AI-Powered Knowledge Graphs for Neuromorphic and Energy-Efficient Computing

Ashish Gautam
Oak Ridge National Laboratory
Oak Ridge, USA
gautama@ornl.gov

Robert Patton
Oak Ridge National Laboratory
Oak Ridge, USA
pattonrm@ornl.gov

Thomas Potok
Oak Ridge National Laboratory
Oak Ridge, USA
potokte@ornl.gov

Ramakrishnan Kannan
Oak Ridge National Laboratory
Oak Ridge, USA
kannanr@ornl.gov

James Aimone
Sandia National Laboratory
Albuquerque, USA
jbaimon@sandia.gov

William Severa
Sandia National Laboratory
Albuquerque, USA
wmsever@sandia.gov

Abstract

The surge in scientific literature obscures breakthroughs and hinders the discovery of new research paths. We propose an artificial intelligence (AI) powered framework using large language models (LLMs) and knowledge graphs (KGs) to automate parts of scientific discovery, focusing on energy-efficient AI circuits. Our hybrid approach combines LLMs, structured data, and ontology-based reasoning to construct a comprehensive knowledge graph that integrates insights across computational neuroscience, spiking neuron models, learning rules, architectural motifs, and neuromorphic device technologies. This multi-domain representation enables the generation of hypotheses that connect biological function with implementable, energy-efficient hardware architectures. Using KG embeddings and graph neural networks, the framework generates hypotheses for novel circuits, validates them through optimization on exascale HPC systems, and with tools like SuperNeuro and Fugu, the most promising designs will be prototyped in hardware. This open-source system aims to accelerate discoveries and bridging neuroscience with hardware innovation, drive collaboration, and unlock new opportunities in low-power AI computing.

CCS Concepts

• **Hardware → Methodologies for EDA**; **Emerging architectures**; • **Computing methodologies → Information extraction**; *Knowledge representation and reasoning*.

Keywords

Knowledge graphs, neuromorphic computing, energy-efficient circuits, large language models, hypothesis generation, spiking neural networks, architectural synthesis, scientific discovery automation, STDP, cortical microcircuits.

ACM Reference Format:
Ashish Gautam, Robert Patton, Thomas Potok, Ramakrishnan Kannan, James Aimone, and William Severa. 2025. AI-Powered Knowledge Graphs for Neuromorphic and Energy-Efficient Computing. In *Great Lakes Symposium on VLSI 2025 (GLSVLSI '25), June 30–July 02, 2025, New Orleans, LA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3716368.3735295

1 Introduction

The scientific method generally involves understanding prior research, forming a hypothesis based on this prior work, making a prediction, and verifying or rejecting the prediction through experiments or simulations. The volume of publications generated daily makes it impossible for the research community to keep up with the latest findings. This results in information overload, duplicate research efforts and unreliable or unreproducible results [4]. However, significant advances in artificial intelligence (AI), particularly in natural language processing and representation learning, have led to the development of increasingly powerful methods for encoding and analyzing textual information. These innovations facilitate the extraction of nuanced semantic meaning, syntactic structure, and contextual dependencies from text, enabling more accurate and scalable analysis in diverse applications in scientific research, industry, and beyond.

Consequently, automated text extraction from scientific publications using AI has made it possible to build large-scale knowledge graphs (KG) by systematically pulling key facts, relationships, and entities from thousands of research papers. This process uses natural language processing to identify and organize information such as relationships, patterns, or outcomes found in text-based data. As a result, data that was once locked in dense, unstructured text becomes accessible and connected, allowing researchers to use graph algorithms to explore patterns, generate hypotheses, and integrate findings across disciplines at a scale that manual curation could never match.

This work describes our approach to utilize cutting-edge large language models (LLMs) to extract and encode the research literature into an extensive knowledge graph to design more energy-efficient circuits for artificial intelligence (AI). We focus on integrating insights from diverse domains—including neuromorphic engineering, spiking neural networks (SNNs), and computational neuroscience—into a unified KG to guide hypothesis generation and circuit design. In particular, we aim to extract and structure knowledge related to third-generation spiking neuron models and cortical microcircuit architectures inspired by biological brains.

GLSVLSI '25, New Orleans, LA, USA

ACM ISBN 979-8-4007-1496-2/25/06
https://doi.org/10.1145/3716368.3735295

While such architectures are well studied in neuroscience, they remain underutilized in neuromorphic systems and spiking AI models. Our framework seeks to bridge this gap, enabling the automated discovery and simulation of biologically inspired circuits through scalable tools and HPC-enabled analysis.

2 Background

Traditionally, knowledge extraction from scientific publications has relied heavily on manual curation, keyword-based searches, and rule-based systems. These approaches often depend on predefined taxonomies and human effort to identify relevant information, summarize findings, and establish conceptual links between studies. Although effective in limited contexts, such methods are labor intensive, scale poorly with the rapid growth of the literature, and frequently miss deeper semantic relationships embedded in the text. Consequently, the ability to comprehensively map and integrate scientific knowledge has been constrained by both methodological limitations and the sheer volume of unstructured data.

2.1 Language Models for Scientific Knowledge Extraction

Large language models (LLMs) have significantly advanced automated scientific knowledge extraction by enabling the interpretation of complex linguistic patterns and the structuring of unstructured data. Foundational models like BERT[8], SciBERT[5], and BioBERT[19] have shown strong performance in tasks such as named entity recognition and relation extraction. More recent tools, such as SciDaSynth [26] and domain-specific models like MatSciBERT [10], demonstrate the growing role of LLMs in literature synthesis and targeted retrieval. Parallel to this, LLMs have transformed knowledge graph (KG) construction, replacing rigid, rule-based methods with prompt-based and zero-shot learning approaches. Plug-and-play systems using GPT-4 [17], frameworks like SAC-KG [6], and tools such as itext2kg [18] and KnowGPT [27] highlight how LLMs now serve as both generators and enhancers of KGs. This integration enables cross-domain reasoning and hypothesis generation at scale, while also raising challenges in maintaining consistency, scalability, and trustworthiness [21] engineering research.

2.2 Neuromorphic Design and Learning Principles

Neuromorphic and low-power circuit design techniques aim to replicate the brain's energy-efficient computation using compact, event-driven hardware. Neuromorphic systems often employ subthreshold CMOS circuits and memristive devices for spike-based communication and learning, leveraging mechanisms like spike-timing-dependent plasticity (STDP) to minimize energy use while enabling local adaptation [3] [22]. CMOS-based synapse and neuron designs further reduce area and power through voltage scaling and asynchronous operation [13]. Recent work also explores hybrid analog-memristive architectures for in-memory neuromorphic learning, enhancing energy efficiency and system integration [16][9]. However, most existing designs are handcrafted and rely on simplified models like leaky integrate-and-fire (LIF) neurons, lacking automated methods to explore biologically inspired configurations. Moreover, while novel neuromorphic devices offer distinct physical properties—such as non-volatility, local state-dependent switching, or stochasticity—their potential has not been systematically mapped to specific computational motifs or learning rules. To address this gap, we propose extracting and structuring knowledge on neuron models, microcircuit architectures, plasticity mechanisms, and device characteristics from literature spanning neuroscience, SNNs, and neuromorphic hardware. This information will be encoded in a multi-domain knowledge graph to support cross-level reasoning and generate biologically plausible, hardware-feasible architectural hypotheses.

2.3 Circuit Design Using Knowledge Graphs

Knowledge graphs have been applied in prior circuit design work to support analog design reuse, circuit validation, and component matching. In [12], a method is introduced to extract and represent meta-knowledge from analog circuit design literature. The model includes a conceptual hierarchy, performance tradeoffs, and causal design reasoning. It helps identify new applications, improve designs, and ensure correctness. A case study on 30 high-frequency circuits showcases its use. More recently, the work of [24] discusses the use of a knowledge graph for circuit design. The system models electrical circuits as RDF-based graphs, using components as nodes and wiring as edges, and simulates circuit behavior with Ngspice to calculate parameters. These parameters, combined with formulae and matching criteria also stored in RDF, help identify compatible real-world components from a product knowledge graph, enriched with data like price and specifications from vendor sites. Although full cost optimization isn't yet possible, the system enables engineers to shortlist and compare suitable components based on constraints, pricing, and availability, with future plans for automated recommendations and constraint checks. Similarly, several recent works have explored combining large language models (LLMs) with symbolic knowledge graphs (SKGs) for analog circuit synthesis. In [23], LLMs generate high-level circuit modules, while the SKG provides structured guidance on their assembly. This offers a more explainable and modular synthesis pipeline, though it remains focused on electrical component composition and does not extend to higher-level architectural reasoning. However, these works are primarily focused on analog or RTL-level circuits and operate at low abstraction levels. They are not designed to support hypothesis-driven architectural exploration.

Our work addresses a critical gap in the design flow by introducing a structured, hypothesis-driven architectural search phase that precedes circuit-level synthesis. Rather than focusing on transistor- or component-level implementation, we leverage knowledge graphs to generate and evaluate high-level hypotheses—such as functional microcircuit motifs, spiking neuron models, and biologically plausible learning rules. These hypotheses can be validated using architectural simulators like SuperNeuro and Fugu[2, 7]. Additionally, our KG captures information about the neuromorphic device landscape, allowing us to reason about which emerging substrates (e.g., memristors, spintronic devices, phase-change materials) may align with specific architectural patterns or computational properties. This integration bridges functional hypotheses with device-level feasibility, forming a cohesive framework for design exploration across abstraction layers. By utilizing KGs as active reasoning engines, our approach paves way for the automated discovery of biologically inspired architectures that are both plausible and hardware-aware.

3 Circuit Assessment Workflow

The system operates as a closed-loop architectural exploration pipeline. Beginning with structured knowledge extraction, each phase builds upon and informs the next. Hypotheses generated from the knowledge graph are evaluated via simulation, and results are fed back into earlier stages to refine future predictions. This feedback loop is central to making the system adaptive, efficient, and capable of converging toward high-performing, energy-efficient neuromorphic designs. Figure 1 illustrates a four-phase, AI-enhanced workflow for the design, evaluation, and optimization of neuromorphic circuits. It represents a complete design cycle, spanning from knowledge discovery to performance-driven refinement, and is structured to enable continuous improvement through feedback and AI-based optimization.

In Phase 1, the system collects information from diverse sources such as technical publications, circuit diagrams, and simulation code. A large language model (LLM) is employed to process and understand the unstructured content within these sources. The LLM identifies key technical concepts, component interactions, performance characteristics, and design constraints. This extracted information is then used to construct a structured knowledge graph, where nodes represent components, parameters, or concepts, and edges denote the relationships between them. The knowledge graph serves as a foundational resource for reasoning and inference in subsequent phases, enabling the system to make informed design decisions and uncover non-obvious connections across circuit designs and literature.

Phase 2 builds on the structured knowledge assembled in the previous phase. The knowledge graph acts as a source for identifying meaningful patterns and relationships that can inspire new circuit designs. By analyzing patterns and relationships within the graph, the system can identify potential combinations of components, configurations, and design strategies that align with desired circuit functions. This is especially useful for proposing novel neuromorphic circuit architectures, which mimic the behavior of biological neural networks. The system leverages prompt engineering, graph traversal techniques, and AI reasoning to hypothesize how different components might interact to produce spiking behavior, enable energy-efficient signaling, or replicate synaptic plasticity. These hypotheses form the basis for new circuit implementations, guiding the exploration of cutting-edge designs that traditional methods might overlook.

Phase 3 focuses on testing the design hypotheses generated in the previous phase. Proposed neuromorphic architectures are implemented and evaluated using neuromorphic simulation tools such as SuperNeuro and Fugu. These simulators enabling accurate assessment of circuit behavior under biologically inspired conditions. SuperNeuro supports large-scale, parallel evaluations of network dynamics, while Fugu offers a modular framework for defining and simulating spiking neuron models across different hardware backends. With these tools, the system measures performance metrics such as latency, energy efficiency, spiking activity, and synaptic behavior. The simulation results provide critical feedback on how well the circuits meet their design objectives, identifying areas for improvement and guiding the next iteration in the design loop. Models that show promise at this point could then be evaluated by low-level circuit simulators like Xyce [11] to evaluate the electrical properties and physical realism of the underlying analog or mixed-signal components. Xyce enables fine-grained simulation of transistor-level behavior, power consumption, and signal timing, making it valuable for verifying that neuromorphic circuits function correctly not only at the algorithmic level but also at the hardware implementation level. Together, these tools provide a comprehensive understanding of each circuit's behavior across abstraction levels, offering feedback that informs both performance evaluation and iterative design refinement.

Phase 4 serves as the intelligent refinement stage of the design loop, where the results of circuit simulations from Phase 3 are systematically analyzed to improve the overall workflow. Performance results are assessed to determine which hypotheses fell short of design goals and which ones demonstrated promise. This feedback is then used to guide optimization efforts across multiple levels of the pipeline.

At the knowledge extraction level (Phase 1), the system uses this performance feedback to fine-tune how information is prioritized and interpreted during extraction. For instance, if specific circuit patterns consistently lead to poor performance, the extraction algorithms can be adjusted to deprioritize or reinterpret similar structures in future iterations.

Simultaneously, the knowledge graph used in Phase 2 is updated based on these insights. Failed hypotheses may trigger the removal or weakening of certain relationships in the graph, while promising ones may reinforce or introduce new connections. This dynamic adjustment helps the system become more selective and informed in its hypothesis generation process.

The overarching goal of this optimization phase is to minimize the creation of low-value or redundant hypotheses and maximize the discovery of novel, high-performing designs. By leveraging both success and failure data, the system becomes progressively more efficient and creative, continuously refining its understanding of circuit design space and enhancing its ability to innovate.

These phases collectively create a closed-loop system in which knowledge, simulation, and AI-driven feedback operate in coordination.

4 Hypothesis Generation from the Knowledge Graph

An extensive knowledge graph forms the foundation of the workflow, enabling researchers to initiate and iteratively refine the design of neuromorphic circuits. A knowledge graph enables efficient organization and retrieval of complex interconnected information, thereby facilitating better understanding, reasoning, and insights across various domains by structuring data relationships. To the best of our knowledge, no knowledge graph currently captures relations of circuits, devices, and systems from scientific papers,

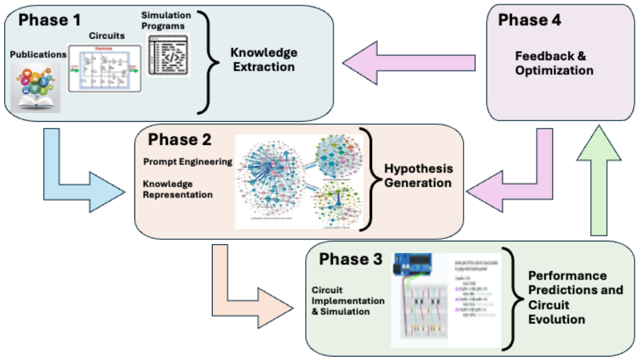

Figure 1: Circuit Assessment Workflow: Raw content in the form of publications, circuit designs, and prior simulation codes are processed into a knowledge graph. The knowledge graph is then used for hypothesis generation. Hypotheses are then evaluated through simulation to assess performance and applicability.

particularly in the neuromorphic computing field. We posit that comprehensive knowledge graphs generate hypotheses that are more insightful and contextually relevant. The significance of different parts of the KG, however, can vary across research communities and change over time. For example, the importance of dexamethasone, a corticosteroid commonly used to reduce inflammation, grew during the COVID-19 pandemic, especially among medical communities focused on treating severe acute respiratory syndrome [25]. This shift shows how the relevance of certain concepts can increase in response to new challenges and evolving priorities. Consequently, as the KG structure evolves, so too does the hypotheses that are generated. We frame hypothesis discovery as a link prediction task within the knowledge graph. Knowledge graph embedding models predict new relationships between entities that represent scientific concepts, publications, co-author relationships, circuit information, and experimental details. To perform hypothesis discovery, we leverage [15] to analyze the KG and generate fully formulated hypotheses, including newly discovered triples (hypothesis statements) by using supporting statements from publications, embeddings of these statements using LLMs, relevant reference publications, and interested authors (hypothesis evidence and history) extracted from the knowledge graph. The resulting hypotheses will then be modeled and evaluated for accuracy, energy efficiency, and computational speed.

4.1 Proof of Concept

As an initial demonstration of our circuit assessment workflow, we applied our knowledge extraction and hypothesis generation pipeline to a curated set of twenty position papers from the U.S. Department of Energy's 2024 Neuromorphic Computing for Science Workshop [1]. To guide the extraction and structuring of knowledge from the selected position papers, we designed a series of structured prompts tailored for the GPT-4o model [20]. These prompts instructed the model to identify key scientific concepts—such as neuron models, synaptic plasticity mechanisms, architectural motifs, device technologies, and implementation strategies—and organize them into labeled knowledge graphs. The model was explicitly directed to avoid generating disconnected nodes and to label relationships such as causality, similarity, contradiction, and usage. We began by building individual knowledge graphs for each paper, followed by prompts to integrate these graphs into a single, unified graph capable of supporting complex, cross-domain queries. To assess the reasoning capabilities of the resulting graph, we posed exploratory prompts such as: "Design a biologically plausible, scalable spiking neuron model for neuromorphic hardware," and "Which device technologies are best suited for implementing dynamic, task-adaptive learning rules?" These prompts enabled the model to leverage the structured knowledge for hypothesis generation and illustrate the potential of our approach in synthesizing actionable research insights. To expand this proof-of-concept into the full workflow, we plan to simulate generated hypotheses using tools like SuperNeuro and Fugu, map promising designs to appropriate hardware platforms, and feed simulation results back into the graph to refine future hypothesis generation.

Although the knowledge graph constructed from the 20 position papers enabled meaningful hypothesis generation, the results were less promising in a parallel experiment using 20 full-length research papers (the maximum number allowed in a single ChatGPT project). In that case, the responses to complex queries were often incoherent, lacked important cross-domain connections, or included incompatible elements across abstraction levels—for example, linking biologically inspired mechanisms to physically unfeasible or mismatched hardware implementations. Several factors likely contributed to this performance gap. Unlike the concise and conceptually focused position papers, research papers contain significantly more technical detail, variability in structure, and dense mathematical notation, which may reduce the model's ability to consistently extract and relate relevant concepts. Additionally, the language model used was not fine-tuned for domain-specific literature in neuromorphic computing, limiting its semantic alignment with specialized terminology and reasoning patterns. Lastly, the knowledge graph construction process may require more advanced parsing and ontology-based validation to ensure consistency and completeness at larger scales. These challenges highlight the need for domain-adapted LLMs, more structured scientific corpora, and post-processing enhancements to scale up this framework to broader, more heterogeneous literature collections.

5 Simulation, Validation, and Optimization

Following hypothesis generation, candidate neuromorphic architectures are evaluated through a multi-tiered simulation and optimization workflow. This process ensures that the proposed spiking network designs—derived from the knowledge graph—are not only functionally plausible but also meet practical constraints such as energy efficiency, latency, and scalability.

To support this, we employ simulation tools that span multiple levels of abstraction. At the high level, SuperNeuro [7] and NEST [14] are used to evaluate global network behavior, spiking dynamics, and learning patterns under biologically inspired conditions. SuperNeuro is particularly well-suited for parallel simulation of large-scale SNNs, while NEST offers mature support for neuron-level detail and biological fidelity.

Fugu [2], our open-source platform for spiking neural networks, enables the construction of modular, reusable network components referred to as Bricks. These Bricks serve as composable units that can be assembled into complex SNN architectures and compiled for various hardware backends. Previously, we used evolutionary learning methods to construct Fugu Bricks. In this work, we extend that approach by integrating Fugu with knowledge graph–derived motifs to generate and evaluate more sophisticated neuron types and microcircuit structures. A database of reusable Bricks will be created and dynamically updated as part of the design loop.

Simulation outcomes are fed back into the design pipeline as part of a closed-loop feedback system. Successful design patterns reinforce knowledge graph connections and hypothesis generation rules, while poorly performing designs are used to deprioritize ineffective motifs. This adaptive mechanism enables the system to improve its understanding of design constraints over time, enhancing its capacity to generate novel, resource-efficient neuromorphic systems.

6 Summary & Future Work

This work presents a novel framework for automating hypothesis-driven neuromorphic circuit design by integrating large language models and structured knowledge graphs. While we demonstrate only a small proof of concept using a curated set of position papers, the proposed system lays the foundation for a closed-loop, AI-enhanced design process that can scale across multiple abstraction layers (e.g., from biological insight to hardware implementation). The framework addresses a key gap in the design space by enabling the integration of neuroscientific principles, such as cortical microcircuit motifs and learning rules like STDP, with circuit-level parameters and emerging device technologies. The use of LLM-based extraction and KG embeddings allows for scalable hypothesis generation, and future iterations will incorporate continuous feedback from simulations—both architectural (e.g., SuperNeuro, Fugu) and low-level (e.g., SPICE, Xyce)—to refine and prioritize design candidates. However, challenges remain. For instance, LLM-extracted knowledge can introduce noise or bias, and integrating cross-disciplinary data from neuroscience, hardware, and machine learning poses non-trivial alignment issues. Looking ahead, we plan to expand the system to include real-world prototyping on neuromorphic hardware, enable co-design with emerging substrates, and support explainable KG reasoning paths. We also aim to develop community-driven tools for collaborative refinement and embed temporal awareness to adapt hypothesis generation to evolving research priorities. Ultimately, this framework seeks to accelerate scientific discovery in energy-efficient computing by transforming unstructured knowledge into actionable, hardware-aware design hypotheses.

Acknowledgments

This manuscript has been authored in part by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. This article has been authored by an employee of National Technology & Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employee owns all right, title and interest in and to the article and is solely responsible for its contents. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

References

[1] 2024. Neuromorphic Computing and Engineering Workshop 2024. https://web.cvent.com/event/8b49ef3c-5bec-4054-a0bf-773fc129fd96/websitePage:6f1070b9-1b00-43fe-b0f5-46836a5cb23b. Accessed: 2025-04-15.

[2] James B Aimone, William Severa, and Craig M Vineyard. 2019. Composing neural algorithms with Fugu. In *Proceedings of the International Conference on Neuromorphic Systems*. 1–8.

[3] Meysam Akbari and Kea-Tiong Tang. 2022. Low-Voltage Implementation of Neuromorphic Circuits for a Spike-Based Learning Control Module. *IEEE Access* 10 (2022), 2619–2630. doi:10.1109/ACCESS.2021.3139387

[4] Mohamed Amine Belabbes, Ian Ruthven, Yashar Moshfeghi, and Diane Rasmussen Pennington. 2023. Information overload: a concept analysis. *Journal of Documentation* 79, 1 (2023), 144–159. doi:10.1108/JD-06-2021-0118

[5] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 3615–3620. doi:10.18653/v1/D19-1371

[6] Hanzhu Chen, Xu Shen, Qitan Lv, Jie Wang, Xiaoqi Ni, and Jieping Ye. 2024. SAC-KG: Exploiting Large Language Models as Skilled Automatic Constructors for Domain Knowledge Graphs. arXiv:2410.02811 [cs.AI] https://arxiv.org/abs/2410.02811

[7] Prasanna Date, Chathika Gunaratne, Shruti R. Kulkarni, Robert Patton, Mark Coletti, and Thomas Potok. 2023. SuperNeuro: A Fast and Scalable Simulator for Neuromorphic Computing. In *Proceedings of the 2023 International Conference on Neuromorphic Systems* (Santa Fe, NM, USA) *(ICONS '23)*. Association for Computing Machinery, New York, NY, USA, Article 40, 4 pages. doi:10.1145/3589737.3606000

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. doi:10.18653/v1/N19-1423

[9] Nikhil Garg. 2024. *Neuromorphic in-memory learning with analog integrated circuits and nanoscale memristive devices*. Theses. Université de Lille ; Université de Sherbrooke (Québec, Canada). https://hal.science/tel-04821563

[10] Tanishq Gupta, Mohd Zaki, N. M. Anoop Krishnan, and Mausam. 2022. MatSciBERT: A materials domain language model for text mining and information extraction. *npj Computational Materials* 8, 1 (2022), 102. doi:10.1038/s41524-022-00784-w

[11] S Hutchinson, E Keiter, R Hoekstra, H Watts, A Waters, T Russo, R Schells, S Wix, and C Bogdan. 2002. The Xyce™ parallel electronic simulator–an overview. *Parallel Computing: Advances and Current Issues* (2002), 165–172.

[12] Fanshu Jiao, Sergio Montano, Cristian Ferent, Alex Doboli, and Simona Doboli. 2015. Analog Circuit Design Knowledge Mining: Discovering Topological Similarities and Uncovering Design Reasoning Strategies. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34, 7 (2015), 1045–1058. doi:10.1109/TCAD.2015.2418287

[13] Bomin Joo, Jin-Woo Han, and Bai-Sun Kong. 2022. Energy- and Area-Efficient CMOS Synapse and Neuron for Spiking Neural Networks With STDP Learning. *IEEE Transactions on Circuits and Systems I: Regular Papers* 69, 9 (2022), 3632–3642. doi:10.1109/TCSI.2022.3178989

[14] Jakob Jordan, Rajalekshmi Deepu, Jessica Mitchell, Jochen Martin Eppler, Sebastian Spreizer, Jan Hahne, Eric Thomson, Itaru Kitayama, Alexander Peyser, Tanguy Fardet, et al. 2019. *NEST 2.18. 0.* Technical Report. Jülich Supercomputing Center.

[15] Ramakrishnan Kannan, Piyush Sao, Hao Lu, Drahomira Herrmannova, Vijay Thakkar, Robert Patton, Richard Vuduc, and Thomas Potok. 2020. Scalable

Knowledge Graph Analytics at 136 Petaflop/s. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–13. doi:10.1109/SC41405.2020.00010

[16] Youngmin Kim, Ji Hyun Baek, In Hyuk Im, Dong Hyun Lee, Min Hyuk Park, and Ho Won Jang. 2024. Two-Terminal Neuromorphic Devices for Spiking Neural Networks: Neurons, Synapses, and Array Integration. *ACS Nano* 18, 51 (2024), 34531–34571. doi:10.1021/acsnano.4c12884

[17] Yassir Lairgi, Ludovic Moncla, Rémy Cazabet, Khalid Benabdeslem, and Pierre Cléau. 2024. Knowledge Graph Construction Using Large Language Models. In *Journée nationale sur la fouille de textes*. Lyon, France. https://hal.science/hal-04607294

[18] Yassir Lairgi, Ludovic Moncla, Rémy Cazabet, Khalid Benabdeslem, and Pierre Cléau. 2025. iText2KG: Incremental Knowledge Graphs Construction Using Large Language Models. In *Web Information Systems Engineering – WISE 2024*, Mahmoud Barhamgi, Hua Wang, and Xin Wang (Eds.). Springer Nature Singapore, Singapore, 214–229.

[19] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (09 2019), 1234–1240. doi:10.1093/bioinformatics/btz682 arXiv:https://academic.oup.com/bioinformatics/article-pdf/36/4/1234/48983216/bioinformatics_36_4_1234.pdf

[20] OpenAI. 2024. ChatGPT-4o. https://openai.com/index/chatgpt. Accessed: 2025-04-15.

[21] Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, Russa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and Damien Graux. 2023. Large Language Models and Knowledge Graphs: Opportunities and Challenges. arXiv:2308.06374 [cs.AI] https://arxiv.org/abs/2308.06374

[22] Bipin Rajendran, Abu Sebastian, Michael Schmuker, Narayan Srinivasa, and Evangelos Eleftheriou. 2019. Low-Power Neuromorphic Hardware for Signal Processing Applications: A Review of Architectural and System-Level Design Approaches. *IEEE Signal Processing Magazine* 36, 6 (2019), 97–110. doi:10.1109/MSP.2019.2933719

[23] Hashmath Shaik and Alex Doboli. 2025. Using a Symbolic Knowledge Graph to Address LLM Limitations in Analog Circuit Topology Generation. In *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*. 00528–00533. doi:10.1109/CCWC62904.2025.10903951

[24] Mizanur Rahman Syed and Johannes Bayer. 2021. Knowledge Graph based Electrical Circuit Simulation and Component Selection.

[25] Anika Tabassum, Ramakrishnan Kannan, Junqi Yin, Seung-Hwan Lim, Guojing Cong, S M Shamimul Hasan, Robert Patton, and Thomas E. Potok. 2024. Knowledge Graph Embedding using Large Language Models for COVID-19. doi:10.13139/ORNLNCCS/2229136

[26] Xingbo Wang, Samantha L. Huey, Rui Sheng, Saurabh Mehta, and Fei Wang. 2024. SciDaSynth: Interactive Structured Knowledge Extraction and Synthesis from Scientific Literature with Large Language Model. arXiv:2404.13765 [cs.HC] https://arxiv.org/abs/2404.13765

[27] Qinggang Zhang, Junnan Dong, Hao Chen, Daochen Zha, Zailiang Yu, and Xiao Huang. 2024. KnowGPT: Knowledge Graph based Prompting for Large Language Models. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 6052–6080. https://proceedings.neurips.cc/paper_files/paper/2024/file/0b8705a611ed1ce19cdb759031078705-Paper-Conference.pdf