Research Statement

Zhaocheng Zhu, Mila / University of Montreal

Reasoning, the ability to logically draw conclusions from existing knowledge, is a hallmark of human intelligence. Together with perception, they constitute the two major themes of artificial intelligence. Recently, deep learning has pushed the limit of perception beyond human-level performance in computer vision and natural language processing. Nevertheless, the progress in reasoning domains is way behind that in perception domains. As many advanced tasks (e.g. visual question answering, dialog systems) heavily rely on reasoning, it is progressively becoming a substantial hurdle on the path to artificial general intelligence. My research aims to develop models with better reasoning abilities and use them to power various applications in the broad range of artificial intelligence.

My research topic stems from the structure of reasoning problems. Unlike perception problems that have a fixed input structure (e.g. images, sequences), reasoning problems usually have flexible structures for both knowledge (e.g. knowledge graphs) and queries (e.g. multi-step queries). While the community has made much progress in scenarios where the structure is fixed, it is necessary to develop methodologies that solve general reasoning problems. However, there are two challenges: **the flexibility of knowledge structures and the large number of query structures**. For example, due to the ever growing nature of Wikipedia, any reasoning models that operate on a fixed set of knowledge will be unreliable in the future. In math problems such as arithmetic, there are an exponential number of possible query structures, and it is infeasible to train reasoning models over all structures. Both challenges have a pressing need for breakthroughs in **generalization across structures**. Additionally, the flexibility of structures makes it very challenging to scale them for industrial applications, which demands for systems that facilitate machine learning development on structured data.

Towards these goals, my research topics center around the algorithms and systems of reasoning in different modalities, such as knowledge graphs, natural languages. Since the input to a reasoning problem typically consists of some context knowledge and a query, my research can be classified as follows: **1. reasoning algorithms that generalize across different knowledge structures** [16, 9, 3, 20, 4]; **2. reasoning algorithms that generalize across different query structures** [17, 3, 12, 19] and **3. machine learning systems for developing algorithms on structured data** [15, 18]. With experts from other domains, I also collaborated in works that unify structured data and natural languages [22, 26].

Impact. By generalizing to structures of new entities, my works [16, 20] **replaced knowledge graph embedding methods** (e.g. new state-of-the-art works are mostly built on top of my NBFNet) that are limited to fixed structures and had been the *de facto* approach for 10 years. With further generalization to structures of different relation vocabularies, my work [4] can **perform reasoning inference on any knowledge graphs**, leading to the first foundation model for knowledge graph reasoning. For logical queries with different structures, my work [17] designs a model with **better inductive bias that only requires 1% training samples to achieve competitive performance as previous methods**, and remains strong performance 2 years after its publication. For multi-step reasoning problems in natural language, my work [19] **enables black-box large language models (LLMs) to learn interpretable rules that generalize to longer problems**. GraphVite [15] introduced the first method for **training node embeddings on large-scale graphs that do not fit into the memory of a single GPU**, and its idea was integrated into DGL-KE (Amazon). TorchDrug [18] **simplified the development of machine learning on structured data** (reduce the lines of code by 20×), and **engaged the machine learning community into the field of knowledge graphs and drug discovery** (43k downloads & 1.3k stars). It was **recognized by the PyTorch ecosystem, and won an NVIDIA Applied Research Accelerator grant**. TorchDrug was adopted by ChemicalX (AstraZeneca), GT4SD (IBM) and many other works.

1 Generalization across Knowledge Structures

Knowledge is the cornerstone for reasoning. Typically, high-quality knowledge is encoded as structures, such as knowledge graphs. A knowledge graph consists of sets of entities (nodes), facts (edges) and relation types. Knowledge graphs are first defined by a vocabulary of relations, then grow by adding more entities and facts. This raises an important question on reasoning models: **how can we design representation learning models that generalize to new entities?** In early embedding methods [2, 14], since every entity is tied to a learned embedding vector, they cannot deal with new entities unless they are retrained or fine-tuned on new entities. In my work NBFNet [16], instead of learning an embedding vector for each entity, we **learn the representation of every entity pair as a function of the relational paths between them**, since knowledge graph reasoning only requires representing pairs of entities. Such a design naturally releases the need for any entity embeddings, and thereby generalizes to new entities or even new knowledge graphs of the same relation vocabulary. NBFNet ranked 12 out 39 teams in 1st OGB large-scale challenge, being **the strongest single model and the most parameter efficient** (8.5M for NBFNet v.s. 11G for embedding methods) due to deprecation of entity embeddings. NBFNet has become a common baseline for knowledge graph reasoning, and its expressiveness has been discussed in several works [25, 6].

Generalization across knowledge structures also plays a key role in the era of foundation models. In order to train a single generic model that performs reasoning on arbitrary input, we need to solve the challenge of **generalization to new relation vocabularies**. In my collaboration work ULTRA [4], we revisit the success of NBFNet and solve this challenge. NBFNet generalizes to new entities because it parameterizes the relative representations of entities as a function of an invariant relation vocabulary. Hence the question is: **what is an invariant vocabulary for representing relations in various knowledge graphs?** The answer is relation-to-relation interactions, e.g. the interaction between *father* and *child* is similar to the interaction between *teacher* and *student*. Based on this insight, we construct a relation graph where nodes are relations from the original knowledge graph and edges are relation-to-relation interactions. In this way, knowledge graph reasoning is converted into two stages of representation learning: 1. computing relative relation representations as a function of relation interactions and 2. computing relative entity representations as a function of the relation representations. Both can be implemented by graph neural networks (GNNs) conditioned on the query relation and query entity respectively, akin to NBFNet. By training on 3 standard knowledge graphs, ULTRA shows **strong zero-shot generalization performance on 40 knowledge graphs of various domains and sizes, on par or surpassing state-of-the-art methods on 32 datasets**. ULTRA was the first foundation model for knowledge graph reasoning and profoundly influenced subsequent research.

2 Generalization across Query Structures

Many applications of reasoning require to deal with queries that inherently contain multiple steps, which applies to both knowledge graphs and natural languages. For example, the query "*at what universities do Turing Award winners in the field of deep learning work*" is composed of three sub queries: 1. *Who won the Turing Award?* 2. *Who works in the field of deep learning?* 3. *What universities do answers of the previous two queries work in?*. On knowledge graphs, such queries were modeled by embedding methods [5, 11] that simulate logic operations in the neural space, which does not generalize well to different combinations of steps. In my work GNN-QE, we decompose query symbolically as relation projection and logical operations, which align with the subgraph matching algorithm that generalizes perfectly when the graph is complete. To tackle the incompleteness of knowledge graphs, we parameterize relation projection with GNNs and relax logical operations with fuzzy logic. Since fuzzy logic satisfies many logic laws (e.g. De Morgan's laws), this design endows GNN-QE with better generalization across multi-step logical queries. As a result, GNN-QE achieves **a relative gain**

of 22.3% on existential positive first-order (EPFO) queries and 95.1% on negation queries under full training data, or reaches **the same performance as previous state-of-the-art method with 1% training data**. Meanwhile, it supports visualization of entity distributions for every intermediate steps. My collaboration work [3] further finds that the good inductive bias of GNN-QE leads to **generalization to graphs with new entities up to 500% larger than the training one**.

With the rise of large language models, reasoning in natural languages has gradually drawn the attention of the community. Chain-of-Thought (CoT) prompting [23] showed that we can teach LLMs to perform multi-step reasoning using a small set of in-context examples with intermediate steps. However, CoT **relies on the implicit knowledge stored in the parameters of LLMs**, of which the errors may exacerbate in multi-step reasoning. Such an issue is particularly common in problems that are not frequently covered in the pretraining data. Can we improve or fix the knowledge of LLMs, even if they are black boxes? My work Hypotheses-to-Theories (HtT) [19] gives a prompting solution that **learns explicit knowledge as textual rules**. HtT first uses an induction stage to induce rules from training samples, and then employs the learned rules as prompts in the deduction stage to solve test samples of new structures. Because the rules are expressed and learned as text, HtT opened up **a new learning paradigm that is transparent and interpretable to humans**. HtT not only improves the performance over CoT by 11-27% in accuracy, but also **discovers rules that are aligned with human knowledge**. Since the text space is shared by different LLMs and problems, the learned rules also naturally transfer to different models and to different forms of the same problem.

3 Machine Learning Systems for Structured Data

Modern machine learning heavily relies on batch processing of tensors on GPUs to accelerate the computation, which structured data does not conform to. Many previous implementations resorted for paddings to convert structured data into grid data that can be represented by tensors. However, such implementations have two issues: 1. they are very inefficient when structured data is sparse, multi-dimensional or has a broad range of sizes; 2. in order to simulate operations on structured data, one needs to implement many non-intuitive operations on the grid data. These issues set up **a high barrier between average machine learning developers and domains related to structured data**. To break the barrier, I developed TorchDrug, a machine learning system that **treats structured data as first-class citizens** and supports various applications, such as knowledge graph reasoning, molecular property prediction and many protein representation learning tasks. Technically, TorchDrug encapsulates structured data and arbitrary attributes (e.g. node features, edge types) as a set of tensors in PyTorch, with an intuitive interface of PyTorch-like operations (e.g. `concat`, `repeat`), graph operations (e.g. `subgraph`, `connected_components`) and domain-specific operations (e.g. `atom_feature`, `residue_mask`), all backed by efficient tensor implementations on GPUs. As an open-source software, TorchDrug has **brought many researchers and developers into the field of knowledge graph reasoning and drug discovery**, with 43k downloads and 1.3k stars as a metric. It is adopted by industrial open-source software such as ChemicalX from AstraZeneca and GT4SD from IBM.

4 Future Directions

In future work, I plan to continue addressing **fundamental challenges regarding generalization in reasoning**, with a focus on generalization across task structures. This will bridge many tasks that are studied independently in machine learning and lead to a single generic model for reasoning. From the perspective of reasoning, existing LLMs serve as a mixture of both factual knowledge and reasoning skills, yet we lack good control over such knowledge and skills. Therefore, we plan to investigate techniques for **updating knowledge in LLMs**, which will enable us to add up-to-date knowledge and remove unwanted knowledge. Considering the prohibitive costs of fine-tuning

LLMs, another direction is to **learn skills in the form of natural languages** without changing the base model. Additionally, I would like to collaborate with system experts to develop **model optimization, compilation and systems** that smooth the path of machine learning development.

Generalization across task structures. My work ULTRA [4] solved generalization across relation vocabularies, enabling zero-shot generalization to new knowledge graphs. If we treat queries of the same relation as a sub task of knowledge graph reasoning, this work can be also viewed as a preliminary exploration towards generalization across tasks. We plan to push this kind of generalization to new heights. For example, we expect models trained on chess to generalize to poker, or models trained on math to generalize to programming, which humans often do subconsciously. One direction is to abstract task-specific procedures as tools and leverage LLMs to learn task-independent reasoning strategies that generalize to new tasks and tools. Such models have many benefits in the era of foundation models. First, they will disentangle reasoning strategies from specific tasks, thereby being applicable to new tasks without fine-tuning. Second, they will save the cost of data collection for low-resource tasks by learning reasoning strategies from high-resource tasks.

Understanding and updating knowledge in large language models. Before the advent of LLMs, knowledge was usually stored in natural languages, or handcrafted forms such as knowledge graphs and databases. LLMs have emerged as a new source of knowledge. However, unlike knowledge bases, we do not know what facts are stored in LLMs and how to update these facts. A recent work [7] points out that LLMs cannot memorize all long-tail knowledge, even with reasonable scaling in model size. Consequently, controlling what is memorized is crucial to maximize the utility of LLMs. Currently, this can only be achieved by retraining LLMs with carefully curated data, which is extremely expensive and rigid. We would like to break this dilemma by developing lightweight tools that probe knowledge in LLMs, and precisely edit a certain piece of knowledge. Such tools will allow us to inject commonsense knowledge, up-to-date information and alignment goals into LLMs, as well as remove toxic knowledge and offload rarely used knowledge to external databases.

Learning skills in the form of natural languages. The increasing size of LLMs demands more hardware for fine-tuning, leading most developers to rely on prompt engineering to teach LLMs skills for downstream tasks. However, this approach not only has scalability issues, but also results in redundant efforts in the community. A recent work in library learning [21] suggests that LLMs can learn textual skills given an environment. By pushing such an idea to the extreme, we may let LLMs write a textbook for themselves by learning skills from various datasets. We foresee several advantages for this framework. First, it does not require gradients, thereby reducing the hardware requirement to inference level. Second, text is composable and decomposable, so we can decentralize the learning process and merge the learned skills afterwards [10]. Third, as discovered in my work [19], textual skills are transferable and reusable across models. Furthermore, retrieval-augmented generation [8] can further scale up this framework to numerous skills without changing the base LLM.

Model optimization, compilation and systems. The trend of foundation models drives the community to develop larger models and train them with more data, creating a growing demand for efficient and scalable systems. Based on my past experience in machine learning systems [15, 18], I would like to contribute to the open-source community with optimized implementations and systems. My past collaboration with researchers from high-performance computing suggests that there is a significant gap in time between cutting-edge machine learning research and highly-optimized implementations, since people only optimize the implementations once we know the algorithm is effective. A more scalable way is to develop an automatic solution for model optimization, i.e. compilation. I will collaborate with researchers from high-performance computing and compilers to investigate new strategies for operation fusion, automatic memory optimization and efficient batching, especially for large language models and structured data. I will also maintain an open-source platform for my research works if permitted by funding terms, to encourage reproduction, reuse and education.

References

[1] D. Beaini, S. Huang, J. A. Cunha, G. Moisescu-Pareja, O. Dymov, S. Maddrell-Mander, C. McLean, F. Wenkel, L. Müller, J. H. Mohamud, F. Wenkel, M. Craig, M. Koziarski, J. Lu, **Z., Zhu**, et al. Towards foundational models for molecular learning on large-scale multi-task datasets. *International Conference on Learning Representations*, 2024.

[2] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[3] M. Galkin, **Z., Zhu**, H. Ren, and J. Tang. Inductive logical query answering in knowledge graphs. *Advances in Neural Information Processing Systems*, 35:15230–15243, 2022.

[4] M. Galkin, X. Yuan, H. Mostafa, J. Tang, and **Z., Zhu**. Towards foundation models for knowledge graph reasoning. *International Conference on Learning Representations*, 2024.

[5] W. Hamilton, P. Bajaj, M. Zitnik, D. Jurafsky, and J. Leskovec. Embedding logical queries on knowledge hraphs. *Advances in neural information processing systems*, 31, 2018.

[6] X. Huang, M. R. Orth, İ. İ. Ceylan, and P. Barceló. A theory of link prediction via relational weisfeiler-leman. *arXiv preprint arXiv:2302.02209*, 2023.

[7] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR, 2023.

[8] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[9] M. Qu, **Z., Zhu**, and J. Tang. Reasoning on knowledge graphs: Symbolic or neural? `https://aaai2022kgreasoning.github.io/`, 2022. AAAI 2022 Tutorial.

[10] C. Raffel. Building machine learning models like open source software. *Communications of the ACM*, 66(2):38–40, 2023.

[11] H. Ren, W. Hu, and J. Leskovec. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. *International Conference on Learning Representations*, 2020.

[12] H. Ren, M. Galkin, M. Cochez, **Z., Zhu**, and J. Leskovec. Neural graph reasoning: Complex logical query answering meets graph databases. *arXiv preprint arXiv:2303.14617*, 2023.

[13] C. Shi, M. Xu, **Z., Zhu**, W. Zhang, M. Zhang, and J. Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *International Conference on Learning Representations*, 2020.

[14] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *International Conference on Learning Representations*, 2019.

[15] **Z., Zhu**, S. Xu, J. Tang, and M. Qu. Graphvite: A high-performance cpu-gpu hybrid system for node embedding. In *The World Wide Web Conference*, pages 2494–2504, 2019.

[16] **Z., Zhu**, Z. Zhang, L.-P. Xhonneux, and J. Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 34:29476–29490, 2021.

[17] **Z., Zhu**, M. Galkin, Z. Zhang, and J. Tang. Neural-symbolic models for logical queries on knowledge graphs. In *International Conference on Machine Learning*, pages 27454–27478. PMLR, 2022.

[18] **Z., Zhu**, C. Shi, Z. Zhang, S. Liu, M. Xu, X. Yuan, Y. Zhang, J. Chen, H. Cai, J. Lu, et al. Torchdrug: A powerful and flexible machine learning platform for drug discovery. *arXiv preprint arXiv:2202.08320*, 2022.

[19] **Z., Zhu**, Y. Xue, X. Chen, D. Zhou, J. Tang, D. Schuurmans, and H. Dai. Large language models can learn rules. *arXiv preprint arXiv:2310.07064*, 2023.

[20] **Z., Zhu**, X. Yuan, M. Galkin, S. Xhonneux, M. Zhang, M. Gazeau, and J. Tang. A*net: A scalable path-based reasoning approach for knowledge graphs. *Advances in Neural Information Processing Systems*, 2023.

[21] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

[22] 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*, 9:176–194, 2021.

[23] 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*, 35:24824–24837, 2022.

[24] M. Xu, Z. Zhang, J. Lu, **Z., Zhu**, Y. Zhang, M. Chang, R. Liu, and J. Tang. Peer: A comprehensive and multi-task benchmark for protein ssequence understanding. *Advances in Neural Information Processing Systems*, 35:35156–35173, 2022.

[25] M. Zhang, P. Li, Y. Xia, K. Wang, and L. Jin. Labeling trick: A theory of using graph neural networks for multi-node representation learning. *Advances in Neural Information Processing Systems*, 34:9061–9073, 2021.

[26] J. Zhao, L. Zhuo, Y. Shen, M. Qu, K. Liu, M. Bronstein, **Z., Zhu**, and J. Tang. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*, 2023.