, constructs, and implementation and assessment efforts (2019–2023). *Computers and Education Open*, 6, Article 100173. <https://doi.org/10.1016/j.caeo.2024.100173>
- Biehler, R. & Fleischer, Y. (2024). Vom Zufall zur optimalen Strategie: Bestärkendes maschinelles Lernen am Beispiel Bauernschach. *mathematik lehren*, 244, 19–23.
- Brennan, K. & Resnick, M. (2012). New Frameworks for Studying and Assessing the Development of Computational Thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*. <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>
- Brennan, R. L. & Prediger, D. J. (1981). Coefficient Kappa: Some Uses, Misuses, and Alternatives. *Educational and Psychological Measurement*, 41(3), 687–699. <https://doi.org/10.1177/001316448104100307>
- Celik, I. (2023). Exploring the Determinants of Artificial Intelligence (AI) Literacy: Digital Divide, Computational Thinking, Cognitive Absorption. *Telematics and Informatics*, 83, Article 102026. <https://doi.org/10.1016/j.tele.2023.102026>
- Chee, H., Ahn, S. & Lee, J. (2025). A Competency Framework for AI Literacy: Variations by Different Learner Groups and an Implied Learning Pathway. *British Journal of Educational Technology*, 56, 2146–2182. <https://doi.org/10.1111/bjet.13556>
- Chiu, T. K. F., Ahmad, Z., Ismailov, M. & Sanusi, I. T. (2024). What are artificial intelligence literacy and competency? A comprehensive framework to support them. *Computers and Education Open*, 6, Article 100171. <https://doi.org/10.1016/j.caeo.2024.100171>
- Cuny, J., Snyder, L. & Wing, J. M. (2010). *Demystifying computational thinking for non-computer scientists*. [unveröffentlichtes Manuskript] <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- Dauscher, P. (2019). Künstliche Intelligenz selbst programmiert: Eine Unterrichtsreihe zum Bestärkenden Lernen (Reinforcement Learning). *MNU*, 06.2019, 472–477.
- Fleischer, Y. & Biehler, R. (2025). Exploring students' constructions of data-based decision trees after an introductory teaching unit on machine learning. *ZDM – Mathematics Education*, 57(1), 153–173. <https://doi.org/10.1007/s11858-025-01663-6>
- Havrilla, A., Du, Y., Raparthy, S. C., Nalmpantis, C., Dwivedi-Yu, J., Zhuravinskiy, M., Hambro, E., Sukhbaatar, S. & Raileanu, R. (2024). *Teaching Large Language Models to Reason with Reinforcement Learning* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2403.04642>

- Humenberger, J. & Reichel, H.-C. (1995). *Fundamentale Ideen der angewandten Mathematik und ihre Umsetzung im Unterricht*. BI-Wiss.-Verl.
- ISB. (2025). *Lehrplan PLUS Bayern Gymnasium*. <https://www.lehrplanplus.bayern.de/schulart/gymnasium>
- Kadijevich, D. M. (2018). A Cycle of Computational Thinking. In B. Trebinjac & S. Jovanović (Hrsg.), *Proceedings of the Ninth International Conference on e-learning* (S. 75–77). Metropolitan University. <https://protect-au.mimecast.com/s/78-iCMwvygsq3qw6ghYidai?domain=elearning.metropolitan.ac.rs>
- Kadijevich, D. M. (2023). Computational/algorithmic thinking in school mathematics. In A. Hujdurović, K. Kutnar, D. Marušič, Š. Miklavič, T. Pisanski & P. Šparl (Hrsg.), *European Congress of Mathematics* (1. Aufl., S. 749–769). EMS Press. <https://doi.org/10.4171/8ecm/40>
- Kadijevich, D. M., Stephens, M. & Rafiepour, A. (2023). Emergence of Computational/Algorithmic Thinking and Its Impact on the Mathematics Curriculum. In Y. Shimizu & R. Vithal (Hrsg.), *Mathematics Curriculum Reforms Around the World* (S. 375–388). Springer International Publishing. https://doi.org/10.1007/978-3-031-13548-4_23
- Khamis, M. A. & Gomaa, W. (2014). Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Engineering Applications of Artificial Intelligence*, 29, 134–151. <https://doi.org/10.1016/j.engappai.2014.01.007>
- Kieran, C., Pang, J., Schifter, D. & Ng, S. F. (2016). *Early Algebra: Research into its Nature, its Learning, its Teaching*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-32258-2>
- KMK. (2024). *Handlungsempfehlung für die Bildungsverwaltung zum Umgang mit Künstlicher Intelligenz in schulischen Bildungsprozessen*. https://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/2024/2024_10_10-Handlungsempfehlung-KI.pdf
- Kong, S.-C., Cheung, M.-Y. W. & Tsang, O. (2024). Developing an artificial intelligence literacy framework: Evaluation of a literacy course for senior secondary students using a project-based learning approach. *Computers and Education: Artificial Intelligence*, 6, Article 100214. <https://doi.org/10.1016/j.caeai.2024.100214>
- Kuckartz, U. (2016). *Qualitative Inhaltsanalyse. Methoden, Praxis, Computerunterstützung* (3., überarbeitete Aufl.). Beltz.
- Kuckartz, U. (2018). *Qualitative Inhaltsanalyse. Methoden, Praxis, Computerunterstützung*. (4. Aufl.). Beltz Juventa.
- Kuckartz, U. (2019). *Qualitative Text Analysis: A Systematic Approach*. In G. Kaiser & N. Presmeg (Hrsg.), *Compendium for Early Career Researchers in Mathematics Education* (S. 181–197). Springer. https://doi.org/10.1007/978-3-030-15636-7_8
- Landis, J. R. & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1), 159–174. <https://doi.org/10.2307/2529310>
- Laupichler, M. C., Aster, A., Haverkamp, N. & Raupach, T. (2023). Development of the “Scale for the assessment of non-experts’ AI literacy” – An exploratory factor analysis. *Computers in Human Behavior Reports*, 12, Article 100338. <https://doi.org/10.1016/j.chbr.2023.100338>
- Lehmann, T. H. (2023). Using algorithmic thinking to design algorithms: The case of critical path analysis. *The Journal of Mathematical Behavior*, 71, Article 101079. <https://doi.org/10.1016/j.jmathb.2023.101079>
- Lin, X.-F., Zhou, Y., Shen, W., Luo, G., Xian, X. & Pang, B. (2024). Modeling the structural relationships among Chinese secondary school students’ computational thinking efficacy in learning AI, AI literacy, and approaches to learning AI. *Education and Information Technologies*, 29(5), 6189–6215. <https://doi.org/10.1007/s10639-023-12029-4>
- Lintner, T. (2024). A systematic review of AI literacy scales. *Npj Science of Learning*, 9(1), Article 50. <https://doi.org/10.1038/s41539-024-00264-4>
- Long, D. & Magerko, B. (2020). What is AI Literacy? Competencies and Design Considerations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (S. 1–16). ACM. <https://doi.org/10.1145/3313831.3376727>
- Lorenz, U. (2024). *Reinforcement Learning: Aktuelle Ansätze verstehen – mit Beispielen in Java und Greenfoot*. Springer. <https://doi.org/10.1007/978-3-662-68311-8>
- Marques, L. S., Gresse Von Wangenheim, C. & Hauck, J. C. R. (2020). Teaching Machine Learning in School: A Systematic Mapping of the State of the Art. *Informatics in Education*, 19(2), 283–321. <https://doi.org/10.15388/infedu.2020.14>
- Martínez-Tenor, Á., Cruz-Martín, A. & Fernández-Madrigal, J.-A. (2019). Teaching machine learning in robotics interactively: The case of reinforcement learning with Lego® Mindstorms. *Interactive Learning Environments*, 27(3), 293–306. <https://doi.org/10.1080/10494820.2018.1525411>
- Naeem, M., Rizvi, S. & Coronato, A. (2020). A Gentle Introduction to Reinforcement Learning and its Application in Different Fields. *IEEE Access*, 8, 209320–209344. <https://doi.org/10.1109/ACCESS.2020.3038605>
- Ng, D. T. K., Leung, J. K. L., Chu, S. K. W. & Qiao, M. S. (2021). Conceptualizing AI literacy: An exploratory review. *Computers and Education: Artificial Intelligence*, 2, Article 100041. <https://doi.org/10.1016/j.caeai.2021.100041>
- Ng, D. T. K., Su, J. & Chu, S. K. W. (2024). Fostering Secondary School Students’ AI Literacy through Making AI-Driven Recycling Bins. *Education and Information Technologies*, 29(8), 9715–9746. <https://doi.org/10.1007/s10639-023-12183-9>
- Ng, D. T. K., Wu, W., Leung, J. K. L., Chiu, T. K. F. & Chu, S. K. W. (2024). Design and validation of the AI literacy questionnaire: The affective, behavioural, cognitive and ethical approach. *British Journal of Educational Technology*, 55(3), 1082–1104. <https://doi.org/10.1111/bjet.13411>
- Ng, D. T. K., Xinyu, C., Leung, J. K. L. & Chu, S. K. W. (2024). Fostering students’ AI literacy development through educational games: AI knowledge, affective and cognitive engagement. *Journal of Computer Assisted Learning*, 40(5), 2049–2064. <https://doi.org/10.1111/jcal.13009>
- Podobnik, J., Udir, A., Muni, M. & Mihelj, M. (2024). Teaching approach for deep reinforcement learning of robotic strategies. *Computer Applications in Engineering Education*, 32(6), Article e22780. <https://doi.org/10.1002/cae.22780>
- Pólya, G. (1945). *How to Solve It*. Princeton University Press.
- Ris-Ala, R. (2023). *Fundamentals of Reinforcement Learning*. Springer Nature Switzerland. <https://doi.org/10.1007/978-3-031-37345-9>
- Shao, K., Tang, Z., Zhu, Y., Li, N. & Zhao, D. (2019). A Survey of Deep Reinforcement Learning in Video Games [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.1912.10944>

- Shute, V. J., Sun, C. & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Singh, B., Kumar, R. & Singh, V. (2022). Reinforcement learning in robotic applications: A comprehensive survey. *Artificial Intelligence Review*, 55(4), 945–990. <https://doi.org/10.1007/s10462-021-09997-9>
- Sutton, R. S. & Barto, A. G. (2018). *Reinforcement Learning, second edition: An Introduction* (2. edition). Bradford Books.
- Tan, Q. & Tang, X. (2025). Unveiling AI literacy in K-12 education: A systematic literature review of empirical research. *Interactive Learning Environments*, 33(9), 5347–5363. <https://doi.org/10.1080/10494820.2025.2482586>
- Touretzky, D., Gardner-McCune, C., Martin, F. & Seehorn, D. (2019). Envisioning AI for K-12: What Should Every Child Know about AI? *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 9795–9799. <https://doi.org/10.1609/aaai.v33i01.33019795>
- Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. University of Cambridge.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>
- Wing, J. M. (2010). *Research Notebook: Computational Thinking-What and Why?* Carnegie Mellon School of Computer Science. <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
- Yang, Y., Zhang, Y., Sun, D., He, W. & Wei, Y. (2025). Navigating the landscape of AI literacy education: Insights from a decade of research (2014–2024). *Humanities and Social Sciences Communications*, 12(1), Article 374. <https://doi.org/10.1057/s41599-025-04583-8>
- Zhong, B. & Liu, X. (2025). Evaluating AI literacy of secondary students: Framework and scale development. *Computers & Education*, 227, Article 105230. <https://doi.org/10.1016/j.compedu.2024.105230>

Anschrift der Verfasser:innen

Alissa Fock
Universität Würzburg
Institut für Mathematik
Emil-Fischer-Str. 30
97074 Würzburg
alissa.fock@uni-wuerzburg.de

Norbert Noster
Universität Würzburg
Professional School of Education
Josef-Martin-Weg 52
97074 Würzburg
norbert.noster@uni-wuerzburg.de

Hans-Stefan Siller
Universität Würzburg
Institut für Mathematik
Emil-Fischer-Str. 30
97074 Würzburg
hans-stefan.siller@uni-wuerzburg.de