THE IEEE Intelligent Informatics BULLETIN

IEEE COMPUTER SOCIETY

IEEE Computer Society
Technical Committee
on Intelligent Informatics

December 2024 Vol. 24 No. 1 (ISSN 1727-5997)

Profile

Research Articles

Selected PhD Thesis Abstracts

Announcements

On-line version: http://www.comp.hkbu.edu.hk/~iib (ISSN 1727-6004)

The Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society deals with tools and systems using biologically and linguistically motivated computational paradigms such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality. If you are a member of the IEEE Computer Society, you may join the TCII without cost at http://computer.org/tcsignup/.

The IEEE Intelligent Informatics Bulletin

Aims and Scope

The IEEE Intelligent Informatics Bulletin is the official publication of the Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society, which is published once a year in both hardcopies and electronic copies. The contents of the Bulletin include (but may not be limited to):

1) Letters and Communications of the TCII Executive Committee
2) Feature Articles
3) R&D Profiles (R&D organizations, interview profile on individuals, and projects etc.)
4) Book Reviews
5) News, Reports, and Announcements (TCII sponsored or important/related activities)

Materials suitable for publication at the IEEE Intelligent Informatics Bulletin should be sent directly to the Associate Editors of respective sections.

Technical or survey articles are subject to peer reviews, and their scope may include the theories, methods, tools, techniques, systems, and experiences for/in developing and applying biologically and linguistically motivated computational paradigms, such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, and self-organization in the research and application domains, such as data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality.

Publisher: *The IEEE Computer Society Technical Committee on Intelligent Informatics*
Address: *Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong (Attention: Dr. William K. Cheung; Email:william@comp.hkbu.edu.hk)*
ISSN Number: *1727-5997(printed)1727-6004(on-line)*
Abstracting and Indexing: *All the published articles will be submitted to the following on-line search engines and bibliographies databases for indexing—Google(www.google.com), The ResearchIndex(citeseer.nj.nec.com), The Collection of Computer Science Bibliographies (liinwww.ira.uka.de/bibliography/index.html), and* ***DBLP*** *Computer Science Bibliography (www.informatik.uni-trier.de/~ley/db/index.html).*

From Syntax to Semantics: Exploring the Frontiers of NLP at Wilfrid Laurier

JIASHU ZHAO AND GURUNAMEH SINGH CHHATWAL
INTELLIGENT SYSTEMS FOR PREDICTIVE INFORMATION RETRIEVAL AND DATA MODELING LAB
WILFRID LAURIER UNIVERSITY, WATERLOO, CANADA
JZHAO@WLU.CA, CHHA8740@MYLAURIER.CA

I. INTRODUCTION

The Intelligent Systems for Predictive Information Retrieval and Data Modeling (INSPIRE) Lab at Wilfrid Laurier University is dedicated to advancing the fields of information retrieval (IR), machine learning (ML), and large language models (LLMs) through cutting-edge research and innovation. The lab focuses on developing new algorithms, models, and techniques to enhance how information is processed, understood, and retrieved across diverse domains. By integrating modern machine learning approaches and LLMs with traditional IR methods, the lab seeks to improve the accuracy, efficiency, and relevance of search and recommendation systems. Research at the lab includes the application of natural language processing (NLP), deep learning, and large-scale data analytics to tackle real-world challenges in information retrieval, personalization, and knowledge discovery. The lab explores the potential of LLMs in advancing tasks like semantic search, question answering, and content generation. With a collaborative environment, the lab fosters interdisciplinary research and supports the development of advanced technologies that impact fields such as search engines, e-commerce platforms, and recommender systems.

II. RESEARCH AREAS

The INSPIRE Lab at Wilfrid Laurier University focuses on a broad range of research areas that combine information retrieval (IR), machine learning (ML), recommender systems, and large language models (LLMs). Some of the key research areas include:

- Probabilistic Models for Information Retrieval: This research explores the application and enhancement of probabilistic frameworks in information retrieval (IR), focusing on the development of models that assess the likelihood of document relevance.
- Personalized Information Retrieval Techniques: This area investigates methods for adapting IR systems to individual user needs by incorporating user profiles, preferences, and contextual information.
- Graph-Based Approaches in Information Retrieval: Focused on leveraging graph theory and network structures to enhance IR systems, this research examines how to model and exploit the relationships between documents, queries, and users.
- Deep Learning for Textual Data Analysis: This research explores the application of advanced deep learning techniques for analyzing unstructured textual data. We investigate the use of neural networks, including CNNs, RNNs, and transformer-based models like BERT, to extract semantic features and improve the retrieval of relevant documents from large-scale corpora.
- Representation Learning for Textual Data: This research focuses on the development of robust embedding methods for textual data to better capture semantic relationships within language. We investigate approaches such as contextual embeddings (BERT, GPT) to improve the representation of text for downstream retrieval and classification tasks.
- Text Classification for Information Retrieval: Investigating methods to automatically categorize text into predefined labels, this research enhances text classification techniques for various applications, including categorization, spam detection, AI generated text detection, and etc.
- Analysis and Mitigation of Ranking Biases: This research examines the inherent biases that influence the ranking of search and recommendation results, including algorithmic, content, and user-related biases. We focus on developing methods to identify and mitigate these biases, ensuring more fair, diverse, and unbiased ranking outcomes across a variety of applications, from search engines to recommender systems.
- User Behavior Modeling in Recommender Systems: This area investigates techniques for analyzing and modeling user interaction and feedback within recommender systems. Our research aims to develop models that predict user preferences based on clickstream data, user behavior, and contextual cues, enhancing the personalization and relevance of recommendations over time.
- User-Centric Approaches: Focusing on improving the user experience in information retrieval and recommendation systems, this research studies how users interact with IR systems. We explore how to optimize interfaces and algorithms to enhance satisfaction, engagement, and performance, through user-centric models and feedback loops.
- Applications of Large Language Models in Information Retrieval: This research investigates and improves the use of large pre-trained language models (e.g., BERT and ERNIE) to improve various IR tasks such as semantic search and contextual document retrieval. We

explore the fine-tuning of these models to address domain-specific challenges, with an emphasis on enhancing contextual understanding and search relevance.
- Video Search and Retrieval: This research focuses on enhancing video content retrieval by integrating textual, visual, and audio data using multi-modal and multi-task learning approaches. We apply graph neural networks (GNNs) to capture relationships between video elements and user queries, improving search accuracy and efficiency.
- AI Forensics and Authorship Attribution: With the increase in the usage of LLMs in digital environments we focus on attributing the generated content to its neural author. We leverage pre-trained language models and multiple specialized features to distinguish between machine and human generated content. Furthermore, we apply this to more novel applications in public safety and secure digital communication.

III. Projects and Achievements

A. Current Projects

1) Large Language Model Enhanced Retrieval: By tapping into the generative and contextual understanding strengths of LLMs, the project aims to improve the alignment between user queries and retrieved information. The integration of advanced language models enables deeper comprehension of user intent and context, addressing ambiguities and enhancing the overall search experience. At its core, this initiative seeks to redefine how queries are understood and processed in retrieval systems, ensuring more accurate and relevant results across diverse use cases. By combining the scalability and adaptability of LLMs with robust retrieval strategies, the project sets the foundation for next-generation information systems that can dynamically adapt to complex user needs, offering groundbreaking improvements in relevance and precision.

2) Adaptive smart information systems: A key future direction for intelligent systems lies in their ability to dynamically adapt to user needs and contexts while deeply understanding the content they process. The vision is to create adaptive smart systems that go beyond simple information retrieval to interpret, organize, and present data in ways that are contextually relevant and aligned with user intent. These systems will evolve with changing trends, behaviors, and environments, ensuring continuous relevance and value.

By advancing methods for content understanding and user intent modeling, these systems will bridge the gap between raw data and actionable insights. They will enable highly personalized and intuitive experiences across search, recommendation, and decision-making platforms. Through real-time adaptation to user interactions and data nuances, these intelligent systems will redefine how users engage with information, consistently delivering precise and context-aware results across diverse applications.

3) Advanced framework to combat AI misinformation: Misinformation and fake content is a severe issue in the current society as we are succumbed to digital venues for our information consumption. Most of these areas are not regulated by any authority which gives an edge to the bad actors with an intent to spread false narrative. With the widespread use of AI and the rapid development in the quality of work that it can mimic as a human, it is serves as a great tool for proliferation of false content. It may not always be the ill intentions of a user of generative models to spread false content but due to their non-deterministic nature they are prone to hallucinate and make up facts which may not be true. The volume by which this type of content has seeped into our online ecosystems is alarming and requires frameworks that help automate the detection process.

We are currently working on a framework that uses meta and transfer learning strategy to study the common embeddings that can be used to detect fake content as well as AI generated content. The study is comprehensive of both a general strategy as well as specializing to particular use cases.

This will also lead to generation of a baseline dataset that is first of its kind for machine generated false content detection. It will be useful to further research in this and related areas. Moreover, we try to introduce a novel social based prompting strategy to create such datasets which helps in explainability of the models actions on how it learned and reduces bias.

B. Past Achievements

1) Innovative Approaches for Unbiased Learning and Ranking: Bias in information retrieval (IR) and recommender systems is a significant challenge that impacts the accuracy, fairness, and effectiveness of search results and recommendations. This project explored innovative methods to identify, understand, and mitigate various types of biases, addressing both their causes and their effects on system performance. The studies conducted within this project proposed frameworks and algorithms designed to create more equitable and reliable IR and recommender systems.

The first study focused on presentation biases, including position and context biases, which influence user interactions with ranked results. By leveraging propensity score estimation techniques, this study adjusted the probabilities of user actions to enable a more accurate and unbiased learning process. The second study advanced this work by introducing an unbiased deep neural network framework that incorporated a reweighting mechanism and a bias-correction layer to address implicit feedback biases. This approach improved the system's predictive capabilities by compensating for biases present in user-generated data.

The third study extended the scope to encompass biases from multimedia content, trust, and page layout in search result pages (SERPs). It introduced the Bias-Agnostic Whole-Page Unbiased Learning to Rank (BAL) algorithm, which used causal discovery methods to identify and correct hidden biases across multiple components of the page. Unlike traditional models limited to position bias, the BAL algorithm integrated causal inference to detect and mitigate more complex bias interactions, significantly enhancing the relevance and fairness of the rankings. These combined efforts represent a comprehensive approach to developing bias-aware systems, ensuring fairer and more effective IR and recommendation outcomes.

2) Graph-based Search Personalization: Personalized search has become increasingly important as data volumes grow and the need to provide users with highly relevant content becomes critical. This project explored graph-based approaches to search personalization, introducing innovative techniques to better capture and utilize user preferences across diverse data types and search contexts.

The first study developed a Latent Dirichlet Allocation (LDA)-based topic-graph probabilistic model for web search personalization. By modeling both user interests and disinterests using latent topics derived from user behavior (e.g., clicks and skips), the model provided a more nuanced approach to ranking search results. This work also introduced novel methods for refining negative user profiles, ensuring that non-interests were effectively captured and incorporated into the ranking process. The result was a system capable of delivering search results that align closely with users' preferences and query intents.

Recognizing the unique challenges of video search personalization, the second study proposed a Graph Neural Network (GNN)-based framework to enhance personalized video search. To the best of our knowledge, this is the first work analyzed real user behavior specific to video search. By integrating semantic representations of queries and video content with a hierarchical aggregation strategy, the model improved the quality of learned representations. The framework also utilized a multi-task learning approach, allowing it to incorporate additional signals to optimize model parameters. These innovations resulted in more accurate personalized video search rankings, addressing the distinct demands of video-centric search scenarios while advancing the broader field of search personalization.

3) Term Association Modeling In Probabilistic Information Retrieval: Traditional probabilistic information retrieval models, such as BM25, often assume that terms are independent, limiting their ability to account for relationships between terms. This project systematically addressed this limitation by developing enhanced models that incorporate term associations, proximity, and contextual relevance to improve retrieval performance.

The first study introduced the CRoss TErm Retrieval (CRTER) model, which incorporated the concept of ”Cross Terms” to model term proximity. In this approach, the occurrence of a query term was assumed to influence its surrounding text, with this influence attenuating as the distance increased. By integrating bi-gram and n-gram cross terms into basic probabilistic weighting models, CRTER captured the nuanced relationships between query terms and their proximity in the text. The results demonstrated significant improvements in retrieval effectiveness, particularly for queries with closely associated terms.

Building on this foundation, the project further enhanced proximity-based retrieval models by integrating contextual relevance into term proximity analysis. Traditional methods often treat all query term associations equally, neglecting the varying importance of different term associations. This study introduced measures to estimate contextual relevance between query terms using top-ranked documents. These measures were then integrated into a context-sensitive proximity model, allowing the retrieval system to distinguish and prioritize more meaningful term relationships. This refinement resulted in improved retrieval accuracy, addressing the limitations of conventional proximity-based methods and advancing the state of probabilistic information retrieval.

4) Enhance collaborative filtering by contrastive learning: This project tackled the limitations of collaborative filtering (CF) models, such as noise and sparse supervision, by incorporating contrastive learning techniques to redefine item representation learning.

The first study introduced a disentangled contrastive framework to enhance item representations by learning discriminative features. This approach addressed challenges related to noise and insufficient supervision in traditional CF models, resulting in significantly improved recommendation accuracy and robustness.

Building on this foundation, the second study proposed the Hypergraph Contrastive Collaborative Filtering (HCCF) framework, which leveraged self-supervised contrastive learning to enhance GNN-based CF models. This framework captured complex relationships among items and users, improving the discrimination power of the model. The HCCF framework demonstrated superior performance on benchmark datasets, even under conditions of sparse interaction data.

5) Machine Generated Text detection in a Multilingual Setting: In today's era of generative AI models determining data authenticity is a unique challenge. With the widespread use of openly available tools like chatGPT, gemini, metaAI etc., we now have machine generated text that is not only grammatically correct but is at par with humans. This poses a great question to accountability and attribution of the content available to us. The legitimacy of the data is a severe issue in areas such as academic integrity, misinformation, social media spamming, deepfakes etc. Thus we provide advanced methodologies to detect machine generated text in multilingual environment thus making the model more generalized and adaptable to current developments in LLMs.

We propose two unique approaches to solve this problem: a Pretrained language model (PLM) based approach and a Stylometric feature selection based method. The former is based on a multilingual PLM backbone to extract nuanced features as semantic embeddings which are inputs to a multi-layered bagged classification model. This leverages the deep semantic understanding between various languages thus providing a cohesive latent information representation layer. This strategy outperforms the state-of-the-art multilingual models for MGT detection.

The later is an approach where we curate popular and handcrafted multilingual features which are subject to rigorous selection strategy. These features are then fed to various sequential neural networks and ensemble classification methodologies to obtain a lightweight classifier that provides a great accuracy in low resource areas. Furthermore, this strategy succeeds all the non-transformer based methods by a decent margin. Our proposed methodologies consistently outperform popular MGT detection tools such as GPTZero, DetectGPT, etc.

6) E-Commerce Data Understanding: Understanding data in e-commerce is critical for enhancing customer experiences, optimizing inventory, and improving business operations. This project explored innovative frameworks and models to address key challenges in understanding e-commerce queries, product categorization, and purchase prediction.

The first study proposed the Dynamic Product-aware Hierarchical Attention (DPHA) framework for understanding e-commerce query intents. By leveraging dynamic session information, including prior queries and clicked product categories, DPHA maps customer queries into relevant product categories. Using hierarchical attention mechanisms, it learns both query-level and session-level representations, significantly improving the classification of complex queries in dynamic shopping contexts.

To address challenges in product categorization, the project introduced the Neural Product Categorization (NPC) model. This model focuses on fine-grained categorization, tackling issues like blurred category boundaries and evolving product definitions. By employing character-level convolutional embedding and spiral residual layers, NPC extracts intricate context representations from product content. The model is trained with weak labels derived from customer behavior logs and outperforms traditional methods in classifying products into fine-grained categories.

Lastly, the project developed the Graph Multi-Scale Pyramid Networks (GMP) framework to predict users' future purchases. GMP captures multi-resolution temporal patterns and dynamic dependencies among product categories. By using a multi-scale pyramid modulation network, recalibration gating, and a context-graph neural network, GMP models hierarchical temporal factors and dynamic category dependencies. Real-world experiments demonstrated GMP's ability to outperform state-of-the-art predictive models, providing businesses with more accurate insights into customer purchase behavior.

7) Pattern Recognization: Pattern mining is essential in uncovering meaningful insights from large datasets, enabling the identification of complex relationships and trends that are otherwise not readily apparent. This project explored novel methodologies for mining and interpreting patterns in diverse domains, contributing to advancements in understanding dynamic data relationships.

The first study introduced diverging patterns, a new type of contrast pattern that highlights significant frequency changes between two datasets moving in opposite directions. To quantify these changes, the project proposed a diverging ratio and represented patterns with a four-dimensional vector. An efficient algorithm was developed to mine these diverging patterns, revealing insights that traditional frequent pattern mining approaches often overlook. This method provided a deeper understanding of contrasting trends, enabling the discovery of novel and actionable insights across various applications.

The second study focused on mining associations within multivariate time-series data, with a particular emphasis on gene expression analysis. By identifying significant changes and measuring marginal change rates, the project employed a propositional confirmation-guided rule discovery method to uncover associations among temporal changes. This approach facilitated the construction of gene interaction networks and clustering of genes with similar temporal behavior, offering valuable insights into biological processes and enabling a more comprehensive understanding of gene regulation dynamics. These methodologies demonstrate the potential of advanced pattern recognition techniques to drive discovery in complex, data-rich domains.

8) Data Understanding and Prediction for Healthcare : This research focuses on advancing healthcare by developing data-driven solutions that utilize state-of-the-art machine learning and Bayesian inference techniques. The aim is to enhance personalized medical recommendations and diagnostic accuracy by leveraging diverse data types and innovative methodologies.

The first study introduced a Bayesian-based prediction model for recommending personalized medical tests to patients. By analyzing patterns in medical data through Bayesian inference, the model integrates contextual factors such as timing and interaction types to deliver more accurate predictions. This approach is particularly effective for sparse datasets and providing a robust tool for personalized healthcare recommendations.

The second study developed a deep learning approach for skin cancer classification that combines image data with structured clinical information. Utilizing convolutional neural networks (CNNs) to extract features from skin lesion images, the model incorporates patient-specific data such as age, gender, and lesion history to enhance diagnostic precision. This multimodal approach significantly outperforms image-only methods, demonstrating the benefits of integrating multiple data sources for comprehensive and reliable medical diagnoses.

Together, these studies highlight the transformative potential of advanced data analysis in improving personalized healthcare and medical decision-making, paving the way for more effective and patient-centered solutions in the field.

9) Fraud Detection in E-commerce: Fraud activities, such as spam reviews and fake shopping behaviors, disrupt customer decision-making and harm business reputation. The project introduced a novel approach using heterogeneous graph neural networks (GNNs) to detect fraud more effectively by modeling complex relationships among diverse data types, including users, items, devices, and reviews.

The proposed framework, C-FATH, tackled critical challenges in fraud detection, such as structural inconsistencies (e.g., fraudsters often being surrounded by legitimate users) and content inconsistencies (e.g., disparate features across product categories). By employing community-based filtering, the framework grouped nodes exhibiting similar behaviors, while similarity-based sampling ensured that only relevant neighbors were considered during fraud detection. These methods enhanced the system's ability to accurately identify fraudulent entities. Combining heterogeneous graph modeling with embeddings from homogeneous graphs and leveraging a multi-task learning strategy, the framework achieved superior results compared to traditional methods and existing GNN-based models.

IV. Partnerships and Collaborations

We actively engage in collaborations with industry partners to conduct research that addresses the practical needs and interests of users, as well as with academic institutions to advance cutting-edge, theoretical research. Our current and past codllaborating organizations are listed as follows:

- Baidu Search Science Lab
- JD Data Science Lab
- Yahoo! Labs
- York Unviersity
- Unviersity of Waterloo
- Beijing Technology and Business University

V. Funding & Acknowledgments

This lab is currently supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant, Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Early Career Supplement Grant, and Wilfrid Laurier University Start-up Grant.

VI. Vision for the future

The overarching vision for these projects is to pioneer the development of intelligent, data-driven systems that transform how we interact with and benefit from information across diverse domains. By leveraging cutting-edge machine learning, artificial intelligence, and innovative LLM-based modeling techniques, the future lies in creating adaptive, personalized, and scalable solutions that address complex real-world challenges.

We aim to advance the understanding of dynamic patterns, relationships, and behaviors within data, whether in healthcare, e-commerce, information retrieval, or recommender systems. Future systems will seamlessly integrate diverse data types, handle sparsity and noise with resilience, and adapt to evolving contexts to offer insights and recommendations that are precise, fair, and impactful.

Looking forward, the focus will be on creating interpretable, ethical, and human-centric AI systems that not only enhance decision-making but also foster trust and inclusivity. By uniting advancements in graph-based learning, contrastive techniques, and deep neural networks, we envision a future where technology empowers individuals and organizations to make smarter decisions, uncover hidden insights, and improve quality of life across sectors. This vision aims to shape the next generation of intelligent systems, driving innovation, fostering collaboration, and solving the most pressing problems in an increasingly data-rich world.

VII. Representative Publications

- He, L., Zhao, J., Gu, Y., Elbaz, M., & Ding, Z. (2024, January). A bias study and an unbiased deep neural network for recommender systems. In *Web Intelligence* (Vol. 22, No. 1, pp. 15-29). IOS Press.
- Chu, X., Hao, C., Wang, S., Yin, D., Zhao, J., Zou, L., & Li, C. (2024, May). LT 2 R: Learning to Online Learning to Rank for Web Search. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)* (pp. 4733-4746). IEEE.
- Li, Q., Su, L., Zhao, J., Xia, L., Cai, H., Cheng, S., ... & Yin, D. (2024, March). Text-Video Retrieval via Multi-Modal Hypergraph Networks. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining* (pp. 369-377).
- Mao, H., Zou, L., Zheng, Y., Tang, J., Chu, X., Zhao, J., ... & Yin, D. (2024, May). Whole Page Unbiased Learning to Rank. *In Proceedings of the ACM on Web Conference 2024* (pp. 1431-1440).
- Ren, X., Xia, L., Zhao, J., Yin, D., & Huang, C. (2023, July). Disentangled contrastive collaborative filtering. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1137-1146).
- Shao, W., Chen, X., Zhao, J., Xia, L., Zhang, J., & Yin, D. (2023, November). Sequential recommendation with user evolving preference decomposition. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region* (pp. 253-263).
- Zhao, J., Huang, J. X., Deng, H., Chang, Y., & Xia, L. (2021). Are topics interesting or not? An LDA-based topic-graph probabilistic model for web search personalization. *ACM Transactions on Information Systems (TOIS)*, 40(3), 1-24.
- Chu, X., Zhao, J., Fan, X., Yao, D., Zhu, Z., Zou, L., ... & Bi, J. (2022, April). Contrastive Disentangled Graph Convolutional Network for Weakly-Supervised Classification. In *International Conference on Database Systems for Advanced Applications* (pp. 722-730). Cham: Springer International Publishing.
- Zhao, B., Yang, B., Li, Z., Li, Z., Zhang, G., Zhao, J., ... & Bao, H. (2022, October). Factorized and controllable neural re-rendering of outdoor scene for photo extrapolation. In *Proceedings of the 30th ACM International Conference on Multimedia* (pp. 1455-1464).
- Xia, L., Huang, C., Xu, Y., Zhao, J., Yin, D., & Huang, J. (2022, July). Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval* (pp. 70-79).
- Chu, X., Zhao, J., Zou, L., & Yin, D. (2022, July). H-ERNIE: A multi-granularity pre-trained language model for web search. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval* (pp. 1478-1489).
- Wei, W., Huang, C., Xia, L., Xu, Y., Zhao, J., & Yin, D. (2022, February). Contrastive meta learning with behavior multiplicity for recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining* (pp. 1120-1128).

- Zhang, L., Shi, L., Zhao, J., Yang, J., Lyu, T., Yin, D., & Lu, H. (2022, February). A gnn-based multi-task learning framework for personalized video search. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (pp. 1386-1394).
- Huang, C., Chen, J., Xia, L., Xu, Y., Dai, P., Chen, Y., ... & Huang, J. X. (2021, May). Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 35, No. 5, pp. 4123-4130).
- Wang, L., Li, P., Xiong, K., Zhao, J., & Lin, R. (2021, October). Modeling heterogeneous graph network on fraud detection: A community-based framework with attention mechanism. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 1959-1968).
- Huang, C., Zhao, J., & Yin, D. (2021, April). Purchase intent forecasting with convolutional hierarchical transformer networks. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)* (pp. 2488-2498). IEEE.
- Wu, X., Chen, H., Zhao, J., He, L., Yin, D., & Chang, Y. (2021, March). Unbiased learning to rank in feeds recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (pp. 490-498).
- Jian, F., Huang, J. X., Zhao, J., Ying, Z., & Wang, Y. (2020). A topic-based term frequency normalization framework to enhance probabilistic information retrieval. *Computational Intelligence*, 36(2), 486-521.
- Soboroff, I. (2021). Overview of TREC 2021. In *TREC*.
- Zhao, J., Chen, H., & Yin, D. (2019, November). A dynamic product-aware learning model for e-commerce query intent understanding. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 1843-1852).
- Chen, H., Zhao, J., & Yin, D. (2019, November). Fine-grained product categorization in e-commerce. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 2349-2352).
- Huang, C., Wu, X., Zhang, X., Zhang, C., Zhao, J., Yin, D., & Chawla, N. V. (2019, July). Online purchase prediction via multi-scale modeling of behavior dynamics. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2613-2622).
- Huang, C., Zhang, C., Zhao, J., Wu, X., Yin, D., & Chawla, N. (2019, May). Mist: A multiview and multimodal spatial-temporal learning framework for citywide abnormal event forecasting. In *The world wide web conference* (pp. 717-728).
- Huang, J. X., He, B., & Zhao, J. (2018). Mining authoritative and topical evidence from the blogosphere for improving opinion retrieval. *Information Systems*, 78, 199-213.
- Jian, F., Huang, J. X., Zhao, J., & He, T. (2018, June). A new term frequency normalization model for probabilistic information retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (pp. 1237-1240).
- Miao, J., Huang, J. X., & Zhao, J. (2016). TopPRF: A probabilistic framework for integrating topic space into pseudo relevance feedback. *ACM Transactions on Information Systems (TOIS)*, 34(4), 1-36.
- Jian, F., Huang, J. X., Zhao, J., He, T., & Hu, P. (2016, July). A simple enhancement for ad-hoc information retrieval via topic modelling. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 733-736).
- Zhao, J., Huang, J. X., & Ye, Z. (2014). Modeling term associations for probabilistic information retrieval. *ACM Transactions on Information Systems (TOIS)*, 32(2), 1-47.
- Zhao, J., & Huang, J. X. (2014, July). An enhanced context-sensitive proximity model for probabilistic information retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (pp. 1131-1134).
- Zhao, J., Huang, J. X., Hu, X., Kurian, J., & Melek, W. (2012, October). A Bayesian-based prediction model for personalized medical health care. In *2012 IEEE International Conference on Bioinformatics and Biomedicine* (pp. 1-4). IEEE.
- Zhao, J., Huang, J. X., & Wu, S. (2012, August). Rewarding term location information to enhance probabilistic information retrieval. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval* (pp. 1137-1138).
- Zhao, J., Huang, J. X., & He, B. (2011, July). CRTER: using cross terms to enhance probabilistic information retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (pp. 155-164).
- Lupu, M., Zhao, J., Huang, J. X., Gurulingappa, H., Fluck, J., Zimmermann, M., ... & Tait, J. (2011, November). Overview of the TREC 2011 Chemical IR Track. In *TREC*.
- Rohian, H., An, A., Zhao, J., & Huang, X. (2009, November). Discovering temporal associations among significant changes in gene expression. In *2009 IEEE International Conference on Bioinformatics and Biomedicine* (pp. 419-423). IEEE.
- An, A., Wan, Q., Zhao, J., & Huang, X. (2009, November). Diverging patterns: discovering significant frequency change dissimilarities in large databases. In *Proceedings of the 18th ACM conference on Information and knowledge management* (pp. 1473-1476).
- Zhao, J., Huang, X., Ye, Z., & Zhu, J. (2009). York University at TREC 2009: Chemical Track. In *TREC*.
- Gu, Y., Zhao, J., Liang, D., & Xu, Z. (2007, September). Immunity diversity based multi-agent intrusion detection. In *2007 IEEE Congress*

on Evolutionary Computation (pp. 3404-3409). IEEE.

- Zhao, J., Huang, J. X., & Hu, X. (2013). BPLT+: A Bayesian-based personalized recommendation model for health care. *BMC genomics*, 14, 1-10.
- Gu, Y., Zhou, B., & Zhao, J. (2008). PCA-ICA ensembled intrusion detection system by pareto-optimal optimization. *Information Technology Journal*, 7(3), 510-515.
- Yu, G., Jiashu, Z., & Tianjun, Z. (2006). Distributed Intrusion Detection Method Based on the Diversity of Immunity. *JOURNAL-XIAN JIAOTONG UNIVERSITY*, 40(10), 1052.
- Singh, S., & Zhao, J. J. (2022, November). A deep skin cancer classification approach using image and structured information. In *2022 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)* (pp. 930-933). IEEE.
- Zhao, J., McGrath, S., Huang, J. X., Wu, J., & Wu, S. (2018). Extracting Relevant Information from Big Data to Anticipate Forced Migration. *Highlighting the Importance of Big Data Management and Analysis for Various Applications*, 71-78.

Contact Information

Director:
Dr. Jiashu Zhao
E-mail: jzhao@wlu.ca
Address: 75 University Ave W,
Waterloo, ON N2L 3C5

Website:
https://inspire-research.github.io

A Systematic Survey on Math Word Problem Solvers Based on Large Language Models

Yicong Liang[1] and Debby D. Wang[1]

[1]School of Science and Technology, Hong Kong Metropolitan University, Hong Kong SAR
yliang@hkmu.edu.hk

***Abstract*— Nowadays, there is growing interest in improving the reasoning capabilities of large language models (LLMs), represented by designing an LLM-based solver for math word problems (MWPs). This survey provides a comprehensive overview of recent LLM- based methods that aim to solve MWPs. In particular, we first introduce some preliminaries on MWPs and their connection with LLMs. Then, we examine the capabilities of each reviewed method by analyzing its architecture and prompting technique. Finally, we discuss the limitations of LLM-based MWP solvers and provide potential directions for future research.**

***Index Terms*— Large Language Model, Math Word Problem, Prompt Engineering, Mathematical Reasoning**

I. INTRODUCTION

The development of automatic AI systems for math word problems has a long-standing history, dating back to the 1960s [1], [2]. A tool that can generate step-by-step solutions to math word problems has the potential to offer personalized guidance for students and assist educators in curriculum development. However, automatically solving math word problems (MWPs) is challenging, as the solver needs to combine arithmetic skills with commonsense reasoning.

Before pre-trained large language models revolutionized most NLP tasks, some small-scaled models with handcrafted neural networks were proposed to solve MWPs. For example, previous studies [3], [4] consider MWP as a generation task and usually leverage LSTM-based sequence-to-sequence models to learn the mapping from source sequences (i.e., question texts) to target sequences (i.e., math expressions). These neural MWP solvers without using pretrained language models (PLMs) have been surveyed in [5]. However, the main disadvantage of previously proposed neural solvers is that they must be trained from scratch for different MWP datasets, making them unscalable for other downstream tasks.

In recent years, language models (LMs) have reshaped the landscape of the NLP field and demonstrated impressive performance across diverse downstream tasks [6]. The pretrained models, such as BERT [7], RoBERTa [8], DeBERTa [9], BART [10] and GPT [11], have learned world knowledge by parsing a vast amount of texts, which benefits the question answering (QA) task consequently (e.g. commonsense QA [12], answering math word problems [6] and assisting with theorem proving [13]).

Large language models (LLMs) have an improved performance on diverse NLP downstream tasks. However, scaling up the size of language models alone has not demonstrated their effectiveness on some mathematical reasoning tasks, such as MWPs and theorem proving [14]. The dataset GSM8K [15] containing step-by-step reasoning is proposed to evaluate the LLM's reasoning ability by checking the effectiveness of its generated natural language solutions. Cobbe et al. [15] proposed to finetune GPT3 [6] on GSM8K to help the language model generate multi-step rationales and train a neural component to verify the correctness of the model-generated solution. This generate-and-verify framework could enhance the model's capacity of generating accurate answers.

Recently, some studies [16], [17], [18] have reported that language models can demonstrate the emergent ability of performing complex multi-step reasoning tasks when they are large enough (e.g., over 100B parameters). In particular, the breakthrough method, chain-of-thought (CoT) [16] prompting strategy, can unlock the reasoning ability of LLMs when provided with a few examples without any parameter update. A series of intermediate natural language reasoning steps is generated before giving the final answer.

II. MATH WORD PROBLEM

A. Difference between MWP and other QA task

Machine reading comprehension is one of the central tasks in natural language understanding, especially for the task of question answering (QA). In the context of knowledge-based QA tasks, the system leverages knowledge-aware methods to respond with the corresponding answer, e.g., knowledge triple retrieval [19].

The MWP task can be considered as a special case of QA task. However, the MWP task is different from the traditional text-based QA tasks with the following challenges: (1) A MWP needs to parse the human-readable words into machine-understandable mathematical logic to perform quantitative reasoning; (2) A MWP requires complex reasoning scenarios, and MWP solvers need to be capable of mathematical calculation and mathematical reasoning; (3) Unlike natural language understanding, MWPs usually have a single correct numeric answer, which increases the difficulty for the solver to accurately generate the solution.

Math reasoning tasks include arithmetic problems and math word problems. Arithmetic problems mainly correspond to mathematical calculation consisting of arithmetic representation and arithmetic calculation [20]. However, there exist some differences between arithmetic problems and math word problems. In particular, arithmetic problems focus on pure mathematical operations and numerical manipulation, where the input problems rarely contain semantic textual elements.

On the other hand, math word problems are usually presented as verbal descriptions instead of explicit mathematical equations in arithmetic problems. In addition, MWP is related to the task of mathematical reasoning, where the solver can interpret and generate step-by-step natural text before giving the final answer.

Traditional span-based methods of extractive question-answering tasks (e.g., SQuAD [21]) cannot be directly applied to solve MWPs since the answer is usually the result of some computation and is generally not a span in the question or context. A MWP solver generates a numerical math expression and feeds this expression to an external symbolic calculator to obtain the final answer.

B. Preliminary

The math word problems include the following main components: (1) A textual description related to the math problem; (2) A set of known quantities mentioned in the problem text; (3) An unknown quantity whose value needs to be solved. In addition, MWP can be further grouped into the different levels: (1) one-unknown variable to be solved, or multi-unknown variables; (2) linear equations or non-linear equations. In this survey, the reviewed methods only aim to solve math problems solved by linear equations towards one unknown variable.

The solution to the problem denoted as A can be represented in the following format: (1) A single numerical value, e.g., GPT3 [6] leverages standard prompting to output the final answer for a given MWP question; (2) A mathematical expression, as an input to an external tool like Python calculator [22]; (3) a series of textual reasoning steps including the final numeric answer, e.g., chain-of-thought prompt [16] requires LLM to generate rationales before giving the final answer. Previous studies [23], [24] address to generate the solution to a math word problem as a mathematical expression, and these models try to map the problem text to a symbolic space with numbers and operators.

We introduce preliminaries of generating solutions to MWPs with LM prompting based on large language models, and the notations for modeling are listed in Table I. Mathematically, a math word problem is represented in the form of a text sequence $< w1,\ w2,\ \cdots,\ wn >$. There are some known quantities mentioned in the text and one unknown variable that the system needs to solve. How to extract the quantities is a preprocessing problem, and some works simply adopt the string pattern matching method [25] to recognize the numeric entries.

Based on the backbone of large language models, solving MWP can be transformed into text generation tasks. Therefore, the language model objective can be used in the MWP task for solution generation. In the following, we will introduce how to incorporate a pretrained language model to solve the MWP task.

Vanilla QA:

$$p(\mathcal{A}|\mathcal{Q}) = \prod_{i=1}^{|\mathcal{A}|} p(a_i|\mathcal{Q}, a_{<i}) \tag{1}$$

Given a textual question Q, the solution in terms of a single numeric answer can be generated by the LLM. In addition, a prefixed prompt will be added before generating the final answer, e.g., “The answer is ” [6].

In-context learning:

$$p(\mathcal{A}|\mathcal{Q}, \mathcal{D}) = \prod_{i=1}^{|\mathcal{A}|} p(a_i|\mathcal{Q}, \mathcal{D}, a_{<i}) \tag{2}$$

TABLE I
NOTATIONS USED IN THIS PAPER

Notations	Descriptions
$\mathcal{Q}$	a question related to MWP
w	a token in the text sequence
$\mathcal{A}$	a textual solution to the question
$\mathcal{Y}$	a numerical answer of the question
c	one reasoning path
$\mathcal{D}$	A set of exemplars for in-context learning

To enhance the quality of the response from the LLM, in-context learning (ICL) is incorporated into the natural language processing (NLP) tasks. The ability of analogy with respect to ICL is embedded in LLMs, which can be learned from a few examples D in the context [26]. Each exemplar consists of one question and its corresponding solution. Meanwhile, a list of exemplars is concatenated to form the demonstration prompt, which is fed into the LLM as an augmented context for prediction on the testing question. The improvement by leveraging ICL relies on the training stage and inference stage [26], but this survey mainly focuses on the inference stage for the MWP task.

Reasoning-enhanced QA:

$$p(\mathcal{A}|\mathcal{D}, \mathcal{Q}) = \sum_{c} p(\mathcal{A}, c|\mathcal{D}, \mathcal{Q}) = \sum_{c} p(\mathcal{A}|\mathcal{D}, \mathcal{Q}, c)p(c|\mathcal{D}, \mathcal{Q}) \tag{3}$$

Scaling up the size of LLMs can elicit some reasoning abilities that can improve the accuracy of MWP solvers by generating a reasoning path c before giving the final numeric answer. In particular, reasoning abilities can be unlocked by prompting engineering [16], which generates a solution to MWP in the step-by-step format.

III. Surveyed Methods

The principles for selecting reviewed papers in this work are as follows: (1) The proposed methods are based on the pretrained language models (e.g., GPT-3 [6], Codex [28]) as their fundamental backbone, as shown in Table II; (2) The tasks aim to solve math word problems, e.g., the datasets in their experiments are related to MWP; (3) The reviewed articles are published in top venues or with a citation count over 50. The surveyed models are listed in Table III.

This survey first analyzes the compared models to determine whether the parameters of their foundation language models are updated. Accordingly, the reviewed methods are grouped into two mainstreams: finetune-based and prompt-based methods.

A. Finetune-based Models

Numerical reasoning skills are challenging when the LMs are only trained on the objective of the vanilla language model. Geva et al. [23] proposed a multi-task training strategy to inject mathematical reasoning skills into PLMs. Notably, the proposed framework GenBERT [23] incorporates automatic data generation in the pretraining task and is trained on textual data in the form of question-passage pairs. GenBERT is a BERT-based model for generating arbitrary output tokens. It can handle the extractive QA task, where the answer is a text span among the question or context and contains a generative head that can output the numeric answer.

The MWP solver trained on the language model objective may make mistakes in generating mathematical expressions. The models trained to learn the mapping from problem text to math expressions may have unsatisfactory performance because it is difficult for them to learn to distinguish between ground-truth and predictive expressions with minor mistakes [24]. To handle this limitation, Shen et al. [24] additionally introduce a ranker to explicitly train the model to distinguish between accurate and inaccurate expressions. The proposed model Generate & Rank [24] within transformer-based encoder-decoder architecture BART [10] first generates expression candidates and then ranks the candidates to make the final prediction.

It is challenging for autoregressive models to accomplish mathematical reasoning tasks since they cannot correct their errors when a generation is generated. Similar to the idea in [24] that the solutions generated need to be evaluated, Cobbe et al. [15] proposed training verifiers to check the correctness of the candidate solutions. In addition, the proposed model [15] developed reasoning steps in natural language such that the produced solutions were more interpretable by humans instead of producing a math expression in [24]. The generator based on GPT3 [6] is finetuned on a curated math dataset GSM8K [15] for generating rationales to form the full natural language solution. The experimental results suggest that it is essential to allow the MWP solver to generate a natural language solution

TABLE II
Pretrained Language Models used in math word problem.

Pretrained LM	Size
GPT2 [11]	1.5B
GPT3 [6]	175B
GPT-J [27]	6B
Codex [28]	12B
PaLM [17]	540B
BERT [7]	340M
BART [10]	406M
DeBERTa [9]	304M

TABLE III
Surveyed papers.

Model	Venue	Language model	Parameters update	External component
GPT3 [6]	Arxiv	GPT3	No	Nill
CoT [16]	NeurIPS	GPT3	No	Nill
Zeroshot-CoT [29]	NeurIPS	GPT3	No	Nill
Auto-CoT [30]	ICLR	GPT3	No	Sentence-BERT
PAL [31]	ICML	Codex	No	Nill
PoT [32]	TMLR	Codex	No	SymPy
Self-consistency [18]	ICLR	PaLM	No	Nill
Least-to-most [33]	ICLR	GPT3	No	Nill
MathPrompter [25]	ACL	GPT3	No	Calculator
DECLARATIVE [22]	NeurIPS	Codex	No	SymPy
Plan-and-Solve [34]	ACL	GPT3	No	Nill
Complexity [35]	ICLR	GPT3	No	Nill
GenBERT [23]	ACL	BERT	Yes	Nill
Generate & Rank [24]	EMNLP	BART	Yes	Trained Ranker
Verifier [15]	Arxiv	GPT3	Yes	Trained Verifer
STaR [36]	NeurIPS	GPT-J	Yes	Nill
DIVERSE [37]	ACL	DeBERTa	Yes	Trained Verifier
CoRe [38]	ACL	GPT-J	Yes	Trained Verifier

before giving the final answer, and the performance dropped dramatically if directly outputting a final numeric answer without any intermediate steps [15].

Generating intermediate reasoning steps improves LM performance on complex tasks like MWP, but fine-tuning the generator requires massive training examples with rationales. To address this limitation, Zelikman et al. [36] developed a self-taught reasoner (STaR) by iteratively bootstrapping the reasoning ability to generate rationales. In particular, starting with a small prompt dataset that contains intermediate rationales, StaR adopts in-context prompting to annotate each example in the large dataset for further finetuning the language model [36]. To improve the robustness, rationalization is applied [36] to reason backward for problems that the model fails to solve, i.e., given the correct answer as a hint, let the model generate the rationale accordingly. Finally, the finetuning process will be repeated on the enlarged dataset with previously generated rationales.

Few-shot learning for solving MWP is a challenging task that requires the LMs to elicit intermediate reasoning steps. Even equipped with the LLMs like GPT3 (175B) [6] and PaLM (540B) [17], the reasoning abilities are still limited. To further improve the reasoning capacity of PLMs, Li et al. [37] designed a diverse verifier to aggregate different sampled reasoning paths to solve the MWP. Specifically, the proposed model DIVERSE [37] first samples different demonstration exemplars for constructing diverse prompts and then feeds them OpenAI PLMs (e.g., text-davinci-002) to generate various reasoning paths. Second, DIVERSE trains a step-aware verifier by finetuning DeBERTa [9] to score the quality of each path and uses a weighted voting mechanism [37] to obtain the final answer.

Directly prompting PLMs to solve MWPs often does not yield satisfactory results, as the generation process lacks the level of supervision and adaptivity that humans possess. Zhu et al. [38] argued that the human-like reasoning framework could be modeled using a dual system approach, with a generator for immediate reactions and a verifier for more nuanced reasoning. Their proposed model CoRe [38] leverages the direct interaction between the generator and verifier to improve the generalization ability of LLMs. First, the verifier can provide reliable feedback to supervise the generator for rationale generation. Second, the verifier leverages Monte Carlo Tree Search (MCTS) [39] to score the tokens of reasoning paths produced by the generator. Finally, the proposed strategy of self-thinking can provide informative self-produced data to teach the generator and verifier.

B. Prompt-based Models

Based on the transformer-based framework with self-attention technique [40], numerous downstream tasks have been transferred into text generation problems by following the "pretrain, prompt and predict" paradigm. In this paradigm, instead of finetuning the PLMs to adapt to a new task, solving text-based tasks is reformulated to the original language model pretraining with the help of an appropriate textual prompt [41]. Inspired by the idea that humans can perform a new language task from a few demonstration examples or simple task instructions, Brown et al. [6] leverage in-context learning within few-shot settings and prompt the language model GPT3 to solve the target tasks by providing some task examples as additional context during the inference stage without any gradient update [6]. The significant advancement of this setting is that the users can prompt the model with a few input-output demonstration exemplars instead of finetuning a separate LM checkpoint for each new task. The experimental results [6] show that scaling up LMs greatly improves task-agnostic, few-shot performance.

The critical limitation of a finetune-based MWP solver is that it is expensive to collect a large set of training data with high-quality rationales. The traditional few-shot prompting technique used in [6] has been successful for a series of simple QA tasks but performs poorly on tasks requiring reasoning ability even with increasing LM scale [42]. Wei et al. proposed chain-of-thought prompting [16] to elicit the LLMs' reasoning ability for complex tasks, e.g., commonsense reasoning and arithmetic. This approach involves a sequence of intermediate reasoning steps expressed in natural language, leading to the final output. CoT prompting can be considered as a special type of in-context learning where each exemplar includes the reasoning thought process instead of just a single final answer. The experimental results suggest that CoT prompting improves performance by allowing the sequential reasoning steps embodied in the generation [16].

Pretrained large language models are well-known as excellent few-shot learners with task-specific exemplars and can generate complex rationales via step-by-step solution examples. Kojima et al. [29] show that PLMs are also decent zero-shot learners by leveraging a simple but effective prompt "Let's think step by step" before giving the final answer. The proposed model Zero-shot-CoT [29] does not require handcrafted task-specific exemplars and outperforms zero-shot LLMs on diverse downstream reasoning tasks, e.g., arithmetic math word problems.

According to the prompting design regarding the number of exemplars, CoT prompting can be classified into two significant paradigms: few-shot CoT and zero-shot CoT. In general, few-shot CoT with task-specific rationales demonstrations outperforms zero-shot CoT in most cases. However, manually constructing exemplars with step-by-step reasoning chains for a specific task or even each testing question is nontrivial. Randomly or heuristically selecting in-context examples in CoT prompting may have a high risk of unstable performance in reasoning tasks. To address this limitation, Zhang et al. [30] proposed automatically constructing demonstrations with questions and reasoning chains instead of handcrafting in-context exemplars. Prompting LLM to generate reasoning chains for each exemplar directly often comes with mistakes. The proposed solver Auto-CoT [30] increases the diversity of demonstration questions to help the LLM lower the mistakes in generating reasoning chains. Particularly, the questions are clustered based on their representation obtained by Sentence-BERT [43], and a representative question from each cluster is selected.

LLMs have demonstrated their effectiveness in diverse downstream reasoning tasks by using CoT and in-context learning. However, LLMs often make arithmetic mistakes in the solutions, even if the generated rationales are logically correct. Program-Aided Language model (PAL) [31] addressed the underlying problem that LLMs often struggle with performing arithmetic operations and proposes to use an external tool (i.e., Python program) to deal with the calculation work. PAL leverages the LLM (i.e., Codex [28]) to read a testing problem and then generates a program [31] instead of natural language rationales [16], [29], [30] as intermediate reasoning steps to assure the calculation accuracy by using the Python interpreter.

Previous methods [16], [30], [29] output the chain-of-thought reasoning steps in natural language and let the PLMs do reasoning and computation jobs simultaneously. Chen et al. [32] argued that LMs could express reasoning steps as programs in a few lines of code, and the external language interpreter (e.g., Sympy [44]) can finish the computation work. The proposed model Program of Thoughts (PoT) decouples complex computation from reasoning and language under- standing. To compare PoT with vanilla CoT [16] and Zero-shot CoT [29], PoT leverages the program of thoughts in each exemplar and does not require an extra step to extract the answer from the reasoning steps since the generated program can be executed by the interpreter to return the final answer. Intuitively, there exist multiple different ways of solution leading to its unique correct answer for a complex reasoning problem. Inspired by that, Wang et al. [18] designed a novel decoding strategy, self-consistency, to sample multiple solutions instead of greedily decoding only one as used in vanilla CoT [16]. In particular, self-consistency first samples a diverse set of reasoning paths and then aggregates them by using a majority voting scheme to select the most consistent answer. Compared to the sample-and-rank methods [15], [37], self-consistency does not require training an additional component or finetuning the backbone LM to verify the correctness probability of the generated solution.

In-context learning and CoT have been widely adopted in prompting engineering to help LLMs solve various reasoning tasks. However, LLMs will have poor performance in the case when the examples in the ICL demonstration prompt are easier than the testing problem[1]. To address this issue, Zhou et al. [33] introduced a new prompting strategy, Least-to-most, to help LLMs improve easy-to-hard generalization. Specifically, least-to-most decomposes a complex problem into a series of simpler subproblems and then solves them sequentially. In both stages, the decomposition and subproblem solving are accomplished by the LM without any parameter update. In addition, Least-to-most can be incorporated with ICL and CoT to further enhance the performance in reasoning tasks.

In the zero-shot setting, Zero-shot CoT [29] has demonstrated its remarkable performance in mathematical reasoning tasks. However, Imani et al. [25] pointed out two limitations in CoT-based prompting methods: (1) lack of checking the validity of reasoning steps; (2) lack of confidence in the predictions. The proposed model MathPrompter [25] first transforms the question into an Algebraic template with value-variable mapping and then sends it to the LLM to generate two different solutions in Algebraic and program ways for cross-checking. Afterward, MathPrompter evaluates the two generated solutions using multiple randomly selected values to check consensus among the answers. Repeat the above steps several times to extract the most frequent value observed for the answer.

Some methods [31], [32] offload the calculation to a language interpreter to eliminate the arithmetic error that LLMs often make in generating the solutions. However, these program-based models favor those problems with simple procedures and are less effective for problems requiring declarative reasoning. He-Yueya et al. [22] proposed the approach DECLARATIVE to perform mathematical declarations. DECLARATIVE prompts the LLM to formalize MWPs as a set of variables and equations incrementally and solves the equations by passing them to Sympy [44].

Zero-shot CoT can lower the effort to manually handcrafted step-by-step reasoning exemplars, but it still suffers from missing-step and calculation errors. To address the missing-step limitation, Wang et al. [34] manually craft the Plan-and-solve prompt to guide the LM to devise a plan to decompose the entire task into several subtasks and solve each subtask sequentially. To address the arithmetic limitation, the researchers add a detailed instruction prompt [34] to ask the LM to pay more attention to variables and calculation results.

Demonstration exemplars with reasoning steps can improve the prediction performance for new inputs. However, different exemplars may influence the testing question differently when making inferences. Which reasoning exemplars make the most the most effective in-context learning prompts becomes a critical question. Fu et al. [35] proposed complexity-based prompting, a novel example selection scheme, for CoT multi-step reasoning. Specifically, select complex instances[2] with CoT reasoning steps in the in-context learning prompt before the testing question. The experimental results suggest that selecting complex questions as in-context learning exemplars improves the performance on math word reasoning tasks [35].

[1] A MWP can be simply measured its difficulty by the number of solving steps [33].

[2] The complexity indicator is the number of steps to solve the question.

IV. Challenges and Future Directions

When the large language models (e.g. GPT [45] and LLaMA [46]) are released to the public, the trends show that fewer models will be proposed for outputting math expression only, and more research will focus on leveraging the LLMs directly to generate a natural language solution for MWPs, including the rationales and numeric final answer. The possible reason may be that, due to the difference between natural language text sequences and mathematical expressions, one minor mistake will change the semantics and lead to an incorrect answer, whereas natural language generation is more robust to these tiny mistakes [24]. In addition, the performance of MWP solvers for generating mathematical expressions will only degrade quickly when the expression gets longer [4].

Generating stepwise rationales can enhance the performance of language models on complex reasoning tasks. However, inducing the rationale generation from LMs requires constructing a large amount of data containing reasoning steps. It is very expensive to construct such datasets manually to finetune the LLMs. To the best of our knowledge, only GSM8K [15] and MATH [14] provide full step-by-step solutions to finetune the LLMs, in order to generate the solution for a testing problem. In addition, the language model finetuned on one specific dataset may not generalize well on another MWP dataset.

Another line of method to elicit LLMs to generate reasoning steps automatically is to adopt some prompting strategies, including instructions, trigger sentences, and in-context learning. LLMs have shown promising performance in solving new reasoning problems by simply conditioning on a few demonstration examples (e.g., few-show learning). However, small variations in prompt configuration have been known to affect few-shot performance dramatically [35]. Handcrafting instructing prompts for different reasoning MWP datasets needs much expertise and annotation work. The order of exemplars listed in the demonstration, the complexity of problems provided in in-context learning, and the relation (e.g., similarity) between demonstration examples and the testing problem may affect the prediction performance.

Based on the challenges illustrated above, we will outline some future directions for improving the reasoning ability of MWP solvers. As a formula is not only a simple sequence of mathematical symbols but also has strong logical and se- mantic relation with its context [47], selecting the appropriate formulas is the key step to solve an MWP. Hence, training the representation of the formula is essential for the neural solver. In addition, it is interesting to check the relatedness between the formula and the supporting generated rationale in each step and in different steps by jointly training with the formula and its surrounding context, which could further improve the interpretability of the solver system for users.

Combining Chain-of-thought prompting with in-context learning has shown the efficiency in unlocking the reasoning capability of LLMs without any gradient update or finetuning models [16], [18], [30], [6]. There is still some work to be done to optimize the selection of demonstration examples. Given an exemplar base with a reasoning-step solution for each question, train a small-scaled neural module to select in-context exemplars automatically for the testing problem.

Although LLMs have decent performance in many NLP tasks, building an LLM-based education system is still challenging. Li et al. [48] argue that LLMs basically need to integrate five educational abilities to address students' concerns for their studies. Besides automatically solving MWPs, the solver can be constructed as a multi-functional education system. For example, given a problem that a student cannot solve by herself the first time, the system can generate other questions with similar principles for her to make more practice. On the other hand, instead of directly giving a complete solution to the question, it is more helpful to generate some hints based on the student's partial solution. The traditional MWP solver can be transferred to an LLM-based MWP assistant to meet students' various requirements.

V. Conclusion

Automatically generating high-quality and step-by-step solutions to math word problems has numerous applications in education. This paper presents an overview of the current state of knowledge on math word problems based on LLMs. We divided the reviewed models into two groups, namely finetune-based and prompt-based, and examined their fine- tuning framework and prompting strategies. Finally, we issued the limitations of existing LLM-based MWP solvers and highlighted the future directions worth working on. We hope this survey can highlight the current state of MWP research and provide some insight into future work in this direction.

Acknowledgment

The work described in this paper was supported by the Katie Shu Sui Pui Charitable Trust — Academic Publication Fellowship (Project Reference No. KSPF/2023/05).

References

[1] E. A. Feigenbaum, J. Feldman et al., *Computers and thought.* New York McGraw-Hill, 1963, vol. 37.

[2] E. Charniak, "Computer solution of calculus word problems," in *Proceedings of the 1st international joint conference on Artificial Intelligence*, 1969, pp. 303-316.

[3] J. Li, L. Wang, J. Zhang, Y. Wang, B. T. Dai, and D. Zhang, "Modeling intra-relation in math word problems with different functional multi-head attentions," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 6162-6167.

[4] Z. Xie and S. Sun, "A goal-driven tree-structured neural model for math word problems." in *Ijcai*, 2019, pp. 5299-5305.

[5] D. Zhang, L. Wang, L. Zhang, B. T. Dai, and H. T. Shen, "The gap of semantic parsing: A survey on automatic math word problem solvers," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 9, pp. 2287-2305, 2019.

[6] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877-1901, 2020.

[7] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[8] Y. Liu, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[9] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: Decoding-enhanced bert with disentangled attention," *arXiv preprint arXiv:2006.03654*, 2020.

[10] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[11] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[12] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "Commonsenseqa: A question answering challenge targeting commonsense knowledge," *arXiv preprint arXiv:1811.00937*, 2018.

[13] Y. Wu, A. Q. Jiang, W. Li, M. Rabe, C. Staats, M. Jamnik, and Szegedy, "Autoformalization with large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 353-32 368, 2022.

[14] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, Song, and J. Steinhardt, "Measuring mathematical problem solving with the math dataset," *arXiv preprint arXiv:2103.03874*, 2021.

[15] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano *et al.*, "Training verifiers to solve math word problems," *arXiv preprint arXiv:2110.14168*, 2021.

[16] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824-24 837, 2022.

[17] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scal- ing language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1-113, 2023.

[18] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *arXiv preprint arXiv:2203.11171*, 2022.

[19] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang, "Kepler: A unified model for knowledge embedding and pre-trained language representation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 176-194, 2021.

[20] W. Liu, H. Hu, J. Zhou, Y. Ding, J. Li, J. Zeng, M. He, Q. Chen, B. Jiang, A. Zhou *et al.*, "Mathematical language models: A survey," *arXiv preprint arXiv:2312.07622*, 2023.

[21] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[22] J. He-Yueya, G. Poesia, R. E. Wang, and N. D. Goodman, "Solving math word problems by combining language models with symbolic solvers," *arXiv preprint arXiv:2304.09102*, 2023.

[23] M. Geva, A. Gupta, and J. Berant, "Injecting numerical reasoning skills into language models," *arXiv preprint arXiv:2004.04487*, 2020.

[24] J. Shen, Y. Yin, L. Li, L. Shang, X. Jiang, M. Zhang, and Q. Liu, "Generate & rank: A multi-task framework for math word problems," *arXiv preprint arXiv:2109.03034*, 2021.

[25] S. Imani, L. Du, and H. Shrivastava, "Mathprompter: Mathematical reasoning using large language models," *arXiv preprint arXiv:2303.05398*, 2023.

[26] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui, "A survey on in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.

[27] B. Wang and A. Komatsuzaki, "GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model," https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[28] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[29] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199-22 213, 2022.

[30] Z. Zhang, A. Zhang, M. Li, and A. Smola, "Automatic chain of thought prompting in large language models," *arXiv preprint arXiv:2210.03493*, 2022.

[31] L. Gao, A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, and G. Neubig, "Pal: Program-aided language models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 10 764-10 799.

[32] W. Chen, X. Ma, X. Wang, and W. W. Cohen, "Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks," *arXiv preprint arXiv:2211.12588*, 2022.

[33] D. Zhou, N. Scharli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le et al., "Least-to-most prompting enables complex reasoning in large language models," *arXiv preprint arXiv:2205.10625*, 2022.

[34] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee, and E.-P. Lim, "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models," *arXiv preprint arXiv:2305.04091*, 2023.

[35] Y. Fu, H. Peng, A. Sabharwal, P. Clark, and T. Khot, "Complexity- based prompting for multi-step reasoning," in *The Eleventh International Conference on Learning Representations*, 2022.

[36] E. Zelikman, Y. Wu, J. Mu, and N. Goodman, "Star: Bootstrapping reasoning with reasoning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 476-15 488, 2022.

[37] Y. Li, Z. Lin, S. Zhang, Q. Fu, B. Chen, J.-G. Lou, and W. Chen, "Making large language models better reasoners with step-aware verifier," *arXiv preprint arXiv:2206.02336*, 2022.

[38] X. Zhu, J. Wang, L. Zhang, Y. Zhang, Y. Huang, R. Gan, J. Zhang, and Y. Yang, "Solving math word problems via cooperative reasoning induced language models," *arXiv preprint arXiv:2210.16257*, 2022.

[39] L. Kocsis and C. Szepesvari, "Bandit based montecarlo planning," in European conference on machine learning. Springer, 2006, pp. 282- 293.

[40] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[41] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1-35, 2023.

[42] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young *et al.*, "Scaling language models: Methods, analysis & insights from training gopher," *arXiv preprint arXiv:2112.11446*, 2021.

[43] N. Reimers, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[44] A. Meurer, C. P. Smith, M. Paprocki, O. Certik, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh *et al.*, "Sympy: symbolic computing in python," *PeerJ Computer Science*, vol. 3, p. e103, 2017.

[45] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730-27 744, 2022.

[46] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.A. Lachaux, T. Lacroix, B. Rozie`re, N. Goyal, E. Hambro, F. Azhar et al., "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[47] S. Peng, K. Yuan, L. Gao, and Z. Tang, "Mathbert: A pre- trained model for mathematical formula understanding," *arXiv preprint arXiv:2105.00377*, 2021.

[48] Q. Li, L. Fu, W. Zhang, X. Chen, J. Yu, W. Xia, W. Zhang, R. Tang, and Y. Yu, "Adapting large language models for education: Foundational capabilities, potentials, and challenges," *arXiv preprint arXiv:2401.08664*, 2023.

Bridging the MAC Tunnel Vision: System-Level Performance Analysis of CNN Model Deployment at the Edge

Dwith Chenna

AMD http://www.amd.com
dwith.chenna@ieee.org

***Abstract*— Convolutional Neural Networks (CNNs), widely used in computer vision tasks, require substantial computation and memory resources, making it challenging for these models to run efficiently on resource-constrained devices. Network Architecture Search (NAS) methods have been developed to design compute-efficient models like MobileNet and EfficientNet. However, many of these models suffer from inefficiencies in hardware utilization due to their ineffectiveness in understanding the system-level details like software framework, memory bandwidth limitations and hardware capabilities for model deployment on devices. This excessive focus on compute efficiency, sometimes referred to as the "MAC tunnel vision" problem, leads to sub-optimal performance during model deployment. In this paper, we aim to bridge this gap by analyzing the performance across popular network architectures like ResNet, EfficientNet for different network hyper-parameters such as input size, feature dimensions, grouped convolution and network depth. This analysis provides CNN modeling engineers with the necessary tools to design models that can efficiently utilize the resources of available hardware. This approach not only incorporates hardware awareness into the model deployment but also considers different aspects of deployment, such as optimization algorithms (e.g., layer fusion, quantization), execution model, efficient kernels and hardware capabilities.**

***Index Terms*— Convolutional Neural Networks, Computer Vision, Optimization Algorithms**

I. INTRODUCTION

Convolutional Neural Networks (CNNs) have significantly impacted Embedded Vision and Edge AI by enabling AI applications on resource-constrained devices. With compute requirements of AI models growing each year, hardware accelerators have become crucial for efficiency. The ubiquitous presence of CNN models for vision applications has spurred the development of various CNN accelerator platforms [1-3]. These accelerators are designed to handle a wide range of CNN models and deliver efficient performance. However, the fast-evolving nature of CNN architectures presents a persistent challenge. This results in hardware accelerators constantly striving to keep pace with the latest models. Given the longer refresh cycles and lifecycles of hardware, this issue is increasingly prevalent across various fields. Tradeoffs related to compute and memory bandwidth may need to be reconsidered for future hardware versions. Bridging the gap between these developed models and inference platforms is necessary for overall system efficiency and enabling Edge AI applications. Achieving optimal system performance requires a deep understanding of hardware capabilities and the selection or design of architectures better suited to the hardware. CNN model deployments on hardware involves optimizing algorithms (i.e. layer fusion, pruning, quantization), execution models, efficient kernels, and leveraging hardware capabilities. Network Architecture Search (NAS) methods [4] for automatic search and design of CNN models have been used to find the optimum tradeoff between accuracy and compute efficiency. This led to development of many smaller and compute efficient model architectures like MobileNet[5] and EfficientNet[6]. A significant limitation of current NAS designs is a lack of consideration for hardware platform capabilities or features, focusing solely on compute or Multiply and Accumulate (MAC) efficiency. This results in inefficient design choices, also referred to as "MAC tunnel vision," where memory bandwidth constraints and other system and hardware limitations are overlooked.

While the goal of NAS methods is to reduce compute or the number of parameters, this does not always lead to improved inference speed, as the software framework and underlying hardware capabilities play a crucial role. For instance, MobileNetV2[7], with 307M MACs and a model size of 13MB, exhibits a 30% higher runtime compared to ResNet18[8], which has 1.8B MACs and a model size of 45MB. This comparison clearly demonstrates that the total MACs or model size does not necessarily correlate with runtime performance. Table 1 shows the performance comparison between ResNet18 and MobileNetV2 on hardware accelerator [9], highlights inefficiencies in model deployment. This discrepancy raises an important question: *how can we design models for better efficiency during deployment*? The answer involves considering various factors such as network architecture, optimizations, implementation details, and hardware capabilities. Understanding the performance characteristics of these models on the inference framework and device is crucial. Evaluating models for deployment efficiency requires a system-level approach that goes beyond just the compute or parameter count. It involves a detailed analysis of how different network design choices effects the perform under specific framework and hardware constraints. This approach ensures that the models not only leverage the full potential of the hardware but also achieve the desired efficiency in real-world applications.

TABLE I: PERFORMANCE COMPARISON RESNET18 VS MOBILENETV2

Network	Model Size (Float)	Model Size (Quantized)	# Layers	MACs	Latency (Ms)	Throughput (fps)
ResNet18	45MB	12MB	169	1.8B	7.35	538.7fps
MobileNetV2	13MB	3.5MB	357	307M	10.34	381.95fps

Many model developers often lack knowledge or understanding of the deployment framework or device, making the deployment process even more challenging. A paradigm shift in model design that considers hardware capabilities can unlock significant performance improvements [10-11]. However, understanding the implementation details and tradeoffs related to inference hardware can be daunting. To address this, a simpler interface is needed to evaluate on-device performance, enabling its integration with automation frameworks like NAS for efficient model design. While existing benchmarks provide some guidance on the performance of different model architectures, they often fail to offer useful insights due to their limited design exploration space. Fig. 1 shows the comparison of latency (runtime) vs MACs performance of popular CNN architectures like ResNet, EfficientNet on hardware accelerator, which shows a clear gap in performance during model deployment. These results are execution runtime (ms) of CNNs on hardware accelerator using software frameworks with quantization tool.

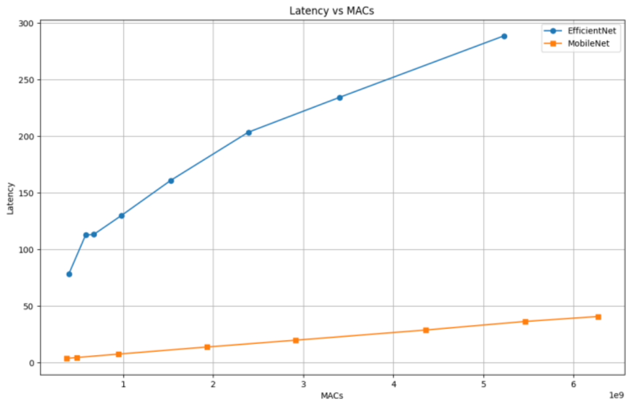

Fig. 1. Comparison of latency (ms) vs MACs performance for popular ResNet and EfficientNet architecture

In this paper, we aim to understand the effects of various macro network design parameters—such as input sizes, feature sizes, grouped convolutions and network depth—to facilitate better model design. We explore the design space, providing valuable insights for optimizing model deployment on hardware. In the following section, we will explore NAS methods and derive insights into architectural choices by examining popular network architectures. The Implementation section provides an overview of the software framework, optimizations, and hardware capabilities. Lastly, the Results and Analysis section discusses the outcomes and insights gained from performance measurements.

II. NETWORK ARCHITECTURE DESIGN

Network Architecture Search (NAS) is a powerful and increasingly essential tool in the field of machine learning and artificial intelligence, particularly for designing efficient and high-performance neural network models [4-6]. Traditional methods of manually crafting neural network architectures have become insufficient due to the rapidly growing complexity and diversity of applications in computer vision, natural language processing, and other domains. NAS automates the design process by leveraging search algorithms to explore a vast space of possible network architectures, optimizing for various performance metrics such as accuracy, model size and compute efficiency [4]. The need for NAS arises from the observation that different neural network architectures can exhibit significantly varied performance depending on the architecture, optimizations, software framework and underlying hardware. This variability poses a challenge for model developers, who must balance tradeoffs between computational cost, memory usage, and inference speed. NAS addresses this challenge by systematically evaluating a wide range of architectures, identifying optimal designs that might not be apparent through manual tuning.

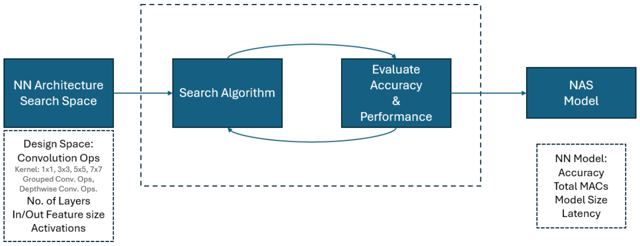

Fig. 2 Show the process of NAS design space exploration for efficient model design

Recent advancements in NAS have introduced sophisticated techniques, including reinforcement learning [12], evolutionary algorithms [13], and gradient-based methods [14], to efficiently navigate the search space. These techniques have enabled the discovery of novel architectures that outperform human-designed models on several benchmarks. Moreover, the integration of hardware- aware NAS [15] has further enhanced the applicability of these models by ensuring they are tailored to the constraints and capabilities of specific hardware platforms, such as GPUs, TPUs, and NPUs. However, in the ever-growing space of accelerator hardware there is no one solution fits all solution, which means the design choices need to be understood specific to the software framework and underlying hardware to achieve optimal performance.

In this section, we explore the different macro network design choices and try to understand its impact on the model. We will start by examining popular network architectures like ResNet and EfficientNet to understand the tradeoffs made by NAS and its effectiveness. Next, we look at macro hyper-parameters for network architecture design like input size, in/out feature size, grouped convolution, width and network depth.

A. ResNet

ResNet, created by He et al. [8], marked a significant advancement in CNN architecture by introducing residual learning and techniques for efficient deep network training. This development addressed the vanishing gradient problem, allowing the creation of even deeper CNN models. ResNet's breakthrough enabled a 152-layer deep CNN, which won the 2015 ILSVRC competition. Compared to AlexNet and VGG, ResNet achieved 20x and 8x greater depth, respectively, with relatively lower computational complexity. Empirical evidence indicated that ResNet models with 50, 101, and 152 layers outperformed their shallower counterparts. These models demonstrated notable accuracy improvements in complex visual tasks such as image recognition and localization on the COCO dataset. ResNeXt [16] further improved upon this by considering it as an ensemble of smaller networks, employing diverse convolutions (1x1, 3x3, 5x5) alongside 1x1 bottleneck convolution blocks to explore various topologies across different paths.

B. EfficientNet

EfficientNet, developed by AutoML NAS [17], is crafted to enhance both accuracy and computational efficiency. Utilizing mobile inverted residual bottleneck convolutions (MBConv) similar to MobileNet, it adopts compound scaling [6] to create various networks tailored to different computational budgets and model sizes. EfficientNet achieved superior accuracy for compute to existing CNNs, significantly reducing model size and MACs/FLOPs. For example, EfficientNet-B0 outperforms ResNet-50 while using 5x fewer parameters and 10x fewer FLOPs. These models surpass alternatives like ResNet, DenseNet, and Inception with considerably fewer parameters.

One of the significant design choices when comparing ResNet and EfficientNet is the use of Depthwise convolution, that reduce the number of compute/MACs for the same parameter size. Fig. 3 below show the basic building blocks used to create the CNN models. The 1x1 convolutions are used to expand/compress the feature maps and 3x3 convolution allowing to work on larger feature map size within the block. Using Depthwise convolution, allowed EfficientNet to be much deeper allowing them to have a much larger receptive field and better learning capability for the same compute budget.

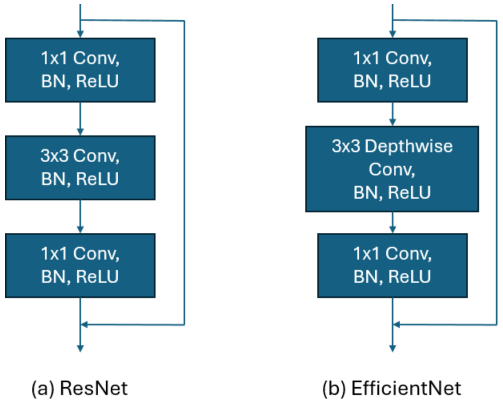

Fig. 3 Building blocks of the Convolutional Neural Network (CNN) models

The prevalence of NAS methods for model design has made these building blocks ubiquitous in many popular architectures. The design choices for such models include macro parameters that significantly influence the performance and efficiency of models. In this analysis, we start with the ResNet based NAS based on the basic building block with (1x1, 3x3, 1x1) convolutions. We slowly evolve the architecture choices to better understand its impact on the model performance. Key parameters include:

1) **Input Size**: The dimensions of the input images or data affect the network's computational load and memory bandwidth requirements. Larger input sizes can capture more detailed information but require more processing power, while smaller input sizes reduce computational demand at the cost of potentially losing fine-grained details. In this analysis we explore how the impact of input size ranging from 32x32 to 1024x1024 affects the performance on device due to its compute and memory bandwidth implications.

2) **In/Out Feature Size**: The input and output feature sizes determine the breadth of the feature maps at each layer. These sizes are crucial for balancing the network's capacity to learn complex features against the computational and memory resources required. The input/output feature sizes are intentionally multiples of 16 or 32 for better efficiency on the hardware, across wide range of values ranging from 4 to 256. The features map sizes are multiples for 2x, 4x, 8x, the basic width within the different modules.

3) **Grouped Convolution**: Grouped convolutions divide the input channels into smaller groups for separate processing. This reduces the number of parameters and computational complexity for efficient model training and inference. It is particularly useful in architectures like ResNeXt and MobileNet. For groups parameters in the range [4, 256], whenever the groups > input/output features, it is replaced by depth-wise convolution. We use groups = nan as a placeholder for depth-wise convolution i.e. it adopts the groups = input channels.

4) **Activations**: Activation functions introduce non-linearity into the network, enabling it to learn complex patterns.

Common activation functions include ReLU, Leaky ReLU, Softmax and Swish. The choice of activation function impacts the model's training dynamics and overall performance. The activations significantly impact the training and inference performance, with ReLU/ReLU6 activations popular due to its simplicity and hardware friendly implementation. In this analysis we will be limiting to ReLU activation, as it is popular and allows for layer fusion optimizations for model deployment.

5) **Network Depth**: The number of layers in a network defines its depth. Deeper networks can model more complex functions and hierarchical features but are prone to issues like vanishing gradients and increased computational demands. In the case of deployment on accelerators (NPU), the network depth also impacts the memory latency significantly as the activations need to be moved in/out of the local memory of the acceleration due to limited memory size.

Careful design and tuning of these parameters are essential to developing efficient and effective neural networks tailored to specific platform and its compute/memory constraints.

III. Implementation

Deploying CNN models effectively requires a robust software framework and various optimization techniques to ensure efficient performance on diverse hardware platforms. Popular frameworks such as TensorFlow, PyTorch, and ONNX provide the tools necessary for model training, evaluation, and deployment. To enhance model efficiency, optimizations like pruning and quantization are employed. Pruning reduces model size by removing redundant parameters, while quantization converts model weights and activations from floating-point to lower precision, thereby decreasing memory usage and computational requirements. Hardware accelerators such as GPUs, TPUs, and NPUs (specialized AI chips) are integral to speeding up CNN inference. These accelerators are designed to handle the parallel nature of CNN operations, delivering significant performance boosts and enabling real-time processing capabilities in applications such as image recognition and object detection.

Most of the current NAS algorithms focus on model accuracy and compute, without understanding the implications of these choices on inference or hardware accelerator efficiency. This leads to replacing compute intense operations (i.e. Convolutions) with relatively more memory intense operations (i.e. Depth-wise Convolutions), which fail to map efficiently to the hardware accelerators that are designed for CNN. This leads to degradation in performance or runtime inference time as shown in Fig 1. As the choices for different hardware accelerators varies, it is not possible to have a one size fits all solution. The model needs to understand the on-device capabilities and make decisions accordingly for efficiency. This work explores the CNN design space to help model design choices and guide the design choices. These methodologies can be applied to different platforms.

A. *Edge AI Inference*

The trained model is deployed on the edge device using offline tools. These deployment tools need to support different frameworks like PyTorch/Tensorflow or use a model exchange format like ONNX. For our analysis we will be using Ryzen AI software to deploy ONNX models on to the NPU (Phoenix).

The Ryzen AI software framework does the following:

i. Convert the model to format compatible with the inference platform i.e. converting from TFLite/PyTorch model to ONNX model.

ii. Optimize the model graph through layer fusion and quantization.

iii. Split the graph into sub-graphs for model execution on different hardware CPU/GPU/NPU

iv. Precompute buffer and allocation for model execution on the device

v. Mapping models Ops to hardware low-level kernels and code generation.

Layer fusion is an important graph optimization that combines adjacent operations like Conv + ReLU or Conv + BN + ReLU, to avoid additional memory latency involved to move the intermediate results. Quantization [18] is another popular technique that reduces the memory footprint and compute efficiency of CNN model by reducing the precision of weights and compute from float (FP32) to integer (INT8) operations. While it does have its own challenges in term of maintaining model accuracy, it is a popular technique due to inherent support for INT8 compute on many edge inference platforms.

B. *Measurement Tool*

The measurement tools measure the throughput and latency of the model deployment on hardware. The graph is converted to ONNX model, which is future optimized through quantization (i.e. INT8) for the target platform by using the ONNX quantization tool. The quantized graph is deployed on the device for latency and throughput measurement. For better analysis of hardware efficiency, we compute the M MACs/Sec, which is a normalized measure of hardware throughput. This provides more balanced view of hardware efficiency across different network models.

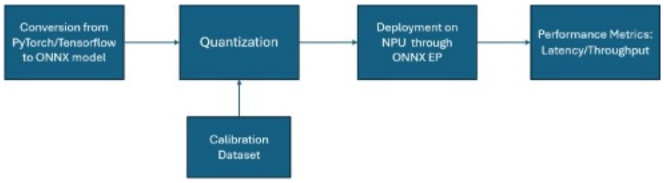
Conversion from PyTorch/Tensorflow to ONNX model
Quantization
Deployment on NPU through ONNX EP
Performance Metrics: Latency/Throughput
Calibration Dataset

Fig. 4 Measurement tools for the inference latency/throughput

IV. Results and Analysis

In this section, we summarize the results for different network parameters and characterize the performance of the inference framework and inference platform. These results can be guidelines for CNN design for runtime deployment. It also enables efficient model design of inference deployment on hardware. It gives useful insights into the effect of different building blocks, input size, grouped convolution operations, network width and network depth. While these results and analysis are focused on CNN due to its better understanding of design space, they can be applied to Transformers and similar architectures. These results characterize the performance of NPU for different network design choices, giving valuable insights to the AI/ML engineers about the inference framework and platform.

A. NPU Performance Characteristics

To better understand the performance characteristics of the CNN model deployment at the Edge, we try to highlight one dimension by sweeping it across the search space while keeping all the other dimension constant. This allows us to analyze the impact of the hyperparameter on the model deployment. During this analysis we try to answer the following: i) what is the impact on model performance, ii) what is the reason for this characteristic and iii) what are guidelines for implementation of NN design.

B. Input Size

Input image size has significant effect on the network accuracy and performance. As it decides the amount of computation needed and determines the intermediate activation size. Hence, it has a twofold impact on both compute and memory bandwidth. In Fig. 5, where we compare the different input size profiles for combinations of CNN models on CPU, we see a roof line curve maximizing at 224x224 resolution. There might be different factors contributing to this, such as the efficient or customized kernels might be designed for the more popular input resolutions, or the inputs hardware tradeoffs yield best results for that specific resolution. This gives AI/ML engineers the necessary information to make design choices for input resolution for optimal performance.

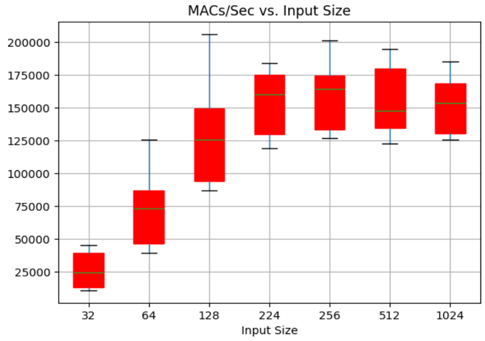

Fig. 5 Hardware throughput efficiency (M MACs/sec) vs Input Size on CPU

C. Feature Size/Width

These feature sizes determine the number of input/output feature maps at each layer a.k.a network width. The hardware accelerators are designed as SIMD machines to exploit the parallelism within the convolutions that form the bulk of network compute. These accelerators have SIMD width that are multiples of 32/64. To have the efficient implementation we see in Fig. 6 when the width is increased in multiples of 32, the overall hardware efficient is improved.

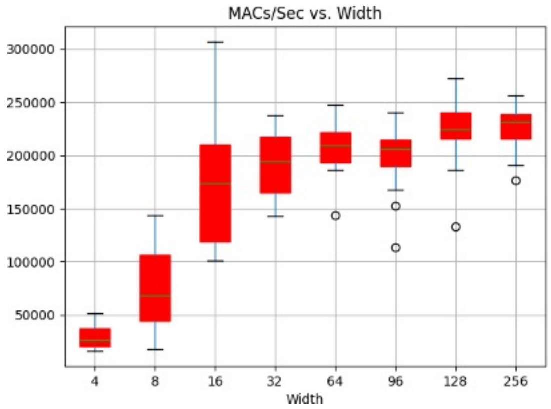

Fig. 6 Hardware throughput efficiency (M MACs/sec) vs width on CPU

Fig. 7 shows the impact of width = 32/64 on a range of CNN models. We see a significant improvement in hardware efficiency, which is even more pronounced for regular convolution compared to depth-wise convolution. In summary, instead of slowly growing the feature map size, if the network is able increase feature maps size to multiples of 32 at the earliest i.e. in the first few layers, we see a boost in the overall performance on the device.

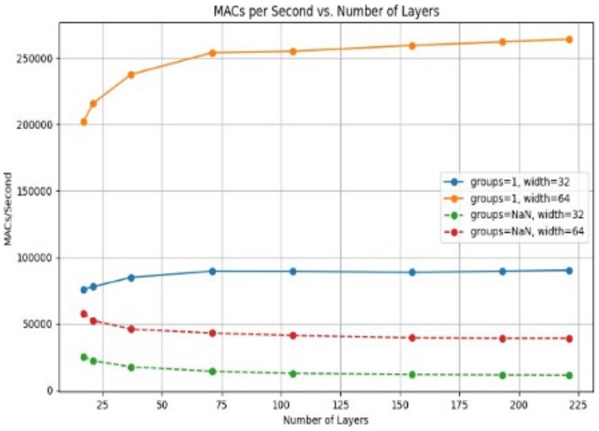

Fig. 7 Hardware throughput efficiency (M MACs/sec) vs number of layers on CPU

D. Network Depth

With the advent of efficient architectures, we see more and more depth-wise convolutions. These have only a fractional computational cost compared to regular convolutions, allowing these networks to grow much deeper for the same compute budget. However, in case of deployment on accelerators, the network depth also impacts the memory latency significantly as the activations need to be moved in/out of the local memory of the acceleration due to limited memory size. Fig. 8 shows the latency profiles of these networks across different widths. We see that convolutions (groups=1) show higher latency due to significantly larger MACs compensating for the hardware inefficiencies in the depth-wise convolutions (groups=nan). An interesting observation is how CPU is better able to handle the depth-wise convolutions across different depths compared to which shows a linear increase in latency, due to memory latency overheads for the data movement.

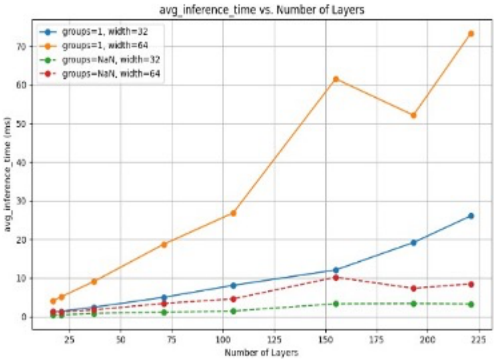

Fig. 8 Latency (ms) vs network depth on CPU

E. Grouped Convolution

Grouped convolutions is a compute efficient convolution operation used to reduce the compute intensity of the model. In Fig. 9, we can clearly see distinct characteristic profiles for grouped convolutions on CPU. In the case of CPU, we see the groups = 1,2 seems to have high hardware efficiency compared to larger groups like 16/32. As shown in the figure, as it draws closer to depth-wise convolution we again see a spike in the performance. This can be explained by the generic processor (CPU) and cached based memory system is able to efficiently handle depth-wise convolution. On the contrary we see a steady fall in efficiency, which are designed to leverage the parallelism and data reuse, which suffers from the excessive memory latency due to limited reusability of weight/inputs for depth-wise convolutions. Moreover, these bulky regular convolution with potential redundancy make them robust to quantization noise.

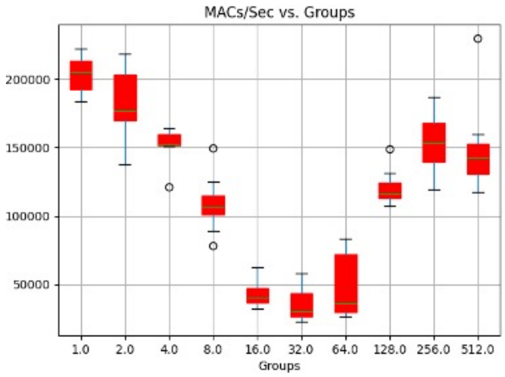

Fig. 9 Hardware throughput efficiency (M MACs/sec) vs groups on CPU

In summary, we present a comprehensive performance results in the CNN design space to learn the characteristics of inference platform. The findings provide an explanation for implication of design choices in the design space.

V. Conclusion

In conclusion, we have characterized a collection of CNN model architectures to enable efficient design choices. Our analysis provides guidelines across various parameters to enhance the efficiency of CNN models on hardware. Our findings demonstrate that hardware-aware design choices for CNNs can significantly improve overall efficiency, maximizing the capabilities of both software frameworks and hardware accelerators. This research underscores the importance of considering hardware constraints and opportunities in the design of CNN models to achieve optimal performance.

VI. Future Work

In this paper, we presented a systematic study on the impact of various macro network architecture choices in a controlled setting. While this paper primarily discusses CNN models, similar analysis can be applied to transformers or similar models for better model design. Moreover, expanding this study on larger datasets of NAS models with various network configurations will give a more comprehensive understanding of performance characteristics on a wide range of model architectures.

References

[1] Facebook, "Accelerating Facebook's Infrastructure with Application specific Hardware," https://engineering.fb.com/2019/03/14/data-center-engineering/accelerating-infrastructure/, 2021.

[2] "Edge TPU," https://cloud.google.com/edge-tpu, accessed: 2021-01-09.

[3] EETimes, "AWS Rolls Out AI Inference Chip," https://www.eetimes.com/aws-rolls-out-ai- inference-chip/, 2021.

[4] Ren, P., Xiao, Y., Chang, X., Huang, P.-y., Li, Z., Chen, X., and Wang, X. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Comput. Surv.*, 54(4), 2021.

[5] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam.

MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[6] Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105-6114, 2019.

[7] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuarells and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510-4520, 2018

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016.

[9] "RyzenAI Software": https://www.amd.com/en/developer/resources/ryzen-ai-software.html

[10] Hadjer Benmeziane et al. 2021. A comprehensive survey on hardware-aware neural architecture search. *arXiv preprint arXiv:2101.09336*, 2021.

[11] Chaojian Li, Zhongzhi Yu, Yonggan Fu, Yongan Zhang, Yang Zhao, Haoran You, Qixuan Yu, Yue Wang, Cong Hao, and Yingyan Lin. 2021. HW-NAS-Bench: Hardware-Aware Neural Architecture Search Benchmark. In *Proc. Int. Conf. Learn. Represent.* https://arxiv.org/abs/2103.10584

[12] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2016. [Online]. Available: http://arxiv.org/abs/1611.01578

[13] Z. Lu, I. Whalen, V. Boddeti, Y. Dhebar, K. Deb, E. Goodman, and W. Banzhaf, "Nsga-net: Neural architecture search using multiobjective genetic algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2019. [Online]. Available: https://doi.org/10.1145/3321707.3321729

[14] X. Zhang, Z. Huang, and N. Wang, "You only search once: Single shot neural architecture search via direct sparse optimization," CoRR, vol. abs/1811.01567, 2018. [Online]. Available: http://arxiv.org/abs/1811.01567

[15] L. Zhang, Y. Yang, Y. Jiang, W. Zhu, and Y. Liu, "Fast hardware-aware neural architecture search," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 2959-2967.

[16] Hitawala, S. Evaluating ResNeXt Model Architecture for Image Classification. *arXiv 2018, arXiv:1805.08700*

[17] X. He, K. Zhao, and X. Chu, "AutoML: A survey of the state-of-theart," *Knowl.-Based Syst.*, vol. 212, Jan. 2021, Art. no. 106622.

[18] Dwith Chenna, "Quantization of Convolutional Neural Networks: A Practical Approach", *International Journal of Science & Engineering Development Research*, Vol.8, Issue 12, page no.181 - 192, December-2023, Available: http://www.ijrti.org/papers/IJRTI2312025.pdf

Large Language Models (LLMs): Quantization

Dwith Chenna[1] and Rahul Kavi

[1]AMD http://www.amd.com
dwith.chenna@ieee.org

I. INTRODUCTION

Large language models (LLMs) have demonstrated exceptional performance across a range of tasks, but their computational and memory demands are significant. These models, such as GPT-3 [1], which boasts 175 billion parameters, require substantial resources to operate—needing at least 350GB of memory to store and run in FP16. This setup demands multiple high-capacity GPUs, like 8×48GB A6000 or 5×80GB A100, merely for inference. Figure 1, show the comparison of LLM model size and GPU memory. The substantial computation and communication overhead often result in impractical latency for real-world applications.

One effective strategy to mitigate these challenges is quantization, which enhances the efficiency of LLMs by reducing their memory footprint and computational requirements. Quantization achieves this by representing weights and activations with low-bit integers, such as INT8 or INT4, thereby lowering GPU memory consumption and improving throughput, especially in operations like General Matrix Multiply (GEMM) in linear layers and Batch Matrix Multiply (BMM) in attention mechanisms. This approach can significantly decrease the cost of deploying LLMs, making them more feasible for practical applications [3-4]. Activations of LLMs are challenging to quantize due to observed large magnitude outliers, which leads to quantization error and degradation in accuracy [5]. This makes it difficult to have a quantization method that can work across models without significant degradation in model accuracy and maintaining performance.

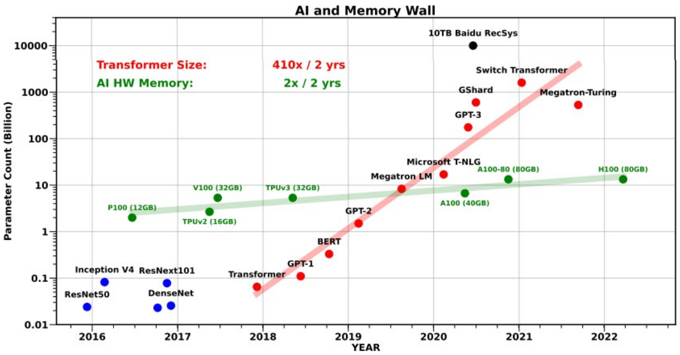

Fig. 1 AI model size over the years along with AI accelerator memory capacity [11]

In this article, we will review some of the popular quantization methods and its impact on accuracy and performance of popular models like OPT, Llama-2/3, understanding how hardware friendly and post training quantization methods for LLMs that can leverage support of low precision INT8/INT4 compute efficiency available in the hardware accelerators. More recently, we have seen the introduction of smaller Small Language Models (SLMs). Unlike LLMs that need hundreds of billions of parameters, these SLMs require a few billion parameters. However, these SLMs are trained on much cleaner data (textbook quality data), and are designed to be more efficient.

II. QUANTIZATION

In this section, we present the mathematical framework for the quantization scheme, which facilitates the efficient execution of integer arithmetic operations on quantized values. The transformation from real numbers r to quantized integers q is defined by equation for Asymmetric quantization, where S and Z, represent the scale and zero-point quantization parameters, respectively. For 8-bit quantization, q is an 8-bit integer, the scale is typically a floating-point value that is represented using a fixed-point format, and the zero-point is of the same type as the quantized value. A key constraint is placed on the zero-point to ensure that real zero values are quantized accurately, without error. The reverse mapping, from quantized values back to real values, is described by equation (2).

Quantization:

$$q = round\ (r/S + Z)$$

De-Quantization:

$$r = S\ (q - Z)$$

Symmetric Quantization:

$$q = round(r/S)$$

Where "S" is the scale and "Z" is the zero points, which are determined from the original float distributions using:

$$S = (r\_max - r\_min) / (q\_max - q\_min)$$
$$\mathrm{Z = round\ (q\_max - r\_max / S)}$$

III. Transformers Architecture

Transformers have emerged as a transformative technology in the realm of natural language processing (NLP), fundamentally altering how machines understand and generate human language. Unlike traditional models that process text sequentially, word by word, Transformers can analyze an entire sentence at once. This parallel processing capability allows them to grasp the nuances of language more effectively, leading to significant improvements in both speed and accuracy. The concept of the Transformer was first introduced in the seminal paper Attention Is All You Need [6]. Originally designed to tackle sequence-to-sequence tasks such as machine translation and text-to-speech, Transformers have since become the cornerstone of many advanced NLP applications.

A. Self-Attention

A key innovation of the Transformer architecture is the self-attention mechanism. This enables the model to assess the significance of each word in a sentence relative to all others, facilitating a deeper contextual understanding. By handling entire sentences simultaneously, Transformers not only expedite processing but also maintain a coherent understanding of context across long distances in text. This capability has revolutionized tasks like machine translation, content generation, and even the creation of human-like text, setting new benchmarks in NLP.

B. Encoder-Decoder Architecture

Transformers are built upon a two-part architecture: the encoder and the decoder. The encoder is responsible for reading and processing the input text, effectively distilling it into a form that the model can comprehend. This process involves breaking down a sentence into its core elements. The decoder, on the other hand, takes this processed information and generates the output sequence, such as translating the sentence into another language. This encoder-decoder interaction is crucial for tasks that require a nuanced understanding of context, like translation.

Within the encoder, multiple layers are employed, each consisting of self-attention mechanisms and feed-forward neural networks. The self-attention mechanism enables the encoder to weigh the importance of other words in the sentence when considering a specific word. This is mathematically facilitated by generating Query (Q), Key (K), and Value (V) vectors, which together allow the model to dynamically interpret the sentence's context. The decoder, starting

C. Positional Encoding

Since Transformers analyze all words in a sentence simultaneously, they require a mechanism to capture the order of words—this is achieved through positional encoding. Each word is assigned a unique positional code that represents its location in the sentence, ensuring the model understands the sequence and flow of language. This is essential for preserving the meaning and structure of sentences.

D. Multi-head Attention

A distinguishing feature of Transformers is the multi-head attention mechanism, which allows the model to focus on different parts of a sentence simultaneously. By applying multiple attention heads, the model can capture various

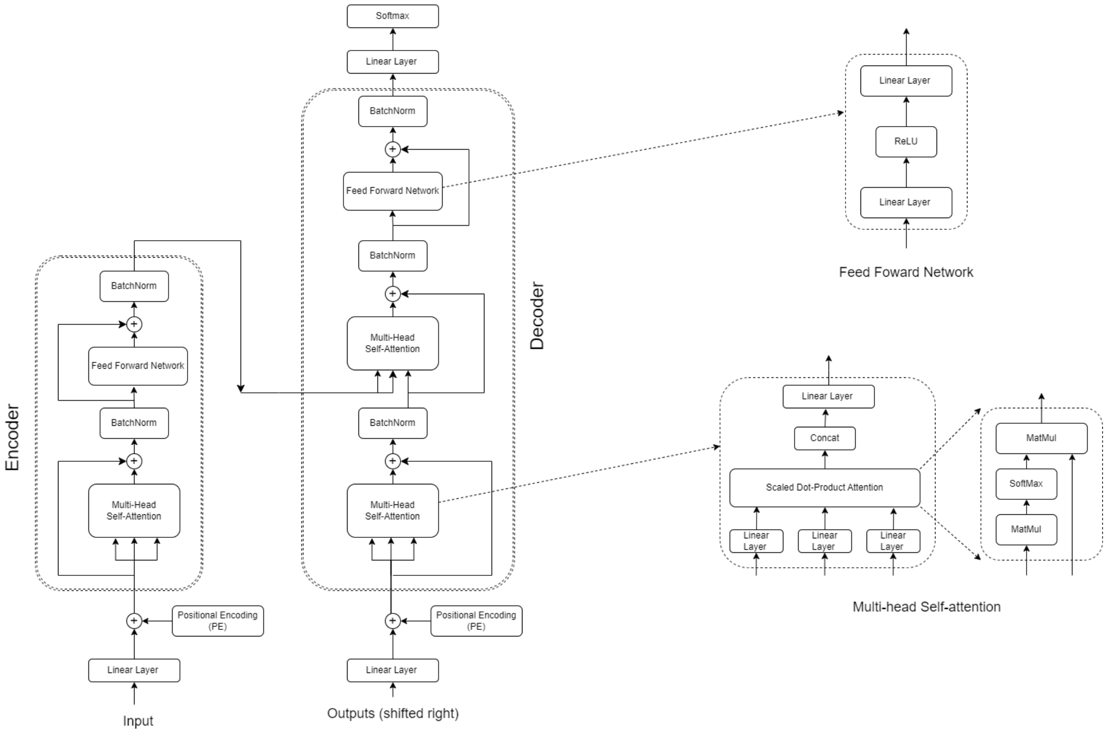

Fig. 2 Transformer architecture (left), Feed Forward Network (right) and Multi-head attention (right)

relationships and dependencies between words, leading to a richer and more nuanced understanding of the text. This parallel processing of attention layers is what gives Transformers their powerful capability to handle complex language tasks with ease.

IV. Transformer Quantization

The quantization methods used of transformer models can be broadly classified as i) Post Training Quantization (PTQ) and ii) Quantization Aware Training (QAT). The quantization parameters need to be adjusted to maintain the accuracy performance after quantization. The process of retraining the model to account for quantization is called Quantization Aware Training (QAT) or without retraining through Post Training Quantization (PTQ). A high-level comparison of two approaches is shown in Figure 3.

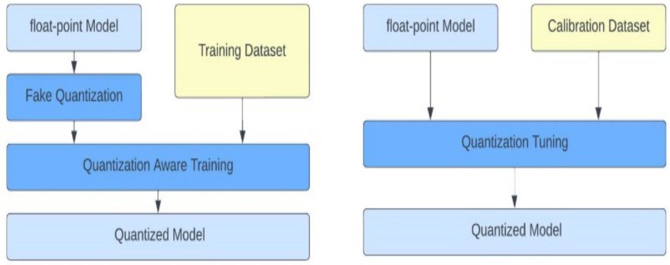

Fig. 3 Model quantization Quantization Aware Training (QAT) and Post Training Quantization (PTQ) [12]

In this article, we will be primarily focusing on PTQ which can be applied to the model trained with QAT. PTQ is the most popular technique because of its low compute requirement and its ability to quantize already trained models without the need for additional finetuning.

A. Mixed Precision

In practice, different levels of precision are applied selectively throughout the Transformer architecture. High-precision operations, such as FP16, are reserved for critical components where accuracy is paramount, and which are not compute intense like SoftMax activation and elementwise operations. These operations are sensitive to small numerical changes, and maintaining high precision ensures that the model can capture the intricate relationships within the data. On the other hand, lower-precision formats, like INT8 or even INT4, are used to optimize performance by reducing the computational load and memory usage. This is particularly advantageous for real-time applications or when operating in resource-constrained environments, though it may come with a slight compromise in accuracy. The decision on where to apply different levels of precision is driven by the specific needs of the application. In scenarios where speed and efficiency are more important, lower precision can be utilized to achieve faster inference times and reduce power consumption. By carefully managing these trade-offs, developers can tailor Transformer models to deliver the optimal balance between accuracy and performance for their intended use cases.

B. Quantization Granularity

Quantization granularity refers to the level at which quantization is applied within a Transformer model, impacting both the precision and the efficiency of computations. There are several approaches to quantization granularity, each suited to different aspects of the model's architecture.

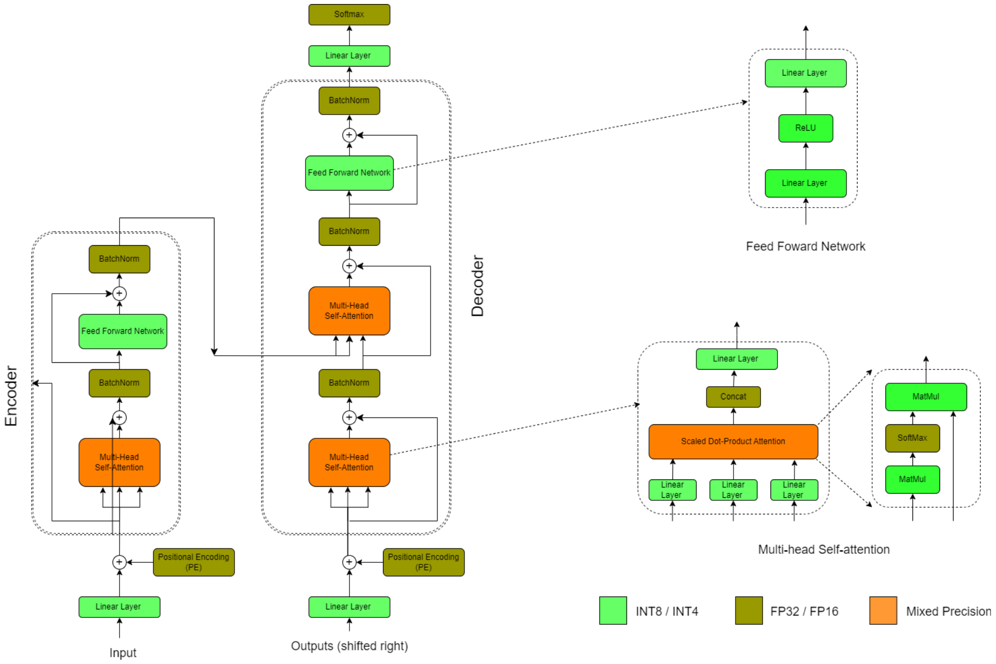

Fig. 4 Quantization precision mapping for Transformers [6]

Per-Tensor Quantization is the most straightforward approach, where a single quantization coefficient is applied across an entire tensor. While this method is computationally efficient, it may lead to a loss in accuracy, especially in complex models like Transformers where the dynamic range of values can vary significantly across different parts of the tensor.

Per-Group Quantization offers a finer level of control by applying quantization across groups of rows or columns within a tensor. For example, M rows (for activations) or K columns (for weights) might correspond to a single quantization coefficient, with K often set to values like 64. This method balances the trade-off between computational efficiency and maintaining accuracy by allowing different parts of the tensor to be quantized differently, depending on their significance.

Per-Channel or Per-Token Quantization provides the highest granularity, applying a separate quantization coefficient to each individual channel or token. In the case of per-token quantization for activations (denoted as X), each row of the activation matrix receives its own quantization coefficient. Similarly, for per-channel quantization of weights (denoted as W), each column is assigned a distinct quantization coefficient. This approach allows for the most precise adjustments, preserving the nuances in data processing but at the cost of increased computational complexity.

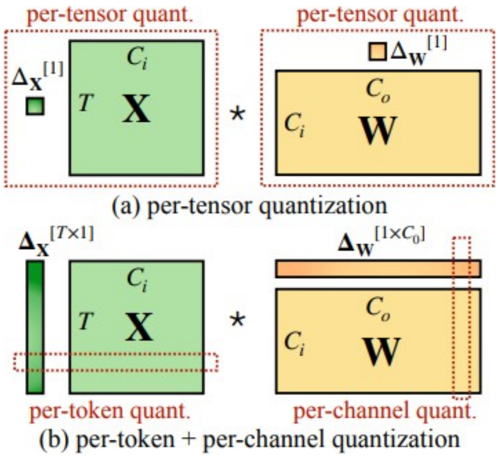

Fig. 5 Quantization granularity for Transformers [1]

By choosing the appropriate quantization granularity, developers can finetune the balance between the model's accuracy and the computational resources required for its deployment. Per-channel and per-token quantization methods are useful when maintaining high accuracy is crucial, while per-group and per-tensor quantization are favored for optimizing performance in resource-constrained environments.

Many quantization algorithms try to use different trade-off quantization schemes, mixed precision and quantization granularity to achieve the highest accuracy for the best performance efficiency. We discuss a few popular quantization techniques applied to transformer based LLMs mainly i) GPTQ ii) AWQ iii) SmoothQuant and iv) Block Quantization. We deep dive into different trade-offs for these different methods and analyze the effect on accuracy for popular models.

C. GPTQ

GPTQ (Gradient Post-Training Quantization) is an efficient algorithm for layerwise quantization of large language models (LLMs), designed to reduce the computational footprint while maintaining model accuracy. The algorithm converts floating-point weight parameters into quantized integers, aiming to minimize errors at the output level. The process begins by performing a Cholesky decomposition of the Hessian inverse matrix, which helps guide the quantization by understanding how weight changes affect the model's output. GPTQ operates in batches to run efficiently on GPUs, where each batch contains a subset of weight matrix columns.

For each column, the algorithm performs the following steps:

i. Quantizes the weights by converting floating-point values into lower-precision integers.

ii. Calculates the quantization error, which represents the difference between the original and quantized values.

iii. Updates the weights within the current block to account for the error, ensuring better approximation.

After processing a batch, GPTQ further updates all the remaining weights, compensating for any errors introduced in the quantized block. This iterative process ensures that the entire weight matrix is adjusted, allowing the model to retain high accuracy despite reduced precision. By processing weights in isolation and updating errors dynamically, GPTQ enables significant model compression with minimal accuracy loss.

D. Activation-aware Weight Quantization (AWQ)

AWQ (Activation-Weighted Quantization) [7] is an advanced quantization technique that leverages activation information to enhance the precision of weight quantization in Transformer models. Unlike traditional methods that primarily rely on the magnitude of weights to guide quantization, AWQ focuses on the sensitivity of weights based on activation patterns. Even minor contributions ranging from 0.1% to 1% can significantly improve the overall quantization results. One of the core principles of AWQ is the use of activation-based scaling, which has proven to be more effective than scaling based solely on weight magnitude. This approach ensures that the scaling factors are aligned with the actual importance of weights in the context of activations, leading to a more accurate quantization. To determine the optimal scaling factors, a small calibration dataset is used, allowing the model to adjust the scales in a way that minimizes quantization error.

Additionally, AWQ adopts a hardware-friendly approach by utilizing integer scales rather than floating-point ones. This minimizes quantization error and is more compatible with the low-precision arithmetic used in modern hardware accelerators.

Empirical heuristics suggest that scaling values less than or equal to 2 yield the best results, producing the least quantization error. AWQ typically employs a 4-bit quantization for weights and a 16-bit quantization for activations (W4A16). This configuration balances between reducing the model size and computational load while maintaining a high level of accuracy, particularly in scenarios where activation sensitivity plays a crucial role. By combining these techniques, AWQ provides a robust solution for deploying high-performance Transformer models in environments where resources are limited.

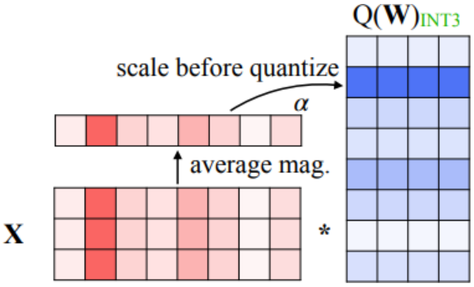

Fig. 6 Overview of AWQ Quantization, scaling weights before quantization [7]

E. Smooth Quant

SmoothQuant [8] is a technique designed to address the challenges of quantizing Transformer models, particularly focusing on the difficulty of quantizing activations due to their wide dynamic range. This wide range often results in a significant limitation on quantization precision, making it challenging to maintain model accuracy. In contrast, weights typically exhibit a more uniform distribution, making them easier to quantize with higher precision. To mitigate the impact of quantization on activations, SmoothQuant proposes a novel scaling method that redistributes quantization error from activations to weights. This redistribution is controlled by a parameter known as “migration strength,” which determines the extent to which quantization error is shifted. A migration strength of 0 indicates that all quantization error remains in the activations, reflecting the original distribution. Conversely, a migration strength of 1 moves all the quantization error to the weights. Through experimentation, it has been observed that a migration strength within the range of 0.4 to 0.5 offers an optimal balance, minimizing the overall quantization error.

SmoothQuant utilizes an 8-bit quantization scheme for both weights and activations (W8A8), a configuration that helps achieve a balance between model efficiency and accuracy. The flexibility of this method is further enhanced by its orthogonality to other quantization schemes, meaning it can be integrated with various existing quantization approaches without being constrained by them. This versatility makes SmoothQuant a powerful tool for improving the performance of Transformer models in resource-constrained environments while maintaining high accuracy.

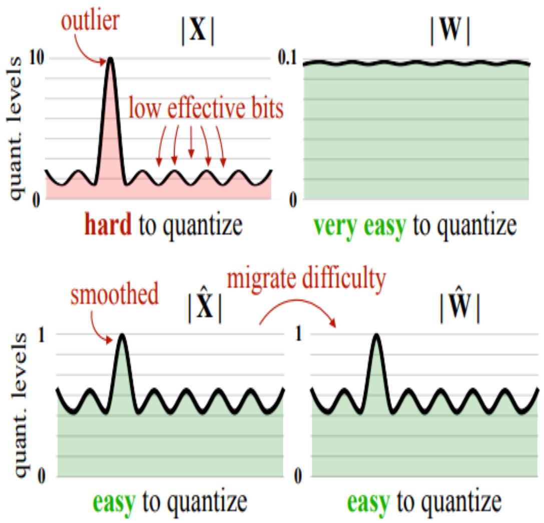

Fig. 7 SmoothQuant showing the migration factors that transfers variability in activations to weights

V. Results and Analysis

In this section, we will be evaluating the results for different quantization techniques on popular models like OPT and Llama, across their different variants in sizes.

A. Model Evaluation

Perplexity is a metric used to evaluate the performance of language models, measuring how well a model predicts a sample of text. Using the WikiText-2 dataset containing a diverse collection of Wikipedia articles, perplexity is calculated by determining the inverse probability of the test words normalized by the number of words. A lower perplexity indicates that the model has a better understanding of the language and predicts the next word more accurately. Perplexity helps in comparing different models and assessing their ability to generate coherent and contextually relevant text.

B. GPTQ

GPTQ can accurately compress some of the largest publicly available models down to 3 and 4 bits. Tables show the perplexity measurement on the wikitest-2 database on different sizes of the OPT models, showing consistent results even for large model sizes. Similarly, Table 2 shows the perplexity metrics for the Llama-2 family of models. These results are generated using the AutoGPTQ library [10].

TABLE I: Perplexity metrics on WikiText-2 for AutoGPTQ on OPT models

PPL	OPT-1.3B	OPT-2.7B	OPT-6.7B	OPT-13B
FP16	14.62	12.47	10.86	10.13
GPTQ (INT4-g128)	16.15	12.84	11.05	10.21

TABLE II: Perplexity metrics on WikiText-2 for AutoGPTQ on Llama-2 models

PPL	Llama-2 7B	Llama-2 13B	Llama-2 70B
FP16	5.47	4.88	3.32
GPTQ (INT4-g128)	5.87	4.97	3.52

C. AWQ

Activation-aware Weight Quantization (AWQ) is an effective way for low-bit 4 bit weight quantization. Table 1. shows the perplexity measurement on the wikitest-2 dataset on different variants of OPT based models, showing consistent results across model sizes from 1.3B to 30B. Similarly, we look at the perplexity metrics for the Llama-2 family of models showing consistent results with 4-bit quantization.

TABLE III: PERPLEXITY METRICS ON WIKITEXT-2 FOR AWQ ON OPT MODELS

PPL	OPT-1.3B	OPT-2.7B	OPT-6.7B	OPT-13B
FP16	14.62	12.47	10.86	10.13
AWQ (INT3-g128)	16.32	13.58	11.39	10.56
AWQ (INT4-g128)	14.92	12.70	10.92	10.22

TABLE IV: PERPLEXITY METRICS ON WIKITEXT-2 FOR AWQ ON LLAMA-2 MODELS

PPL	Llama-2 7B	Llama-2 13B	Llama-2 70B
FP16	5.47	4.88	3.32
AWQ (INT3-g128)	6.24	5.32	3.74
AWQ (INT4-g128)	5.6	4.97	3.41

D. SmoothQuant

SmoothQuant shows reliable results with 8-bit quantization for different variants of popular model i.e. OPT / Llama-2. Table 4 compares the perplexity metric measured on WikiText-2 dataset for FP16 and SmoothQuant models, it shows consistent results with minimal drop in accuracy across different variants of the OPT/Llama-2 models.

TABLE V: COMPARISON OF PERPLEXITY METRICS FOR FP16 AND SMOOTHQUANT(A8W8) ON OPT MODELS

PPL	OPT-1.3B	OPT-2.7B	OPT-6.7B	OPT-13B
FP16	14.62	12.47	10.86	10.13
SmoothQuant (A8W8)	14.82	12.50	10.86	10.14

TABLE VI: COMPARISON OF PERPLEXITY METRICS ON WIKITEXT-2 FOR FP16 AND SMOOTHQUANT (A8W8)

PPL	Llama-2 7B	Llama-2 13B	Llama-2 70B
FP16	5.47	4.88	3.32
SmoothQuant (A8W8)	5.515	4.929	3.359

VI. CONCLUSION

In conclusion, as the demand for LLM applications grows, efficient deployment strategies become increasingly critical. Quantization stands out as a key solution, enabling significant reductions in computational and memory overhead while addressing the pressing concerns of cost, environmental impact, and data privacy at the edge. By exploring and implementing advanced quantization techniques like AWQ, SmoothQuant, and Block Quantization, we can unlock the full potential of large language models in resource-constrained environments.

Along with quantization techniques, choosing quality data based on the intended end-use can greatly improve performance. This presentation will provide valuable insights into the practical application of these techniques, highlighting their benefits and trade-offs and ultimately guiding the path toward more sustainable and efficient GenAI deployments.

REFERENCES

[1] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), Advances in Neural Information Processing Systems, volume 33, pp. 1877-1901. CurranAssociates, Inc., 2020a. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[2] Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., and Zettlemoyer, L. Opt: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

[3] Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*, 2022.

[4] Yao, Z., Aminabadi, R. Y., Zhang, M., Wu, X., Li, C., and He, Y. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers, 2022. URL https://arxiv.org/abs/2206.01861.

[5] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, S. Han, Smoothquant: Accurate and efficient post-training quantization for large language models, in: *ICML, Vol. 202 of Proceedings of Machine Learning Research, PMLR*, 2023, pp. 38087-38099. 2, 20

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[7] J. Lin, J. Tang, H. Tang, S. Yang, X. Dang, and S. Han, "Awq: Activation-aware weight quantization for llm compression and acceleration," 2023.

[8] Phi-3 Microsoft, "Technical Report: A Highly Capable Language Model Locally on Your Phone", 2024.

[9] Multi-task Language Understanding on MMLU, 2024.

[10] AutoGPTQ:https://github.com/AutoGPTQ/AutoGPTQ/blob/main/examples/quantization/bas ic_usage_wikitext2.py

[11] Gholami, A., Yao, Z., Kim, S., Hooper, C., Mahoney, M. W., and Keutzer, K. Ai and memory wall. *IEEE Micro*, pp. 1-5, 2024.

[12] Dwith Chenna, "Quantization of Convolutional Neural Networks: A Practical Approach", *International Journal of Science & Engineering Development Research*, Vol.8, Issue 12, page no.181 - 192, December-2023, Available :http://www.ijrti.org/papers/IJRTI2312025.pdf

[13] AWQ, Github: https://github.com/mit-han-lab/llm-awq

[14] SmoothQuant, Github: https://github.com/mit-han-lab/smoothquant

Selected Ph.D. Thesis Abstracts

This Ph.D thesis abstracts section presents theses defended in 2023 and 2024. These submissions cover a range of research topics and themes under intelligent informatics, such as methods and techniques for counterfactual explainability in graphs, addressing the challenges in using feature engineering to classify cancer, approaches to preserving privacy in graph machine learning, and techniques for self-consistent explanation in mental status computation.

Counterfactual explainability in graphs: foundations, generative methods, and ensembles techniques

Mario Alfonso Prado Romero
marioalfonso.prado@gssi.it
Gran Sasso Science Institute, Italy

Graph Neural Networks (GNNs) have demonstrated exceptional performance across various domains, including social network analysis, recommender systems, and biochemistry. These models excel at learning from graph-structured data, where both the individual attributes of nodes and their relationships (edges) are crucial for predictions. However, GNNs operate as black-box models, which lack interpretability, a critical limitation in fields like healthcare and finance where understanding decision-making processes is essential for building trust and ensuring fairness.

Graph Counterfactual Explanations (GCEs) offer a promising approach to explaining GNN predictions by revealing how slight changes to the graph's structure or node features could lead to different outcomes. Unlike traditional feature-based explanations, counterfactuals provide actionable insights, suggesting concrete steps users can take to achieve a desired outcome. In fields like healthcare, GCEs could explain how a patient might lower their risk of disease, while in finance, they might suggest how to improve creditworthiness for a loan.

Despite their potential, generating effective GCEs presents several challenges: i) the absence of a general definition and taxonomy of GCE approaches, ii) the lack of standardized evaluation methods - as current approaches use diverse datasets, metrics, and oracles - complicates performance comparisons, iii) ensuring that counterfactuals are both plausible (i.e., realistic within the data distribution) and actionable (i.e., providing practical steps), and iv) the intrinsic nature of graph data demands explanations that incorporate both nodes' and relationships' attributes, which are typically intertwined and domain-specific.

This thesis tackles the challenges mentioned above by advancing the SoA through several key innovations. First, we tackled i) and ii) by rationalizing the field of study in a thorough Survey, where we shaped a general definition of GCE and an exhaustive taxonomy of the existing approaches. Beyond that, we provided a qualitative comparison, complemented by an empirical comparison, of existing graph counterfactual explanation methods. Furthermore, we analyzed the most widely used datasets and evaluation metrics in the GCE's SoA, assessing their strengths and limitations. This foundational work, by presenting a unifying perspective that facilitates interpreting and comparing various GCE methods, provided the clarity much needed by the nascent and fragmented field of study of GCE. Second, we set a milestone for ii) by introducing GRETEL, a versatile and extensible framework designed for developing and fairly evaluating GCE methods. This framework offers a comprehensive set of tools, including datasets, oracles, explainers, and evaluation metrics, to standardize and streamline the evaluation process. Third, we contributed to solving iii) by proposing a novel Generative GCE method that produces plausible counterfactuals aligned with the underlying data distribution, ensuring that the explanations are realistic within the context of the problem domain. Lastly, we faced challenges iv) by exploiting ensemble methods that, through selection and aggregation strategies, combine multiple GCE techniques to provide more robust counterfactuals, thus enhancing performance in the diverse datasets from different domains.

Overall, by enhancing the transparency of model behavior, the contributions presented in this thesis will enable users to make well-informed decisions based on AI-driven predictions in critical domains, including healthcare, finance, and social network analysis.

Feature Engineering for Cancer Data Modeling

Markian Jaworsky
markian.jaworsky@gmail.com
University of Southern Queensland, Australia

A Range of risk factors increases the likelihood of developing chronic illness, awareness, and understanding of these possible causes, give patients the best chance of survival by making informed life choices. Finding patterns amongst risk factors and chronic illness can suggest similar causes and provide guidance information to improve healthy lifestyles, and where outliers appear, gives clues for possible treatments. Prior studies have typically isolated data challenges of single disease datasets, however, to establish a truly healthy lifestyle the predictive feature power of many diseases is more useful. We discuss the 4 most common data challenges in health surveys and propose a novel approach to the selection of features to optimize a multi-label classifier of diabetes and 30 types of cancer, to establish a total healthy lifestyle. A novel knowledge graph is constructed from the text of health survey questions and used to determine the weight of feature relationships based on World Health Organization

(WHO) ICD codes, to prioritize selection. The results of our study demonstrate that our knowledge of graph-based feature selection, when applied to several machine learning and deep learning multi-label classifiers, improves precision, recall, and F1 score.

Six known cancer types positively correlate to diabetes diagnosis. Diabetes is known to increase the risks of pancreatic, liver, colorectal, breast, endometrial, and bladder cancer diagnosis and outcomes. The study highlights that the characteristics of diabetes, including high blood sugar and insulin levels, and inflammation are also known risk factors for cancer cells to proliferate, grow, and metastasize. This understanding can help patients determine their cancer risk as being a combination of exposure to risk factors that specifically trigger DNA damage, and secondly create inflammation in the human body that promotes cancer cells to thrive. Alternatively, diabetes decreases the risk of prostate cancer, further understanding of this negative correlation can potentially assist with prostate cancer treatment or prevention.

There exists an abundance of research papers that aim to solve data challenge scenarios in the Feature Engineering for Cancer Data Modeling analysis of health survey data, but rarely presented is a framework designed to address the many combined challenges. An exception to this trend is the multiple contributions of other researchers, who propose a novel feature selection approach to handling high dimensional imbalanced class data and designing a classifier to address the issue of data nonlinearity. A shortcoming of this paper is the preparation of a dataset that most likely will contain samples with features consisting of missing values. The study concluded that outliers were a challenge, yet this additional scenario was to be handled.

This research proposal seeks to take a holistic approach to the classification of cancer, addressing all data challenges presented in the world's largest annual health survey, and attempting to create a multi-label classifier of up to 30 individual cancer subtypes, multiple cancer occurrences, and their association with diabetes.

PRIVACY PRESERVING GRAPH MACHINE LEARNING

Olatunji Iyiola Emmanuel
iyiola_olatunji@yahoo.com
Leibniz Universität Hannover, Germany

MOST real-world data can be represented as graphs, capturing intricate relationships and dependencies among entities. This unique characteristics of graphs makes them applicable in various domains. A special family of machine learning models called graph neural networks (GNNs) are specially designed to handle graph data. In recent years, the widespread adoption of GNNs has revolutionized various analytical tasks involving graph data, such as node classification and link prediction. However, concerns regarding the privacy vulnerabilities of these models have emerged, particularly in sensitive domains like healthcare, finance and recommender systems. This thesis explores the privacy implications of GNNs through a multi-faceted analysis encompassing several attacks, defense strategies and privacy-preserving frameworks.

The first investigation focuses on the susceptibility of GNNs to membership inference attacks. We propose several attacks and defenses to effectively mitigate these attacks while minimizing the impact on model performance. Our findings reveal that structural information rather than overfitting is the primary contributor to information leakage.

Subsequently, we propose a novel privacy-preserving framework, \pkg, leveraging knowledge distillation and two noise mechanisms, random subsampling, and noisy labeling to privately release GNN models while providing rigorous privacy guarantees. The theoretical analysis within the Rényi differential privacy framework is accompanied by empirical validation against baseline methods. We also show that our privately released GNN model is robust to membership inference attacks.

Furthermore, since model explanations have become a desirable outcome of modern machine learning models, we explore the privacy risks involved in releasing model explanations from GNNs. Specifically, we study the interplay between privacy and interpretability in GNNs through graph reconstruction attacks. We demonstrate how model explanations can facilitate the reconstruction of sensitive graph structures. Various attack strategies are evaluated based on auxiliary information available to adversaries, with a proposed defense employing randomized response mechanisms to mitigate privacy leakage.

Lastly, we develop attacks to systematically study the information leakage from latent representation in graph and tabular input data domains. We reveal the susceptibility of latent space representation learning to privacy attacks that reconstruct original input with high accuracy. Furthermore, we utilize these attacks as privacy auditors to evaluate the privacy guarantees of differentially private models on both graph and tabular data, providing valuable insights into the privacy risks associated with releasing latent space representations.

By comprehensively addressing these privacy challenges, this thesis contributes to a deeper understanding of the privacy implications of GNNs and provides practical insights into enhancing their privacy-preserving capabilities in real-world applications.

Link to the official soft-copy version:
https://www.repo.uni-hannover.de/handle/123456789/18093

Website: https://iyempissy.github.io

TOWARDS SELF-CONSISTENT EXPLANATION FOR MENTAL STATUS COMPUTATION

Xiaohua Wu
xhwu@whut.edu.cn
Wuhan University of Technology, China

MENTAL status is an important psychological characteristic, representing an individual's experience of a particular moment, situation, or condition. Computing an individual's mental status can play a significant role in the fields of education, security, human resource management, psychiatric diagnosis, and social decision-making. The current dominant approaches are regression-based and conventional machine learning-based (ML) methods, often applied to data collected from online question- answering communities. While these methods have achieved acceptable prediction performance with strong interpretability, their modeling ability is limited, making it challenging to extract key factors of mental status in the era of large-scale social media data. Previous work has demonstrated the effectiveness of deep learning in mental status computation. However, deep learning-based approaches are commonly considered a "black box" due to their lack of interpretability, which restricts their use in critical areas such as psychiatric diagnosis, security, and socially assisted decision-making.

In this thesis, we propose a self-consistent explanation for mental status computation and focus on the following three problems. The first topic explores how to quantitatively explain the factors involved in mental status computation. Current explainable methods often overlook factor interactions and struggle to quantify each factor's contribution. In response, we employ Shapley value, a post-hoc explanation method based on cooperative game theory with strong theoretical support for quantifying factor contributions. Additionally, a post-hoc explanation method combining multi-factor interaction is designed.

The second topic addresses how to achieve self-consistent explanation results across different models when model-agnostic explanation methods are used to calculate factor explanations. This self-consistency is of benefit to trustworthy decision-making. We propose a multi-model joint factor contribution computation method to obtain consistent interpretation results, which in turn guides model training, leading to significant improvements in mental status computation.

Finally, we identify potential problems with the Shapley value, particularly in the construction of samples that are unlikely to appear simultaneously in real-world applications due to random bias. To address this, we propose an optimized Shapley value that removes contained pseudo-factor samples based on co-occurrence and mutual exclusion constraints. This enhancement improves both the computational efficiency of the explanation method and the consistency of the explanation results with real-world scenarios.

RELATED CONFERENCES, CALL FOR PAPERS/PARTICIPANTS

TCII Sponsored Conferences

WI-IAT 2025
The 24th IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology

London, United Kingdom
November 15-18, 2025
TBA

The International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), for some time the International Conference on Web Intelligence (WI)1), the co-located International Conference on Web Intelligence (WI) and International Conference on Intelligent Agent Technology (IAT), and initially the Asia-Pacific Conference on Web Intelligence (WI) and the Asia-Pacific Conference on Intelligent Agent Technology (IAT), aims to achieve a multi-disciplinary balance between research advances in theories and methods usually associated with Collective Intelligence, Data Science, Human-Centric Computing, Knowledge Management, and Network Science. It is committed to addressing research that both deepen the understanding of computational, logical, cognitive, physical, and social foundations of the future Web, and enable the development and application of technologies based on Web intelligence.

WI-IAT is mainly sponsored by the Web Intelligence Consortium (WI), the Institute of Electrical and Electronics Engineers (IEEE Computer Society, Technical Community on Intelligence Informatics; since 2003), and previously by the Association for Computing Machinery (ACM Special Interest Group on Artificial Intelligence; 2013-2022)

The 2025 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'25) provides a premier international forum to bring together researchers and practitioners from diverse fields for presentation of original research results, as well as exchange and dissemination of innovative and practical development experiences on Web intelligence and intelligent agent technology research and applications. Academia, professionals and industry people can exchange their ideas, findings and strategies in deepening the understanding of all Web's entities, phenomena, and developments in utilizing the power of human brains and man-made networks to create a better world and intelligent societies. WI-IAT aims to achieve a multi-disciplinary balance between research advances in theories and methods usually associated with collective intelligence, data science, human-centric computing, knowledge management, network science, autonomous agents and multi-agent systems.

This year’s conference will take place in London, United Kingdom on November 15-18, 2025. The conference theme, host and program will be announced soon. Please check for the conference website when it goes online for further information about the program, venue and attendance.

IEEE ICDM 2025
The 25th IEEE International Conference on Data Mining

Washington DC, USA
November 12-15, 2025
https://www3.cs.stonybrook.edu/~icdm2025/

The IEEE International Conference on Data Mining (ICDM) has established itself as the world’s premier research conference in data mining. It provides an international forum for sharing original research results, as well as exchanging and disseminating innovative and practical development experiences. The conference covers all aspects of data mining, including algorithms, software, systems, and applications. ICDM draws researchers, application developers, and practitioners from a wide range of data mining related areas such as big data, deep learning, pattern recognition, statistical and machine learning, databases, data warehousing, data visualization, knowledge-based systems, high-performance computing, and large models. By promoting novel, high-quality research findings, and innovative solutions to challenging data mining problems, the conference seeks to advance the state-of-the-art in data mining.

Topics of interest at this year’s conference include, but are not limited to: foundations, algorithms, models and theory of data mining, including big data mining; deep learning and statistical methods for data mining; mining from heterogeneous data sources, including text, semi-structured, spatio-temporal, streaming, graph, web, and multimedia data; data mining systems and platforms, and their efficiency, scalability, security and privacy; data mining for modelling, visualization, personalization, and recommendation; data mining for cyber-physical systems and complex, time-evolving networks; and applications of data mining in social sciences, physical sciences, engineering, life sciences, web, marketing, finance, precision medicine, health informatics, and other domains. ICDM 2025 also encourages submissions in emerging topics of high importance such as ethical data analytics, automated data analytics, data-driven reasoning, interpretable modeling, modeling with evolving environments, multi-modal data mining, and heterogeneous data integration and mining.

Authors are invited to submit original papers, which have not been published elsewhere and which are not currently under consideration for another journal, conference or workshop. Paper submissions should be limited to a maximum of ten (10) pages, in the IEEE 2-column format (https://www.ieee.org/conferences/publishing/templates.html), including the bibliography and any appendices. Submissions longer than 10 pages will be rejected without review. All submissions will be triple-blind reviewed by the Program Committee based on technical quality, relevance to the scope of the conference, originality, significance, and clarity. Please see the conference website for more information about the triple-blind review process.

The current call for full papers has a submission deadline of June 6, 2025. All submission deadlines are end-of-day in the Anywhere on Earth (AoE) time zone. Authors will be notified on August 25, 2025. Manuscripts must be submitted electronically through the online submission system on the conference website.

ICDM is a premier forum for presenting and discussing current research in data mining. Therefore, at least one author of each accepted paper must complete the conference registration and present the paper at the conference, in order for the paper to be included in the proceedings and program. This year's conference is located in Washington DC, USA. Please check the conference website regularly for updates about registration and submission guidelines. For queries regarding ICDM 2025, please contact icdm2025chairs@gmail.com

ICHI 2025
The 13th IEEE International Conference on Healthcare Informatics

Rende, Cosenza, Italy
June 18-21, 2025
https://events.dimes.unical.it/ichi2025/

ICHI 2025 is a premier community forum concerned with the application of computer science, information science, data science, and informatics principles, as well as information technology, and communication science and technology to address problems and support research in healthcare, medicine, life science, public health, and everyday wellness. The conference highlights the most novel technical contributions to stakeholder-centered technology innovation for benefiting human health and the related social and ethical implications. ICHI 2025 will feature keynotes, a multi-track technical program including papers, posters, panels, workshops, tutorials, an industrial track, and a doctoral consortium.

IEEE-ICHI is dedicated to advancing areas such as artificial intelligence (AI), machine learning (ML), foundational principles of computer and information sciences and technology, communication technology to address challenges and problems in healthcare, and everyday wellness. Journal of Healthcare Informatics Research highlights novel, cutting-edge contributions in computing for the healthcare informatics research community. It covers three major tracks: Analytics, focusing on advancing AI/ML-based decision support, generative AI, data analytics, knowledge discovery, and predictive modeling for healthcare research; Systems, focusing on building and deploying novel healthcare informatics systems (e.g., architecture, framework, design, engineering, and application); and Human factors (Human-centered Computing), focusing on intelligent communication with healthcare stakeholders (e.g., doctors, nurses, and patients), adaptive interface design for healthcare systems, user experiences of healthcare informatics applications, and understanding and motivating health behavior. By addressing these areas, IEEE-ICHI aims to bridge the gap between healthcare and information technology, fostering interdisciplinary collaboration and development of novel methods to improve patient outcomes and public health.

When submitting papers, the authors must select a track that is most appropriate for their submission. For example, a paper on information systems for healthcare delivery can be submitted to either the Systems track, or the Human Factors track, depending on the focus of the work. Before a submission is sent to the reviewers, the program chairs will also perform an assessment to determine the best fit for the submission. The conference will accept both regular and short papers. Regular papers (8-10 pages, references not counted towards the page limit) must describe mature ideas, where a substantial amount of implementation, experimentation, or data collection and analysis has been completed.

Short papers (4-6 pages, excluding references) will describe innovative ideas, where preliminary implementation and validation work have been conducted. ICHI uses double-blind reviewing for full and short papers, submissions should therefore be anonymized and all references and links disclosing the authorship should be blinded appropriately. Please check the conference webpage for the link to the submission site. In the current call for papers, the regular submission deadline for long and short papers is January 7, 2025. The notification of acceptance to authors will be March 13, 2025, with camera-ready copies due by Mar 28th, 2025. All paper submissions must adhere to the IEEE Proceedings Format.

For publication opportunities as posters, demos, or in the industry track, for workshops, tutorials, and the doctoral consortium see the respective other calls. The campus of UniCal (University of Calabria) and the conference center "Beniamino Andreatta" inside the UniCal campus will host the IEEE ICHI 2025 conference. Most hotels are located within a walking distance (3-4 minutes) from the Rende train station. Please check the conference website regularly for registration information.

IEEE BigData 2025
The 2025 IEEE International Conference on Big Data

Macau, China
December 8-11, 2025
https://bigdataieee.org/BigData2025/index.html

In recent years, "Big Data" has become a new ubiquitous term. Big Data is transforming science, engineering, medicine, healthcare, finance, business, and ultimately our society itself. The IEEE Big Data conference series started in 2013 has established itself as the top tier research conference in Big Data. The 2025 IEEE International Conference on Big Data (IEEE BigData 2025) will continue the success of the previous IEEE Big Data conferences. It will provide a leading forum for disseminating the latest results in Big Data Research, Development, and Applications.

IEEE BigData 2025 will solicit high-quality original research papers (and significant work-in-progress papers) in any aspect of Big Data with emphasis on 5Vs (Volume, Velocity, Variety, Value and Veracity), including the Big Data challenges in scientific and engineering, social, sensor/IoT/IoE, and multimedia (audio, video, image, etc.) big data systems and applications. Topics of interest for prospective include (but are not limited to) Big Data Science and Foundations; Big Data Infrastructure; Big Data Management; Big Data Search and Mining; Big Data Learning and Analytics; Data Ecosystem; Foundation Models for Big Data, and Big Data Applications. Each topic has a number of sub-themes and sub-topics relevant to Big Data research and industry. Please check the Call for Papers section of the website for the full list.

This year's Industrial Track solicits papers describing implementations of Big Data solutions relevant to industrial settings. The focus of the industry track is on papers that address the practical, applied, or pragmatic or new research challenge issues related to the use of Big Data in industry. A Government Track welcomes papers discussing the usefulness and need for publicly-contribution big data and open data and their use. Specifically, data utilization scenarios, needs analysis, data utilization obstacle analysis and solutions, data integration processes, interfaces as data utilization solutions, visualization, use cases, evidence-based policy making, building an ecosystem for solving social issues, analyzing their cases, comparing international and regional differences, and conducting comparative surveys before and after specific events (like Covid-19). Other big data solutions related to national and local governments, and public services are also welcome. For each track, either an extended abstract (2-4 pages) or a full-length paper (up to 10 pages) should be submitted through the online submission page (Industrial & Government Track dedicated page)

The deadline for electronic submission of full papers is August 31, 2025. Notification of paper acceptance is on October 25, and camera-ready versions for accepted papers are on Nov. 15, 2025. Full papers should be formatted to IEEE Computer Society Proceedings Manuscript Formatting Guidelines (see link to "formatting instructions" on the conference website). A full paper is up to 10-page IEEE 2-column format, with reference pages counted in the 10 pages. Papers must be submitted through the online submission system on the conference website.

The conference will be held in Macau, China. Please refer to the website for announcements, details and registration for IEEE BigData 2025.

IEEE ICKG 2025
The 16th IEEE International Conference on Knowledge Graphs (ICKG)

Limassol, Cyprus
December 12-13, 2025
https://cyprusconferences.org/ickg2025/

The annual IEEE International Conference on Knowledge Graph (ICKG) provides a premier international forum for presentation of original research results in knowledge discovery and graph learning, discussion of opportunities and challenges, as well as exchange and dissemination of innovative, practical development experiences. The conference covers all aspects of knowledge discovery from data, with a strong focus on graph learning and knowledge graph, including algorithms, software, platforms. ICKG 2025 intends to draw researchers and application developers from a wide range of areas such as knowledge engineering, representation learning, big data analytics, statistics, machine learning, pattern recognition, data mining, knowledge visualization, high performance computing, and World Wide Web etc. By promoting novel, high quality research findings, and innovative solutions to address challenges in handling all aspects of learning from data with dependency relationship.

All accepted papers will be published in the conference proceedings by the IEEE Computer Society. Awards, including Best Paper, Best Paper Runner up, Best Student Paper, Best Student Paper Runner up, will be conferred at the conference, with a check and a certificate for each award. The conference also features a survey track to accept survey papers reviewing recent studies in all aspects of knowledge discovery and graph learning. At least five high quality papers will be invited for a special issue of the Knowledge and Information Systems Journal, in an expanded and revised form. In addition, at least eight quality papers will be invited for a special issue of Data Intelligence Journal in an expanded and revised form with at least 30% difference.

ICKG 2025 is looking for papers around topics that include (but are not limited to): foundations, algorithms, models, and theory of knowledge discovery and graph learning; knowledge engineering with big data; machine learning, data mining, and statistical methods for data science and engineering; acquisition, representation and evolution of fragmented knowledge; fragmented knowledge modeling and online learning; knowledge graphs and knowledge maps; graph learning security, privacy, fairness, and trust; interpretation, rule, and relationship discovery in graph learning; geospatial and temporal knowledge discovery and graph learning; ontologies and reasoning; topology and fusion on fragmented knowledge; visualization, personalization, and recommendation of Knowledge Graph navigation and interaction; Knowledge Graph systems and platforms, and their efficiency, scalability, and privacy; applications and services of knowledge discovery and graph learning in all domains including web, medicine, education, healthcare, and business; big knowledge systems and applications; crowdsourcing, deep learning and edge computing for graph mining; large language models and applications; open source platforms and systems supporting knowledge and graph learning; datasets and benchmarks for graphs; neurosymbolic & hybrid AI systems; and Graph Retrieval Augmented Generation.

The conference will also include a survey track for papers reviewing recent study in aspects of knowledge discovery and graph learning. In addition, there are special tracks for each of the following topics: 01 - KGC and Knowledge Graph Building; 02 - KR and KG Reasoning; 03 - KG and Large Language Model; 04 - GNN and Graph Learning; 05 - QA and Graph Database; 06 - KG and Multi-modal Learning; 07 - KG and Knowledge Fusion; and 08 - Industry and Applications.

Paper submissions should be no longer than 8 pages, in the IEEE 2-column format, including the bibliography and any possible appendices. Submissions longer than 8 pages will be rejected without review. All submissions will be reviewed by the Program Committee based on technical quality, originality, significance, and clarity. Please see the Call for Papers section on the conference website for important information about submission requirements for the survey and special tracks.

Manuscripts must be submitted electronically in the online submission system. No email submission is accepted. To help ensure correct formatting, please use the style files for U.S. Letter as template for your submission. These include LaTeX and Word. The submission deadline for abstract and full paper is July 15, 2025 (AoE). Notification for acceptance/rejection is September 15, 2025, with camera-ready deadline and copyright forms on October 15, 2025.

This year, ICKG 2025 will be held at The St. Raphael Resort in Limassol, Cyprus. Special rates have been secured at a number of different budget hotels for conference participants (please check the Accommodation page on the conference website to book via the online registration system). The early registration deadline is October 29, 2025. For information about the program and registration, please regularly check the conference website for updates.

AAMAS 2025
The 24th International Conference on Autonomous Agents and Multi-Agent Systems

Detroit, Michigan, USA
May 19-23, 2025
https://aamas2025.org/

Autonomous Agents and Multiagent Systems (AAMAS) is the largest and most influential conference in the area of agents and multiagent systems, bringing together researchers and practitioners in all areas of agent technology and providing and internationally renowned high-profile forum for publishing and finding out about the latest developments in the field. AAMAS is the flagship conference of the non-profit International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

AAMAS 2025 welcomes the submission of technical papers describing significant and original research on all aspects of the theory and practice of autonomous agents and multiagent systems. Papers will be associated with one of the following areas of interest: Learning and Adaptation (LEARN); Game Theory and Economic Paradigms (GTEP); Coordination, Organizations, Institutions, Norms, and Ethics (COINE); Search, Optimization, Planning, and Scheduling (SOPS); Representation, Perception, and Reasoning (RPR); Engineering and Analysis of Multiagent Systems (EMAS); Modeling and Simulation of Societies (SIM); Human-Agent Interaction (HAI); Robotics and Control (ROBOT); or Innovative Applications (IA).

In addition to the main track, AAMAS 2025 will feature four special tracks: the AAAI Resubmissions Track, the Blue Sky Ideas Track, the JAAMAS Track, and the Demo Track, each with a separate Call for Papers. The focus of the Blue Sky Ideas Track is on visionary ideas, long-term challenges, new research opportunities, and controversial debate. The JAAMAS Track offers authors of papers recently published in the journal Autonomous Agents and Multiagent Systems (JAAMAS) that have not previously appeared as full papers in an archival conference the opportunity to present their work at AAMAS 2025. The Demo Track, finally, allows participants from both academia and industry to showcase their latest developments in agent-based and robotic systems.

PhD students working in the broad research areas served by AAMAS are invited to take part in the Doctoral Consortium (DC) of AAMAS 2025. Each accepted student will be matched with an established researcher from the community who will act as the student's mentor. Accepted students will also have the opportunity to present their work to their peers and senior members of the community attending. The program will be completed with an informal lunch for participating students and mentors as well as a panel discussion focusing on questions of career management. The DC is specifically intended for PhD students who already have a concrete research proposal and preliminary results, but who still have sufficient time before the completion of their dissertation so as to be able to benefit from the DC experience. The deadline for abstracts to the doctoral consortium is January 17, 2025, and the submission deadline is January 22.

AAMAS 2025 will feature tutorials and workshops immediately before the main conference, with these events running on May 19-20. Tutorials are half-day and in-person, covering areas aligned with the topics of the main track. A list of accepted workshops is available on the conference website, with links to websites provided by the facilitators that contain further information about each workshop provided.

This year's venue is the Detroit Marriott at the Renaissance Center, Michigan USA. Submissions for papers were due October 16, 2024, last year and authors were notified on December 23, 2024. Camera-ready papers are due on February 21, 2025. Registration opens January 2025. For information about Visas, accommodation and the venue, please visit the conference website.

____

AAAI 2025
The 39th Annual AAAI Conference on Artificial Intelligence
Philadelphia, Pennsylvania, USA February 25-March 4, 2025
https://aaai.org/conference/aaai/aaai-25/

The purpose of the AAAI conference series is to promote research in Artificial Intelligence (AI) and foster scientific exchange between researchers, practitioners, scientists, students, and engineers across the entirety of AI and its affiliated disciplines. AAAI-25 is the Thirty-Ninth AAAI Conference on Artificial Intelligence. As with AAAI-24, the theme of this conference is to create collaborative bridges within and beyond AI. In addition to the bridge theme, we emphasize the importance of AI for social impact and responsible AI. Like the AAAI 2024 conference, AAAI-25 will feature technical paper presentations, special tracks, invited speakers, workshops, tutorials, poster sessions, senior member presentations, competitions, and exhibition programs, and two other activities: a Bridge Program and a Lab Program. Many of these activities are tailored to the theme of bridges and are selected according to the highest standards, with additional programs for students and young researchers.

AAAI-25 welcomes submissions reporting research that advances artificial intelligence, broadly conceived. The conference scope includes machine learning, natural language processing, computer vision, data mining, multiagent systems, knowledge representation, human-in-the-loop AI, search, planning, reasoning, robotics and perception, and ethics. In addition to fundamental work focused on any one of these areas, the conference expressly encourages work that cuts across technical areas of AI (e.g., machine learning and computer vision; computer vision and natural language processing; or machine learning and planning), bridges between AI and a related research area (e.g., neuroscience; cognitive science), or develops AI techniques in the context of important application domains, such as healthcare, sustainability, transportation, and commerce.

This year's conference has three technical tracks: Main Track, AI for Social Impact, and AI Alignment. As with previous years, the AI for Social Impact track emphasizes the fit between the techniques used and a problem of social importance, rather than simply rewarding technical novelty. This year, AAAI-25 will introduce a special track on AI alignment. This track is motivated by the fact that as we begin to build more and more capable AI systems, it becomes crucial to ensure that the goals and actions of such systems are aligned with human values. To accomplish this, we need to understand the risks of these systems and research methods to mitigate these risks.

There will be several pre-conference programs on before the main technical. On February 25, there will be the tutorial and lab forum, the bridge program and the AAAI/SIGAI Doctoral Consortium. These events will also occur on February 26, along with the Science Communication for AI Researchers tutorial and the Undergraduate consortium. There will be Post-Conference Ancillary Programs on March 3-4. This year, AAAI is continuing its invited speaker program, highlighting AI researchers who have just begun careers as new faculty members or the equivalent in industry. New Faculty Highlight talks will be allotted 30 minutes each; the aim for these talks is to broadly survey the candidate's research to date. Several talks will be released online and publicized each day of the AAAI conference, following which they will be available archivally as part of the conference program. Invited speakers will be further invited to contribute an article to a corresponding series in AI Magazine.

AAAI-25 registration is now open. At least one author is required to present that paper in person and therefore register for the conference at the in-person rate. All presentations must be done in-person. There will be no remote presentation option. The conference will be held at the Pennsylvania Convention Center. Please refer to the AAAI-25 conference website for the full list of events and programs, as well as important information about travel, accommodation, and registration.

SDM25
The 2025 SIAM International Conference on Data Mining

Alexandria Virginia, USA
May 1-3, 2025
https://www.siam.org/conferences-events/siam-conferences/sdm25/

The SIAM International Conference on Data Mining (SDM25) invites submissions of high-quality research papers that present original results on data mining algorithms and their applications. Data mining is a core process within computing and statistics, aimed at discovering valuable knowledge from data. This field has significant applications across various domains including science, engineering, healthcare, business, and medicine. Datasets in these fields are typically large, complex, and noisy, necessitating sophisticated, high-performance analysis techniques grounded in sound theoretical and statistical principles. SDM25 provides a venue for researchers who are addressing these problems to present their work in a peer-reviewed forum. It also provides an ideal setting for graduate students to network and get feedback for their work (as part of the doctoral forum) and everyone new to the field to learn about cutting-edge research by hearing outstanding invited speakers and attending presentations, tutorials and a number of focused workshops. The proceedings of the conference are published in archival form and are also made available on the SIAM website.

SDM24 has three main themes: Methods and Algorithms, Applications of Data Mining, and Human Factors and Social Issues. Each of these main themes has a broad number of relevant sub-topics, including but not limited to areas such as machine learning, data analytics, data mining, data science, IoT, ethics, and privacy, across different fields and applications. Please see the conference website for a list of topics.

This year's conference will include special events, such as the IBM Early Career Data Mining Research Award, which seeks to recognize one individual (no runner up/ honorable mention) who has made outstanding, influential, and lasting contributions in the field of data analysis, and the SDM Doctoral Forum. The SDM doctoral forum will be held in a plenary poster session alongside posters from the main conference, allowing for an interesting cross fertilization of ideas. SDM25 will hold minitutorials concurrently with conference minisymposia, providing focused, hands-on learning from experts in their fields. The list of minitutorials running this year are provided on the conference website. Alongside these, there are several workshops at this year's conference include AI for Time Series Analysis: Theory, Algorithms, and Applications; the 3rd Data Science for Smart Manufacturing and Healthcare Workshop, and the Second Workshop on Metacognitive Prediction of AI Behavior. The online program schedule, speaker index and list of invited presentations will be posted on the conference website soon.

The early registration deadline for SDM25 is April 3, 2025. SIAM offers conference support for all students in good standing as well as early career participants who are affiliated with U.S. institutions. The Travel Fund deadline is February 3, 2025. SDM25 will be held at the Westin Alexandria Old Town Hotel in Alexandria Virginia, USA. Please check the conference website regularly for further updates about registration, travel and accommodation.

IJCAI 2025
The 34th International Joint Conference on Artificial Intelligence

Montreal, Canada
August 16-22, 2025
https://2025.ijcai.org/

The International Joint Conferences on Artificial Intelligence, IJCAI, is the most prestigious international gathering of Artificial Intelligence researchers. Every year IJCAI is held in a different country jointly sponsored by the IJCAI organization and the national AI society of the host nation. Since its founding in 1969, IJCAI has been the premier conference for the global AI community, fostering the exchange of groundbreaking advancements and achievements in artificial intelligence research.

IJCAI 2025 invites submissions for the 34th International Joint Conference on Artificial Intelligence (IJCAI), scheduled to take place in Montreal, Canada, from August 16 to August 22, 2025. To support authors who may experience difficulties obtaining Canadian visas, a satellite event will be hosted in Guangzhou, China, from August 28 to August 31, 2025. In addition to the main track, authors will be able to submit papers to four special tracks (AI for Social Good, AI and Arts, Human-Centred AI, and AI Enabling Critical Technologies), as well as the survey track; these tracks will post their calls for papers later this year, and their deadlines, procedures and policies may differ from each other.

Submissions to IJCAI 2025 should report significant, original, and previously unpublished results on any aspect of artificial intelligence. Papers on novel AI research problems, AI techniques for novel application domains, and papers that cross discipline boundaries within AI are especially encouraged. A selection of distinguished papers submitted to IJCAI 2025 will be invited for expedited reviewing and publication in the Artificial Intelligence Journal (AIJ) or the Journal of Artificial Intelligence Research (JAIR). The abstract submission deadline is January 16, 2025, and the author information and full paper submission deadline is January 23, 2025.

IJCAI 2025 is now inviting submissions of proposals for workshops, held immediately before the main conference on August 16-18. The aim of the workshop program is to provide a structured setting for the discussion of specialized technical topics, and the format of proposed workshops should be designed to promote an active exchange of ideas between attendees. Workshops can vary in length from half a day to two days. All workshops must be held fully in-person according to IJCAI's policy. The deadline for workshop proposal submissions is February 4, 2025 with acceptance notifications on March 4, 2025.

IJCAI 2025 invites proposals for the Tutorial Track. Tutorials will be held immediately before the technical conference. Tutorial attendance is complimentary for all IJCAI 2025 conference registrants; those not registered for the main conference can access the tutorials by paying a tutorial registration fee. Tutorial topics should serve an objective around AI areas or research (see the conference website for a full list of suggested topics). The submission deadline for tutorial proposals is March 28, 2025, with acceptance notification on April 18, 2025.

This year, the IJCAI 2025 Art Gallery is calling for groundbreaking artworks that explore the paradox of artificial intelligence, robotics, and data-driven governance. IJCAI 2025 welcome submissions that critically engage these themes through AI, robotics, augmented reality, virtual reality, and beyond. All works must incorporate AI in their creation process—whether through AI generative tools, AI-assisted techniques, or by featuring AI or robotics as integral components of the artwork. The submission form deadline for artworks is February 11, 2025. Please carefully follow instructions for artwork submissions on the Art Gallery webpage on the conference website.

The conference is also inviting proposals for Competitions and Challenges of IJCAI 2025. The competitions and challenges can be on any topic of interest to the Artificial Intelligence (AI) community. The submission deadline for these is March 18, 2025.

Please see the conference website for information about all events, submission deadlines and the submission process. Review the conference website regularly for updates about the program and registration when details become available.

IEEE Computer Society
1730 Massachusetts Ave, NW
Washington, D.C. 20036-1903