Analyse von State-of-the-Art Large Graph Models und deren Architekturen zur Verarbeitung von strukturierten Daten

Research Paper

W3WI_702: Schlüsselqualifikationen II

aus dem Studiengang Wirtschaftsinformatik Sales & Consulting an der Dualen Hochschule Baden-Württemberg Mannheim

von

Julian Konz

Bearbeitungszeitraum:	12.05.2025 - 24.08.2025
Matrikelnummer, Kurs:	3468097, WWI23SCB
Studiengangsleiter:	Prof. Dr. Clemens Martin
Ausbildungsfirma:	SAP SE Dietmar-Hopp-Allee 16 69190 Walldorf, Deutschland
Wissenschaftlicher Betreuer:	Dr. Jörg Astheimer joerg.astheimer@dhbw.de

I. Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema 'Analyse von State-of-the-Art Large Graph Models und deren Architekturen zur Verarbeitung von strukturierten Daten' selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum	**Unterschrift**

II. Gleichbehandlung der Geschlechter

In dieser Praxisarbeit wird aus Gründen der besseren Lesbarkeit das generische Maskulinum verwendet. Weibliche und anderweitige Geschlechteridentitäten werden dabei ausdrücklich mitgemeint, soweit es für die Aussage erforderlich ist.

III. Disclaimer

Ein Teil der Literatur, die für die Anfertigung dieser Arbeit genutzt wird, sind nicht als Buchfassung verfügbar. Bei diesen Ressourcen existieren keine Seitennummern, es wird bei Verweisen stattdessen die Kapitelnummer oder auf die zugehörige Webseite im Literaturverzeichnis angegeben.

Um den Lesefluss zu verbessern, werden Abbildungen und Tabellen, die den Lesefluss stören, im Anhang platziert, auf den im Text zusätzlich verwiesen wird.

IV. Abstract

Titel:	Analyse von State-of-the-Art Large Graph Models und deren Architekturen zur Verarbeitung von strukturierten Daten
Verfasser:	Julian Konz
Kurs:	WWI 23 SCB
Ausbildungsbetrieb:	SAP SE

Im Rahmen dieser Arbeit wird ein systematisches Review zu Large Graph Models (LGMs) durchgeführt, um deren Architekturen zur Verarbeitung strukturierter Daten zu untersuchen. Ausgangspunkt bildet die Analyse von 40 wissenschaftlichen Publikationen, von denen zehn nach definierten Kriterien vertieft betrachtet wurden.

Die Ergebnisse zeigen zwei zentrale Optimierungsrichtungen: Ansätze zur Leistungssteigerung, etwa durch kontrastives Lernen und SimGRACE, sowie zur Erhöhung der Anwendungsvielfalt, beispielsweise durch PRODIGY oder Graph Retrieval-Augmented Generation (GRAG). Diese Ansätze verdeutlichen, dass Datenqualität und Modellgeneralität entscheidende Faktoren für die Leistungsfähigkeit von LGMs sind.

Hybride Architekturen wie Griffin, welche rekurrente Mechanismen mit Attention-Strukturen verbinden, vereinen beide Richtungen und zeigen damit einen vielversprechenden Weg für die Weiterentwicklung von LGMs auf.

Inhaltsverzeichnis

1. Einleitung 1
2. Methodische Vorgehensweise des Reviews 3
3. Ergebnisse der Literaturrecherche 5
3.1. Entwicklung von Large Graph Models 5
3.2. State-of-the-Art Modellarchitekturen 8
3.2.1. Ansätze zur Verbesserung der Leistungsfähigkeit 9
3.2.2. Ansätze zur Erhöhung der Anwendungsvielfalt 10
3.2.3. Zusammenfassung der Ergebnisse 11
4. Schlussbetrachtung 13
4.1. Fazit 13
4.2. Limitationen und Herausforderungen 14
4.3. Ausblick 15
i. Literaturverzeichnis i
ii. Anhang v

Abbildungsverzeichnis

Abbildung 1: Funktionsweise eines GNNs [1] 6

Abbildung 2: Funktionsweise von GRAG [2] 11

Tabellenverzeichnis

Tabelle 1: Ergebnisse der Literaturrecherche v

Abkürzungsverzeichnis

AI	Artificial Intelligence
GAN	Graph Attention Network
GNN	Graph Neural Network
GRAG	Graph Retrieval-Augmented Generation
KI	Künstlicher Intelligenz
LGM	Large Graph Model
LLM	Large Language Model
PRODIGY	Pretraining Over Diverse In-Context Graph Systems
RAG	Retrieval-Augmented Generation
RDL	Relational Deep Learning
RG-LRU	Real-Gated Linear Recurrent Unit
SimGRACE	Simple framework for Graph Contrastive Learning
kNN	k-Nearest Neighbors

1. Einleitung

Aufbauend auf de Grundlagen von C. E. Shannon [3] und der Forschung von A. Vaswani *u. a.* [4] zu neuronalen Netzen, die kontextabhängige Informationen aus komplexen Eingaben nutzen können, um korrekte Modellausgaben zu generieren, ist ein neues Zeitalter von Künstlicher Intelligenz (KI), auf Englisch als Artificial Intelligence (AI) bekannt, eingetreten. Diese Entwicklungen führten zu erheblichen Fortschritten in der Integration von AI-Technologien in vielfältige Anwendungsfelder, insbesondere in betriebswirtschaftliche Prozesse [5, S. 8-9]. Durch AI-gestützte Dienste wird es Unternehmen ermöglicht, Informationen aus Daten zu schließen, aus denen zuvor keine Erkenntnisse gewonnen werden konnten [6], [7, S. 2-3].

Im Kontext von betriebswirtschaftlichen Prozessen liegen Daten in unterschiedlichen Formaten vor. Im Allgemeinen können diese Datenformate in strukturierte, beispielweise in tabellarischer Form vorliegende, und unstrukturierte, beispielweise in Freitexten vorkommende, Daten differenziert werden, wobei die Vielzahl der Daten in strukturierter Form vorliegt. [8, S. 1-2]. [6]. Large Language Models (LLMs) sind generative Sprachmodelle, die aus unstrukturierten Modelleingaben anhand von Wahrscheinlichkeiten eine natürlichsprachige Ausgabe produzieren [7, S. 2]. Seit die Veröffentlichung von OpenAIs ‚ChatGPT' im Jahr 2022 sind LLM in den Fokus der Öffentlichkeit gerückt und haben sich in den letzten Jahren zu einem der am häufigsten eingesetzten AI-Modellarten entwickelt [6]. Der Durchbruch für AI-Modelle zur Verarbeitung strukturierter Daten blieb zunächst aus.

Large Graph Models (LGMs) werden aktuell als vielversprechender Ansatz diskutiert, um eine präzisere Verarbeitung von strukturierten Daten zu ermöglichen. Analog zu LLMs bezeichnet der Begriff LGM Graphmodelle, die aufgrund großer

Modellkapazität und Vortraining auf umfangreichen Graphdaten neuartige Fähigkeiten im Graphbereich zeigen sollen [9, S. 1]. Graphen sind eine fundamentale Datenstruktur zur Repräsentation von Relationen zwischen Entitäten, beispielweise in Form von Wissensgraphen oder Datenbankrelationen, deren effiziente Auswertung für viele Domänen entscheidend ist. [9, S. 1-2]. LGMs basieren auf den Fortschritten von Graph Neural Networks (GNNs) als auch des Transfer-Lernens mit großen Modellen. GNNs stellen den bislang de-facto Standard zur Verarbeitung von Graphen dar [9, S. 1-4].

In dieser Arbeit sollen folgende Forschungsfragen adressiert werden:

1. Welche Architekturen existieren von LGMs und an welchen weiteren Ansätzen wird geforscht?
2. Wie kann die Leistung und Anwendbarkeit eines LGM-Modells verbessert werden?

Im Rahmen einer Literaturrecherche werden insgesamt vier Datenbanken auf jeweils zehn relevante wissenschaftliche Arbeiten untersucht, die sich mit der Verarbeitung von strukturierten Daten durch LGMs befassen. Dabei wird spezifisch auf die Anwendungsmöglichkeiten eingegangen, LGMs zu optimieren und mehrere Ansätze diskutiert. Diese Arbeit zielt darauf ab, einen Einblick in den aktuellen Stand der Forschung zu geben und mögliche zukünftige Entwicklungen in diesem Bereich aufzuzeigen.

2. Methodische Vorgehensweise des Reviews

Dieses Review folgt einem vorab fixierten Protokoll entlang etablierter Leitlinien für systematische Literaturübersichten in Wirtschaftsinformatik/SE (Planung–Durchführung–Bericht) [10, S. 5–7, 18–26]. Untersucht werden die in Abschnitt 1 definierten Forschungsfragen.

Um den State-of-the-Art von LGMs zur Verarbeitung strukturierter Daten zu erheben, wurde eine systematische Literaturrecherche durchgeführt. Als Quellen wurden wissenschaftliche Datenbanken und Repositorien wie arXiv, DBLP, IEEE Xplore, ACM Digital Library und Fachplattformen wie ResearchGate herangezogen, da diese in der Domäne der Wirtschaftsinformatik sowie KI weit verbreitet und allgemein zugänglich für Studenten der DHBW Mannheim sind. Die Suche konzentrierte sich auf Veröffentlichungen nach der Veröffentlichung von A. Vaswani *u. a.* [4], der erstmals das Konzept von Selbstaufmerksamkeitsmechanismen vorstellte, da diese ein neues Kapitel in der Forschung von KI eingeleitet hat. durch Peers begutachtete Veröffentlichungen (NeurIPS, ICML, KDD, ICLR u. a.) werden im Review höher gewichtet als nicht durch Peers begutachtete Beiträge [10, S. 18–19].

Folgende Begriffe werden in den Datenbanken gesucht, um relevante wissenschaftliche Arbeiten zu identifizieren:

- “Large Graph Model (LGM)”
- “Graph Neural Network (GNN)”
- „LGM-Architectures“
- “Relational Deep Learning”
- “Graph Representation Learning”

- “Graph pre-training”
- “Graph foundation model”

Durch Kombination dieser Begriffe konnten einschlägige Arbeiten identifiziert werden, beispielsweise führte die Suche nach “Graph meets LLM” zur Entdeckung eines Überblicksartikels von Z. Zhang, H. Li, Z. Zhang, Y. Qin, X. Wang, und W. Zhu [9], während “graph in-context learning” auf neuere Ansätze wie Z. Hu *u. a.* [11] und Pretraining Over Diverse In-Context Graph Systems (PRODIGY) von Q. Huang *u. a.* [12] verwies. Die Begriffe sind bewusst auf Englisch formuliert, da die Primärliteratur überwiegend englischsprachig ist.

Um den Umfang dieser Arbeit zu wahren, werden Ausschlusskriterien definiert. Eingeschlossen werden Arbeiten, die explizit großskalige LGMs zum Thema haben, insbesondere welche neue Architekturen betrachten, wobei Studien zu klassischen GNNs ohne Bezug zur Modellskalierung oder zum LLM-Kontext ausgeschlossen werden. Um die neusten Erkenntnisse zu erfassen, werden Arbeiten vor 2018 ausgeschlossen, da sich nach der Implementierung von Aufmerksamkeitsmechanismen durch P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, und Y. Bengio [13] aus dem Jahr 2018 die Einsatzmöglichkeiten von LGMs fundamental verändert haben. [9, S. 2].

Nachdem eine Kernliste von Publikationen zusammengestellt war, erfolgt eine inhaltliche Prüfung der wissenschaftlichen Veröffentlichungen. Die daraus gewonnenen Erkenntnisse wurden kritisch synthetisiert und werden im folgenden Kapitel dargestellt. Im Folgenden werden die Erkenntnisse der Literaturrecherche zusammengefasst und die Forschungsfragen beantwortet.

3. Ergebnisse der Literaturrecherche

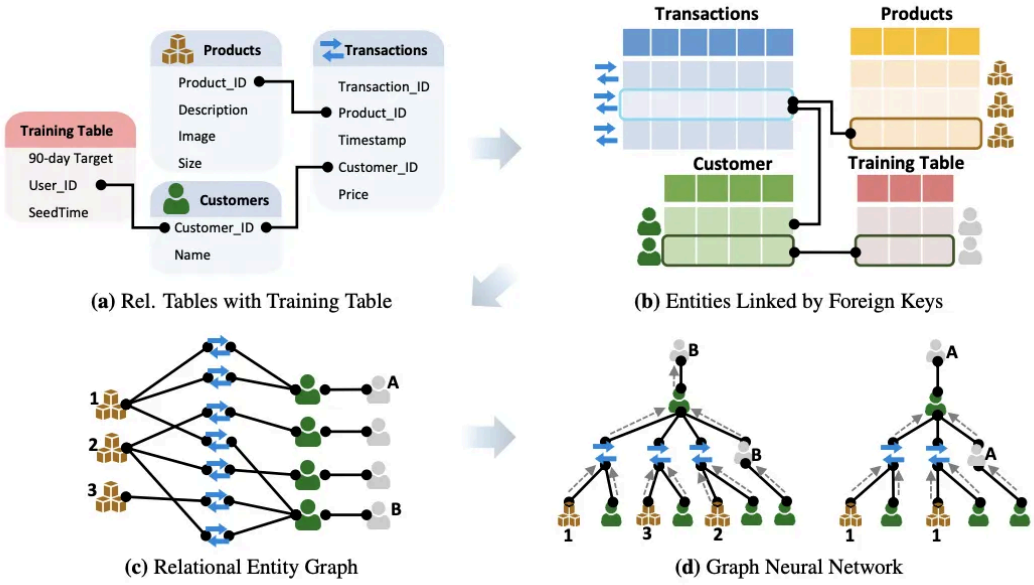
Die Literaturrecherche ergab insgesamt 40 relevante wissenschaftliche Arbeiten, die sich mit der Verarbeitung von strukturierten Daten durch LGMs und LLMs befassen. Diese Arbeiten wurden in einem Review anhand der vorab definierten Ausschlusskriterien evaluiert. Die Ergebnisse der Filterung anhand der definierten Ausschlusskriterien aus Abschnitt 2 sind in Tabelle 1 im Anhang dargestellt und umfassen zehn wissenschaftliche Publikationen.

In den folgenden Kapiteln wird zunächst auf die Entwicklung von LGMs eingegangen, um Kontext und Verständnis für weitere Kapitel zu schaffen. Anschließend werden die State-of-the-Art Modellarchitekturen für LGMs vorgestellt und voneinander abgegrenzt, um ein umfassendes Verständnis der aktuellen Entwicklungen in diesem Bereich zu vermitteln und Gegenpositionen zur Optimierung eines LGMs zu vergleichen. Abgeschlossen wird das Kapitel mit einer Zusammenfassung der Ergebnisse.

3.1. Entwicklung von Large Graph Models

Ein Graph ist eine Datenstruktur, die aus Knoten (Entitäten) und Kanten (Beziehungen zwischen Entitäten) besteht und zur Darstellung von relational strukturierten Daten verwendet wird. [1, S. 1-2]. Diese Knoten stehen i.d.R. in einer Subjekt-Prädikat-Objekt Beziehung zueinander [12, S. 3]. Ein Beispiel hierfür ist die Aussage „Berlin ist die Hauptstadt von Deutschland“, welche in einem Graphen durch die Knoten „Berlin“ und „Deutschland“ sowie die Kante „ist die Hauptstadt von“ dargestellt werden kann.

Vor der Einführung von GNNs wurden Informationen aus relational strukturierten Daten i.d.R. mit traditionellen Machine-Learning-Verfahren wie k-Nearest Neighbors (kNN), bei welcher Datenpunkte als Vektoren in einem hochdimensionalen Raum dargestellt werden, verarbeitet. Diese Verfahren sind jedoch nicht in der Lage, neue Informationen zu verarbeiten, sondern nur bekannte Datenpunkte zu klassifizieren [14, S. 1-2].



(a) Rel. Tables with Training Table (b) Entities Linked by Foreign Keys

(c) Relational Entity Graph (d) Graph Neural Network

Abbildung 1: Funktionsweise eines GNNs [1]

Den Grundbaustein legten F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, und G. Monfardini [15] im Jahr 2009 mit der Einführung von GNNs, die es ermöglichen, Graphen in neuronalen Netzen zu verarbeiten und erste Vorhersagen zu treffen, indem Datenpunkte nicht mittels Suchalgorithmen aus einer bekannten Liste abgerufen werden, sondern mittels einer Funktion diese dynamisch generiert werden [15, S. 2]. Abbildung 1 stellt die Funktionsweise dar. Zur Verarbeitung von relational strukturierten Daten müssen diese zunächst in Graphen umgewandelt werden (b), die die Relationen zwischen Tabellen in Form von Knoten und Kanten brücksichtigt [1, S. 3]. Diese können über ein neuronales Netz (c) in Vektoren

konvertiert und somit verarbeitet werden, um mittels einer Funktion eine Ausgabe zu generieren, welche in Form eines Datenpunktes ausgegeben wird. [16, S. 1-2]

Die Entwicklung von GNNs führte zu einer Vielzahl von Architekturen, die sich auf verschiedene Anwendungsfälle konzentrierten. GraphSAGE aus dem Jahr 2017 von W. Hamilton, Z. Ying, und J. Leskovec [16] ist eine der ersten GNNs-Architekturen, welche im Gegensatz zu traditionellen GNNs-Architekturen einen induktiven Ansatz nutzt, um Graphen zu verarbeiten. Dies bedeutet, dass sie in der Lage ist, neue Knoten und Kanten zu verarbeiten, die nicht im Trainingsdatensatz enthalten sind, indem es bestehende Knoten nicht als Datenpunkte, sondern als Funktionen auffasst. Dadurch ist es möglich, anstatt einer Abfrage bestehender Informationen eine Wahrscheinlichkeit zu berechnen. Erweitert wurde diese Architektur von P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, und Y. Bengio [13] im Jahr 2018, indem Aufmerksamkeitsmechanismen aus A. Vaswani *u. a.* [4] in Graphen eingeführt wurden, um einen stärkeren Fokus auf die Relationen zwischen Entitäten zu legen. Diese Architektur, bekannt als Graph Attention Networks (GANs), beziehen im Gegensatz zu kNN nicht nur die nächsten k Nachbarn eines Knotens in die Verarbeitung ein, sondern beziehen auch weitere, nicht direkt durch Kanten zusammenhängende Knoten in die Verarbeitung ein, um eine höhere Genauigkeit zu erzielen. Diese Architekturen schaffen die Basis für die Entwicklung von LGMs [13, S. 3-4].

LGMs sind eine Weiterentwicklung traditionellen GNNs, die auf Basis von LLMs-Architekturen entwickelt wurden und in spezifisch ausgewählten Domänen in Benchmarks wie „RelBench“, ein Python-Paket der Stanford Univerity, welches zur Umsetzung von Relational Deep Learning (RDL) in GNNs verwendet wird, höhere Ergebnisse als reguläre GNNs erzielen [1, S. 5-7], [9, S. 1-2]. Sie umfassen oft ein mehrschichtiges Netzwerk mit Millionen bis Milliarden von Parametern,

die auf umfangreichen Graphdaten vortrainiert wurden, um eine Vielzahl von Aufgaben zu bewältigen. [9, S. 1-2].

3.2. State-of-the-Art Modellarchitekturen

Unter der Vernetzung mehrerer GNNs zu einem großen, mehrschichtigen Modell mit hoher Parameteranzahl spricht man von einem LGMs [1, S. 1-2]. Die Funktionsweise, wie diese Schichten miteinander interagieren wird durch die LGM-Architektur bestimmt. Oft nutzen die Fortschritte in der LLMs-Architektur, um die Verarbeitung von Graphen zu verbessern und durch erhöhte Parametergröße und Trainingsvolumen eine höhere Genauigkeit zu erzielen [11]. Darüber hinaus erlauben diese durch die Integration von LLMs-Komponenten eine einfache Interaktion mit den Modell durch Prompts, ähnlich wie bei LLMs, wodurch sie für Endbenutzer ohne technische Kenntnisse zugänglicher werden [11, S. 1-2].

In der Forschung gibt es mehrere Ansätze, Performance von LGMs zu verbessern. Diese Ansätze können in zwei Kategorien unterteilt werden:

1. Architekturen, welche die Qualität der Ausgaben eines LGMs in einem spezifischen Anwendungsfall verbessern
2. Architekturen, welche ein LGMs-Modell auf eine Vielzahl von Anwendungsfällen anwendbar machen

Im Folgenden werden Ansätze beider Kategorien vorgestellt und deren Funktionsweise erläutert und die jeweiligen Vorteile durch deren Implementierung dargestellt. Abschließend wird diskutiert, wie zukünftige LGMs-Architekturen von diesen Ansätzen profitieren können und in welche Richting sich die Forschung entwickeln könnte.

3.2.1. Ansätze zur Verbesserung der Leistungsfähigkeit

Eine bekannte Schwäche von AI-Modellen liegt in der Datenqualität, auf welcher das Modell trainiert wurde. Ein LGM kann nur so gut sein wie die Daten, auf denen es trainiert wurde, welche i.d.R. in unzureichenden Mengen vorliegen oder mit hohen Erstellungskosten verbunden sind. Der Fokus der Literatur liegt daher auf der Verbesserung der Trainingsdaten sowie der Reduktion manueller Eingriffe in den Trainingsprozess. [17, S. 1-2]

Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, und Y. Shen [18] aus dem Jahr 2021 stellen einen Ansatz vor, der die Qualität von GNNs-Modellen durch Kontrastives Lernen verbessert. Dieser Ansatz nutzt Augmentierungen, um verschiedene Sichten auf denselben Graphen zu erzeugen und diese dann zu vergleichen, um die Qualität der Repräsentationen zu verbessern. Dies ermöglicht es, die Robustheit und Generalisierungsfähigkeit des Modells zu erhöhen. Dieser Ansatz wird in der Literatur als „Graph Contrastive Learning“ bezeichnet und hat sich als effektiv erwiesen, um die Leistung von GNNs-Modellen zu verbessern [18, S. 1-2].

J. Xia, L. Wu, J. Chen, B. Hu, und S. Z. Li [19] aus dem Jahr 2022 erweiterten diesen Ansatz, indem sie Simple framework for Graph Contrastive Learning (SimGRACE) vorstellt, die die Qualität von GNNs-Modellen durch Kontrastives Lernen ohne Augmentierungen verbessert. SimGRACE erzeugt Permutationen eines Graphen, indem es zwei Sichten auf denselben Graphen generiert Dadurch ist keine Augmentierung der Daten notwendig, was die Effizienz des Trainingsprozesses erhöht und falsche Augmentierung des Graphen vermeidet. Dieser Ansatz hat sich als effektiv erwiesen, um die Leistung von GNNs-Modellen zu verbessern und die Robustheit gegenüber Rauschen und Störungen zu erhöhen. [19, S. 1-4]

3.2.2. Ansätze zur Erhöhung der Anwendungsvielfalt

Damit ein LGMs in einer Vielzahl von Anwendungsfällen eingesetzt werden kann, muss dieses eine ausreichend hohe Qualität in einer breiten Anwendungsvielfalt aufweisen. Folgende Ansätze werden in der Literatur diskutiert:

- **Vereinheitlichung der Eingabeformate**: Mittels einer einheitlichen Schnittstelle zur Interaktion und zum Training eines LGMs wird die Nutzung solcher Modelle vergleichbar zu LLMs, somit intuitiver für Endanwender.
- **Breite Wissensbasis**: Durch eine größere Wissensbasis kann ein Modell vielfältig eingesetzt werden, vergleichbar mit modernen LLMs.

In der Veröffentlichug von Q. Huang *u. a.* [12] aus dem Jahr 2023 wird ein Ansatz namens PRODIGY vorgestellt, der eine durch natürlichsprachige Modelleingaben nutzbare Schnittstelle zur Interaktion mit einem LGMs-Modellen bietet. PRODIGY führt eine einheitliche „Prompt-Graph“-Darstellung ein, mit der sich Knoten-, Kanten- und Graph-Aufgaben gleichermaßen als Kontext-plus-Abfrage ausdrücken lassen [12, S. 1-2]. Ein LGM, welches i.d.R. mehrschichtige GANs nutzt, kann dynamisch aus diesem Teilgraphen extrahieren, welche als Few-Shot-Prompts genutzt werden können und dem Prompt beigefügt werden [12, S. 2]. Durch Hinzugabe von konkreten, sachbezogenen Beispielen für jede Aufgabe wird ein breiter, aufgabenübergreifender Einsatz möglich [12, S. 8-10].

Weitere Ansätze zur Erweiterung der Wissensbasis umfassen Integrationen von Retrieval-Augmented Generation (RAG)-Systemen in LGMs, worunter man eine Anbindung einer Vektordatenbank an ein AI-Modell versteht [20, S. 1]. Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling, und L. Zhao [2] aus dem Jahr 2025 stellen einen Ansatz vor, der die Leistung von LGMs-Modellen durch die Integration von RAG-Systemen verbessert. Graph Retrieval-Augmented Generation (GRAG)

integriert eine Vektordatenbank, die Textchunks als Wissensgraph speichert, die jeweils zum Prompt passenden Chunks aus der Datenbank lädt und diese als Kontext für das LGMs-Modell bereitstellt. Eine vereinfachte Darstellung ist in Abbildung 2 dargestellt. Dies erlaubt es, ein LGM-Modell mit einer breiten Wissensbasis zu betreiben, ohne dass dieses explizit auf diese Daten trainiert werden muss [20, S. 2-3]. [2, S. 1-4].

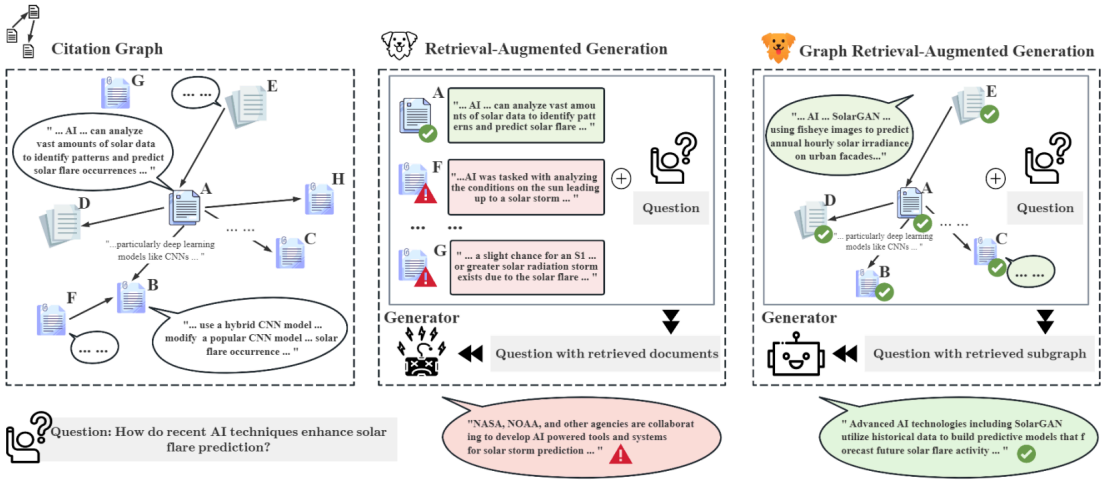


Abbildung 2: Funktionsweise von GRAG [2]

3.2.3. Zusammenfassung der Ergebnisse

Nach Betrachtung beider Ansätze zur Verbesserung der Leistungsfähigkeit und zur Erhöhung der Anwendungsvielfalt zeigt sich, dass eine Kombination vergleichbar mit der Entwicklung im Bereich der LLMs-Architekturen sinnvoll ist. Während die Leistungsfähigkeit durch eine Verbesserung der Trainingsdaten und der Reduktion manueller Eingriffe in den Trainingsprozess gesteigert wird, wird die Anwendungsvielfalt durch eine Vereinheitlichung der Eingabeformate und eine breite Wissensbasis erhöht. Diese Ansätze ermöglichen es, LGMs in einer Vielzahl von Anwendungsfällen einzusetzen und deren Leistung zu verbessern.

Eine LGM-Architektur, die beide Ansätze kombiniert, wurde von S. De *u. a.* [21] im Jahr 2024 vorgestellt. Griffin ist ein hybrides LGM, das Real-Gated Linear Recurrent Unit (RG-LRU) mit lokaler Multi-Query Attention aus GANs kombiniert, um sowohl die spezifische Leistungsfähigkeit auf Modell-Ebene als auch die allgemeine Anwendbarkeit zu gewährleisten. Dabei werden mehrere Expertenmodelle trainiert, die jeweils auf einen spezifischen Anwendungsfall spezialisiert sind. Ein Gating-Netzwerk entscheidet dann, welcher Experte für eine gegebene Eingabe am besten geeignet ist. Der Kern liegt darin, die Stärken rekurrenter und aufmerksamkeitsbasierter Mechanismen zu vereinen, ohne die hohen Rechenkosten globaler Attention-Strukturen in Kauf nehmen zu müssen, wodurch Griffin in der Lage ist, diverse Use-Cases bei niedrigem Tokeneinsatz abzudecken [21, S. 1-3].

Inwiefern ein Fokus auf eine der beiden Kategorien sinnvoll ist, hängt stark vom Anwendungsfall ab, weswegen sich anhand der Literatur kein eindeutiger Trend ableiten lässt. Anhand der Anzahl der in diesem Review betrachteten wissenschaftlichen Arbeiten zeigt sich, dass die Forschung der Verbesserung der Leistung einen höheren Stellenwert zuschreibt, während die Publikationen, welche von Unternehmen veröffentlicht wurden, einen stärkeren Fokus auf die Erhöhung der Anwendungsvielfalt und Einsparung von Kosten legen [21, S. 1-2], [22, S. 5-7]. Aufgrund dieser Differenz kann keine eindeutige Tendenz abgeleitet werden und bei einer Optimierung eines LGMs-Modells sollte sich anhand gegebener Umstände für einen der beiden Ansätze entschieden werden.

4. Schlussbetrachtung

Im Folgenden werden die Ergebnisse zusammengefasst, deren Limitationen und Herausforderungen diskutiert und ein Ausblick auf zukünftige Entwicklungen gegeben.

4.1. Fazit

In dieser Arbeit wurde im Rahmen einer Literaturrecherche wie in Abschnitt 2 erläutert ein Überblick über den aktuellen Stand der Forschung zu LGMs gegeben. Nach einem Einblick in die Entwicklung von Methoden wie kNN bis hin zu LGMs wurden verschiedene Architekturen vorgestellt und deren Funktionsweise erklärt.

Dabei zeigte sich, dass es zwei Hauptansätze zur Optimierung von LGMs gibt: die Verbesserung der Leistungsfähigkeit und die Erhöhung der Anwendungsvielfalt. Beide Ansätze haben ihre Vor- und Nachteile, und die Wahl des geeigneten Ansatzes hängt stark vom jeweiligen Anwendungsfall ab. Umsetzungen wie Griffin von S. De *u. a.* [21] zeigen, dass eine Kombination beider Ansätze möglich ist und sich gegenüber älterer GNNs-Architekturen in Benchmarks wie „RelBench“ durchsetzen kann.

Zukünftige Entwicklungen in diesem Bereich könnten darauf abzielen, die Stärken beider Ansätze zu kombinieren, um LGMs-Modelle zu schaffen, die sowohl leistungsfähig als auch vielseitig einsetzbar sind.

4.2. Limitationen und Herausforderungen

Die Aussagekraft der Ergebnisse dieser Arbeit unterliegt folgenden Limitationen, welche die Generalisierbarkeit der Ergebnisse einschränken:

- **Begrenzte Anzahl an wissenschaftlichen Arbeiten**: Die Literaturrecherche ergab nur eine begrenzte Anzahl an relevanten wissenschaftlichen Arbeiten, was die Generalisierbarkeit der Ergebnisse einschränkt.
- **Schneller Wandel der Forschung**: Der Bereich der LGMs befindet sich in einem schnellen Wandel, wodurch die Ergebnisse dieser Arbeit möglicherweise nur eine Momentaufnahme darstellen und zukünftige Entwicklungen nicht berücksichtigen.
- **Subjektive Auswahl der Arbeiten**: Die Auswahl der wissenschaftlichen Arbeiten erfolgte subjektiv, was zu einer Verzerrung der Ergebnisse führen könnte.
- **Fokus auf bestimmte Datenbanken**: Die Literaturrecherche konzentrierte sich auf bestimmte wissenschaftliche Datenbanken, wodurch relevante Arbeiten, die in anderen Datenbanken veröffentlicht wurden, möglicherweise übersehen wurden.

Ebenfalls ergaben sich Herausforderungen hinsichtlich Zugriffsrechte und Verfügbarkeit von Quellen, welche bei einer wiederholten Recherche berücksichtigt werden sollten und ggf. zu abweichenden Ergebnissen führen könnten. Darüber hinaus wurden die Erkenntnisse der wissenschaftlichen Arbeiten nicht auf ihre praktische Umsetzbarkeit geprüft, wodurch Herausforderungen bei der Implementierung in realen Anwendungsfällen auftreten könnten.

4.3. Ausblick

Zusammenfassend lässt sich sagen, dass LGMs ein vielversprechender Ansatz zur Verarbeitung von strukturierten Daten ist, der in Zukunft weiter erforscht und optimiert werden sollte. Die Vorteile, die eine Speicherung von strukturierten Daten in Form von Graphen bietet, sind vielversprechend und könnten in vielen Anwendungsfällen im Geschäftsumfeld zu einer verbesserten Leistung führen.

Inwiefern die in dieser Arbeit vorgestellten Ansätze zur Optimierung von LGMs-Modellen in der Praxis umsetzbar sind, sollte in zukünftigen Arbeiten untersucht werden. Dabei könnten auch Herausforderungen bei der Implementierung in realen Anwendungsfällen adressiert werden, ebenso der Vergleich der Leistung von LGM-Modellen mit anderen AI-Modellarten, wie tabellenbasierten Modelen oder LLMs, hinsichtlich eines spezifischen Use-Cases, um deren Stärken und Schwächen besser zu verstehen.

i. Literaturverzeichnis

[1] M. Fey *u. a.*, „Relational Deep Learning: Graph Representation Learning on Relational Databases“. Zugegriffen: 20. August 2025. [Online]. Verfügbar unter: https://arxiv.org/abs/2312.04615

[2] Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling, und L. Zhao, „GRAG: Graph Retrieval-Augmented Generation“. Zugegriffen: 22. August 2025. [Online]. Verfügbar unter: https://arxiv.org/abs/2405.16506

[3] C. E. Shannon, „A Mathematical Theory of Communication“, *The Bell System Technical Journal*, Bd. 27, Nr. 3, 4, S. 379–423623–656, 1948.

[4] A. Vaswani *u. a.*, „Attention Is All You Need“, in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2017. Zugegriffen: 27. März 2025. [Online]. Verfügbar unter: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[5] R. Srivastava, S. Prasad, L. Bhat, S. Deshpande, B. Das, und K. Jadhav, „MedPromptExtract (Medical Data Extraction Tool): Anonymization and Hi-fidelity Automated data extraction using NLP and prompt engineering“. Zugegriffen: 7. März 2025. [Online]. Verfügbar unter: https://arxiv.org/abs/2405.02664v3

[6] McKinsey *u. a.*, „The Economic Potential of Generative AI: The Next Productivity Frontier“, Juni 2023. Zugegriffen: 27. Februar 2025. [Online]. Verfügbar unter: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier

[7] W. X. Zhao *u. a.*, „A Survey of Large Language Models“, *arXiv*, 2023, Zugegriffen: 26. Februar 2025. [Online]. Verfügbar unter: https://arxiv.org/abs/2303.18223

[8] A. E. Johnson *u. a.*, „MIMIC-III, a freely accessible critical care database“, *Scientific Data*, Bd. 3, S. 160035, 2016, doi: 10.1038/sdata.2016.35.

[9] Z. Zhang, H. Li, Z. Zhang, Y. Qin, X. Wang, und W. Zhu, „Graph Meets LLMs: Towards Large Graph Models“. [Online]. Verfügbar unter: https://arxiv.org/abs/2308.14522

[10] B. Kitchenham und S. Charters, „Guidelines for Performing Systematic Literature Reviews in Software Engineering“, UK, 2007.

[11] Z. Hu *u. a.*, „Let's Ask GNN: Empowering Large Language Model for Graph In-Context Learning“. [Online]. Verfügbar unter: https://arxiv.org/abs/2410.07074

[12] Q. Huang *u. a.*, „PRODIGY: Enabling In-context Learning Over Graphs“. [Online]. Verfügbar unter: https://arxiv.org/abs/2305.12600

[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, und Y. Bengio, „Graph Attention Networks“. Zugegriffen: 20. August 2025. [Online]. Verfügbar unter: https://arxiv.org/abs/1710.10903

[14] P. Cunningham und S. J. Delany, „k-Nearest Neighbour Classifiers - A Tutorial“, *ACM Computing Surveys*, Bd. 54, Nr. 6, S. 1–25, 2021, doi: 10.1145/3459665.

[15] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, und G. Monfardini, „The Graph Neural Network Model“, *IEEE Transactions on Neural Networks*, Bd. 20, Nr. 1, S. 61–80, 2009, doi: 10.1109/TNN.2008.2005605.

[16] W. Hamilton, Z. Ying, und J. Leskovec, „Inductive Representation Learning on Large Graphs“, in *Advances in Neural Information Processing Systems*, 2017.

[17] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, und G. Neubig, „Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing“, *arXiv preprint*, 2021, Zugegriffen: 27. März 2025. [Online]. Verfügbar unter: https://arxiv.org/pdf/2107.13586

[18] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, und Y. Shen, „Graph Contrastive Learning with Augmentations“. Zugegriffen: 21. August 2025. [Online]. Verfügbar unter: https://arxiv.org/abs/2010.13902

[19] J. Xia, L. Wu, J. Chen, B. Hu, und S. Z. Li, „SimGRACE: A Simple Framework for Graph Contrastive Learning without Data Augmentation“, in *Proceedings of the ACM Web Conference 2022*, ACM, 2022, S. 1070–1079. doi: 10.1145/3485447.3512156.

[20] Y. Mao *u. a.*, „Generation-Augmented Retrieval for Open-Domain Question Answering“, in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, und R. Navigli, Hrsg., Online: Association for Computational Linguistics, 2021, S. 4089–4100. doi: 10.18653/v1/2021.acl-long.316.

[21] S. De *u. a.*, „Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models“. [Online]. Verfügbar unter: https://arxiv.org/abs/2402.19427

[22] C. Ma, Y. Chen, T. Wu, A. Khan, und H. Wang, „Large Language Models Meet Knowledge Graphs for Question Answering: Synthesis and Opportunities“. Zugegriffen: 22. August 2025. [Online]. Verfügbar unter: https://arxiv.org/abs/2505.20099

ii. Anhang

Titel	Autoren	Jahr	Quelle
Graph Attention Networks	Velickovic et al.	2018	arXiv
Graph Meets LLMs: Towards Large Graph Models	Zhang et al.	2023	arXiv
Let's Ask GNNs: Empowering Large Language Models with Graph In-Context Learning	Hu et al.	2025	arXiv
GraphPro: Graph Pre-training and Prompt Learning for Recommendation	Yuhao et al.	2024	ACM Digital Library
PRODIGY: Enabling In-context Learning Over Graphs	Huang et al.	2023	arXiv
RelBench: A Benchmark for Deep Learning on Relational Databases	Robinson et al.	2024	arXiv
Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models	De et al.	2024	arXiv

Tabelle 1: Ergebnisse der Literaturrecherche