Title
Make Knowledge Computable: Towards Differentiable Neural-Symbolic AI

Author
Hu, Ziniu

Publication Date
2023

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Make Knowledge Computable: Towards Differentiable Neural-Symbolic AI

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

Ziniu Hu

2023

ABSTRACT OF THE DISSERTATION

Make Knowledge Computable: Towards Differentiable Neural-Symbolic AI

by

Ziniu Hu
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2023
Professor Yizhou Sun, Co-Chair
Professor Kai-Wei Chang, Co-Chair

Recent deep learning methods could pack a vast amount of world knowledge into its parameters. However, they remain limited in their capacity to carry out symbolic reasoning over the memorized knowledge, such as answering complex questions that require both numerical and logical reasoning. On the other hand, symbolic AI excels in reasoning tasks but is inefficient when it comes to adapting to new knowledge. Existing efforts to combine these two areas typically focus on building parsing-based Neural-Symbolic systems. As the symbolic modules are not differentiable, these parsing-based systems cannot be trained end-to-end from raw data. Instead, they normally require extensive annotation of intermediate labeled programs and present significant scaling challenges.

My ultimate research goal is to enable neural model to interact with symbolic reasoning module in a differentiable manner, and to train such Neural-Symbolic model end-to-end without intermediate labels. To bring this vision about, I have conducted work on:

- **Designing Novel Reasoning Module**: design differentiable neural modules that can conduct symbolic reasoning, including knowledge graph reasoning [Zin+c; Zin+f]

and complex Logical inference [CZS].

- **Learning via Symbolic Self-Supervision**: train the neural model via self-supervision from structural and symbolic knowledge base without additional annotation [Zin+b; Zin+d; ZCS].

- **Generalizing across Domains**: the modular design of Neural-Symbolic system by its nature help to generalize better for Out-of-Distribution [Zin+g], Out-of-Vocabulary [Zin+a], cross-lingual [Che+] and cross-type [Yoo+].

Putting these pieces together, I am pursuing the ultimate vision to build end-to-end Neural-Symbolic system that has the capacity of reasoning, advancing to true human intelligence. In this thesis, I will first emphasize the significance of building such differentiable Neural-Symbolic AI, and then introduce three lines of my works, as well as the future challenges and opportunities.

The dissertation of Ziniu Hu is approved.

Wei Wang

Quanquan Gu

Adnan Darwiche

Kai-Wei Chang, Committee Co-Chair

Yizhou Sun, Committee Co-Chair

University of California, Los Angeles

2023

Two roads diverged in a wood, and I

I took the one less traveled by,

And that has made all the difference.

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGMENTS

I have been incredibly fortunate and privileged throughout my entire PhD life to have the opportunity to work with so many great collaborators.

I express my deepest appreciation to all those who gave me the opportunity to complete this thesis. I would like to give my profound thanks my academic advisors, Prof. Yizhou Sun and co-advisor Prof. Kai-Wei Chang. They not only provide profound help to my research but also teach me how to be a good researcher, insisting on the direction that I believe is correct, instead of blindly chasing hot topics. I am so grateful to have them as my PhD advisors.

I sincerely appreciate my thesis committee members, Prof. Wei Wang, Prof. Quanquan Gu, and Prof. Adnan Darwiche. Their insightful feedback, engaging suggestions, and constant encouragement were indispensable.

First and foremost, I am deeply grateful to the mentors and collaborators I met during my internships. Their unwavering support throughout this journey is something I will always cherish. My time at Google Research was as challenging as it was illuminating. I would like to express my sincere thank to Alireza Fathi, Ahmet Iscen, Sun Chen, Zirui Wang, David Ross, and Cordelia Schmid, at the Perception team. This sentiment echoes my experiences with Zhe Zhao, Xinyang Yi, Tiansheng Yao, Lichan Hong, and Ed H. Chi, at Brain team. A warm acknowledgment to Yichong Xu and Chenguang Zhu at Microsoft Cognitive Service Research, and especially to Yuxiao Dong and Kuansan Wang, who initiated our collaboration at Microsoft Redmond, providing me with immeasurable assistance throughout my PhD journey.

I am also grateful to work in a brilliant research group, including Ting Chen, Junheng Hao, Xuelu Chen, Yunsheng Bai, Kewei (Vivian) Cheng, Song Jiang, Yewen Wang, Patricia Xiao,

Roshni Iyer, Zijie Huang, Shichang Zhang, Zongyue Qin, Derek Xu , Arjun Subramonian, Zeyuan (Fred) Xu, Yanqiao Zhu. I also like to thank my other collaborators and friends: Xiusi Chen, Minji Yoon, Da Yin, Liunian Li, Lingxiao Wang, Difan Zou. The collective wisdom and experience of these collaborators have indelibly shaped my approach to research problems, guiding me to identify worthwhile paths and develop actionable plans to realize my goals.

I would like to express my gratitude to the staff at the University of California, Los Angeles, and all the institutions where I interned. Their support and understanding have been crucial in this endeavor.

And finally, I would like to express my heartfelt thanks to my parents, family and friends. Their encouragement and support have been my rock during the hardest time of my life. Wish all the bests to those I loved.

CHAPTER 1

Introduction

1.1 Motivation

The pursuit of machine-understandable world knowledge modelling has a long history in artificial intelligence. Earlier generations of AI systems, typically symbolic in nature, attempt to hard-code expert-level knowledge into programs. These systems, though powerful in answering complex queries[1], were primarily domain-specific, necessitating significant human effort to maintain the knowledge base. Furthermore, they lacked the ability to continuously learn new knowledge from data.

In recent years, Deep Learning has demonstrated remarkable capabilities in absorbing and storing vast amounts of world knowledge. By using end-to-end training methods on extensive data corpora, large-scale neural models can often outperform humans in a range of language and vision tasks such as machine translation and image captioning. However, these models often store knowledge implicitly within their neural network parameters. Consequently, they struggle to handle complex tasks requiring reasoning over symbolic knowledge. For instance, tasks like answering intricate questions that require multi-step thinking and logical reasoning, or synthesizing high-level hardware programs, which requires understanding structural and symbolic `C/C++` codes, predicting execution results, and

[1] https://www.ibm.com/ibm/history/ibm100/us/en/icons/watson/breakthroughs/

searching optimal programs in discrete space. For example, to answer the question "Which NBA teams in Texas have not won the championship after 2000.", even the powerful GPT4 model that already memorized the required knowledge could make mistakes, as it could fail to understand the logical constraint in the query.

To address these challenges, I am pursuing a novel direction. Instead of statically compiling world knowledge into model parameters, I aim to model this symbolic knowledge using a more modular design. This design will allow both neural models and symbolic AI modules to comprehend the knowledge, compute and conduct reasoning. This vision aligns with traditional **Neural-Symbolic AI systems** that bridge the worlds of neural networks and symbolic systems. These systems employ neural models to parse input queries x into symbolic programs z, such as SQL queries or arithmetic circuits. A symbolic module, like a numerical and logical solver, then uses z for planning, deduction, and reasoning, ultimately generating the answer y.

Despite their merits, most parsing-based Neural-Symbolic AI systems face a significant challenge. The symbolic modules are not differentiable, preventing the end-to-end training of the entire Neural-Symbolic model using only (x, y) pairs. Instead, most previous efforts needed to annotate the intermediate symbolic query z for training the neural module. For many real-world applications, obtaining high-quality intermediate labels is challenging, if not impossible, and the neural parser trained on a limited z training set often fails to generalize across different domains or distributions. This limitation has significantly curtailed the applicability of previous Neural-Symbolic AI systems.

1.2 My Research Overview

My ultimate research goal is to enable the neural model to interact with a symbolic reasoning module in a differentiable way, and to train this Neural-Symbolic model end-to-end without needing intermediate labels. To achieve this goal, my research has focused on:

1. **Designing Novel Reasoning Module**: To create an end-to-end Neural-Symbolic system, it is crucial to make the symbolic reasoning step differentiable. My earlier research efforts have been dedicated to designing differentiable neural modules that can reason over relational and structured graphs and interact with external knowledge graphs for complex question answering.

    - Chapter 2 presents the *Heterogensous Graph Transformer (**HGT**)* architecture that conducts reasoning over relational graph;
    - Chapter 3 presents *Knowledge Graph Empowered Language Model (**OREO-LM**)* that enables Language Model to interact with external Knowledge Graphs and conduct joint reasoning.

2. **Learning via Symbolic Self-Supervision**: Given the trend of deep learning models moving towards training on extensive unlabeled datasets, it is ideal for the proposed Neural-Symbolic models to learn from unlabeled data. However, the complex reasoning module often requires learning signals from structural knowledge instead of raw data. To address this, I propose several techniques to leverage the structural symbolic knowledge as self-supervision to pre-train neural models.

    - Chapter 4 presents algorithm that conducts *Generative Pre-training of Graph Neural Networks (**GPT-GNN**)*.
    - Chapter 5 presents *End-to-End Retrieval-Augmented Visual Language Pre-Training (**REVEAL**)* that encodes multiple knowledge sources into a unified memory and trains a visual-language model to learn to retrieve from it to answer complex visual questions.

3. **Generalizing across Domains**: A key advantage of symbolic reasoning over neural models is its ability to generalize better across distributions and domains. Therefore, by representing data in a compositional and structural manner, we can view each disentangled neural module as handling a specific functionality and change only a particular module when transitioning to a new domain or distribution.

- Chapter 6 presents a *Few-shot hierarchical encoder (**HiCE**)* that inferring Out-Of-Vocabulary word embedding through meta-learning.
- Chapter 7 presents a *Causal Representation Learning framework (**MT-CRL**)* for improving multi-task generalization by alleviating the spurious correlation.

Putting these pieces together, I am pursuing the ultimate vision to build end-to-end Neural-Symbolic system that has the capacity of reasoning, towards achieving true human-like intelligence.

1.3 My Research Contributions

My vision is supported by my prior research, which has led to more than 20 research papers published in top Machine Learning venues (NeurIPS, ICLR, AAAI), Data Mining venues (KDD, WWW, WSDM), and Nature Language Processing venues (ACL, EMNLP). Notably, I received the **Best Full Paper Award** at WWW 2019, **Best Student Paper Award** at DLG-KDD 2020, and **Best Paper Award** at SoCal-NLP 2022. Many models I design have been integrated into machine learning libraries such as Pytorch-Geometric and DGL[2], utilized in many industrial products, including Google Youtube Shorts recommendation, Microsoft Graph, Facebook hate speech detection, Tiktok & Toutial search engine and stock trend prediction service by Microsoft. The software tools I developed and open-sourced have received over 2000 stars in total on Github, and also served as core building blocks for many NSF research grants.

Summary of my publications. The following are my major Ph.D. research that are included in this thesis:

The content of Chapter 2 appears in:

[2]They have become basic building blocks for modern models for structured and geometric data and are widely used in academia and industry. My proposed HGT [Zin+c] model is used as official tutorial in PyG.

> [Zin+c] **Ziniu Hu**, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. "Heterogeneous Graph Transformer". In: *Proceedings of the ACM Web Conference (WWW 2020, mostly cited paper)*

The content of Chapter 3 appears in:

> [Zin+f] **Ziniu Hu**, Yichong Xu, Shuohang Wang, Ziyi Yang, Chengguang Zhu, Kai-Wei Chang, and Yizhou Sun. "Empowering Language Models with Knowledge Graph Reasoning for Question Answering". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2022, also best paper award in SoCal-NLP 2022)*

The content of Chapter 4 appears in:

> [Zin+b] **Ziniu Hu**, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. "GPT-GNN: Generative Pre-Training of Graph Neural Networks". In: *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2020, Oral)*

The content of Chapter 5 appears in:

> [Zin+d] **Ziniu Hu**, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A. Ross, and Alireza Fathi. "REVEAL:

> Retrieval-Augmented Visual-Language Pre-Training with Multi-Source Multimodal Knowledge Memory". In: *Conference on Computer Vision and Pattern Recognition (CVPR 2023, Highlight)*

The content of Chapter 6 appears in:

> [Zin+a] **Ziniu Hu**, Ting Chen, Kai-Wei Chang, and Yizhou Sun. "Few-Shot Representation Learning for Out-Of-Vocabulary Words". In: *Proceedings of the Association for Computational Linguistics (ACL 2019)*

The content of Chapter 7 appears in:

> [Zin+g] **Ziniu Hu**, Zhe Zhao, Xinyang Yi, Tiansheng Yao, Lichan Hong, Yizhou Sun, and Ed H Chi. "Improving Multi-Task Generalization via Regularizing Spurious Correlation". In: *Advances in Neural Information Processing Systems (NeurIPS 2022, Spotlight)*

I have also conducted (or made one of the major contributions to) the following research during my Ph.D. studies.

> [Zin+e] **Ziniu Hu**, Yang Wang, Qu Peng, and Hang Li. "Unbiased LambdaMART: An Unbiased Pairwise Learning-to-Rank Algorithm". In: *Proceedings of the ACM Web Conference (WWW 2019 with US Patent)*
>
> [Zou+] Difan Zou*, **Ziniu Hu*** (equal contribution), Yewen Wang, Song

Jiang, Yizhou Sun, and Quanquan Gu. "Layer-Dependent Importance Sampling for Training Deep and Large Graph Convolutional Networks". In: *Advances in Neural Information Processing Systems (NeurIPS 2019)*

[ZCS] **Ziniu Hu**, Kai-Wei Chang, and Yizhou Sun. "Relation-Guided Pre-Training for Open-Domain Question Answering". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-Finding 2021)*

[CZS] Xuelu Chen, **Ziniu Hu**, and Yizhou Sun. "Fuzzy Logic based Logical Query Answering on Knowledge Graphs". In: *AAAI Conference on Artificial Intelligence (AAAI 2022)*

I have also contributed to the following publications as a co-author.

[Che+] Zhenpeng Chen, Sheng Shen, Ziniu Hu, Xuan Lu, Qiaozhu Mei, and Xuanzhe Liu. "Emoji-Powered Representation Learning for Cross-Lingual Sentiment Classification". In: *Proceedings of the ACM Web Conference (WWW 2019 Best Full Paper Award)*

[Wan+] Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. "Improving Neural Language Generation with Spectrum Control". In: *International Conference on Learning Representations (ICLR 2020)*

[Don+] Yuxiao Dong, Ziniu Hu, Kuansan Wang, Yizhou Sun, and Jie Tang. "Heterogeneous Network Representation Learning". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2020)*

[Wei+] Tianxin Wei, Ziwei Wu, Ruirui Li, Ziniu Hu, Fuli Feng, Xiangnan He, Yizhou Sun, and Wei Wang. "Fast Adaptation for Cold-start Collaborative Filtering with Meta-learning". In: *IEEE International Conference on Data Mining (ICDM 2020)*

[Yin+] Da Yin, Liunian Harold Li, Ziniu Hu, Nanyun Peng, and Kai-Wei Chang. "Broaden the Vision: Geo-Diverse Visual Commonsense Reasoning". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*

[Yoo+] Minji Yoon, John Palowitch, Dustin Zelle, Ziniu Hu, Russ Salakhutdinov, and Bryan Perozzi. "Zero-shot Transfer Learning within a Heterogeneous Graph via Knowledge Transfer Networks". In: *Advances in Neural Information Processing Systems (NeurIPS 2022)*

Part I

Differentiable Symbolic Reasoning Module

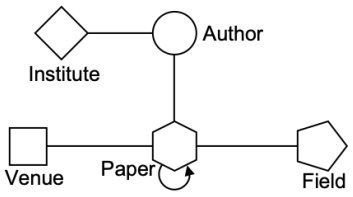
CHAPTER 2

HGT: Heterogeneous Graph Transformer

Currently, most existing GNNs are designed for homogeneous graphs, in which all nodes and edges belong to the same types, making them infeasible to represent more complex relational data, i.e., heterogeneous graphs. To solve this problem, we propose Heterogeneous Graph Transformer (HGT) for modeling web-scale heterogeneous and dynamic graphs. We first abstract the underlying symbolic knowledge in the heterogeneous graph as meta relation triplet. Then, we leverage the meta relation to parameterize the weight matrices for calculating attention over each edge, empowering HGT to maintain dedicated representations for different types of nodes and edges. HGT can incorporate information from high-order neighbors of different types through message passing across layers, so that it can automatically and implicitly learn and extract "meta paths" which are important for different downstream tasks. Extensive experiments on the Open Academic Graph, which contain 179 million nodes and 2 billion edges, show that the proposed HGT model consistently outperforms all the state-of-the-art GNN baselines by 9%–21% on various downstream tasks. In addition, HGT significantly enhances the accuracy of anomaly detection for the Microsoft Office Team, and have ranked ranks 1st on Stanford Open Graph Benchmark's MAG leaderboard for half year.

2.1 Introduction

Heterogeneous graphs have been commonly used for abstracting and modeling complex systems, in which objects of different types interact with each other in various ways. Some prevalent instances of such systems include academic graphs, Facebook entity graph, LinkedIn economic graph, and broadly the Internet of Things network [SH12]. For example, the Open Academic Graph (OAG) [Zha+19b] contains five types of nodes: papers, authors, institutions, venues (journal, conference, or preprint), and fields, as well as different types of relationships between them.



(a) The schema of heterogeneous academic networks

Author	*is_(first/last/other)_author_of*	Paper
Author	*is_affiliated_with*	Institute
Paper	*is_published_(conf/journal)_at*	Venue
Paper	*has_(L_1−L_5)_field_of*	Field
Paper	*has_citation_to*	Paper

(b) The meta relations of heterogeneous academic networks

Figure 2.1: The schema and meta relations of Open Academic Graph (OAG).

Over the past decade, a significant line of research has been explored for mining heterogeneous graphs. One of the classical paradigms is to define and use meta paths to model heterogeneous structures, such as PathSim [Sun+11] and metapath2vec [DCS17]. Recently, in view of graph neural networks' (GNNs) success [KW17; HYL17; Vel+18], there are several attempts to adopt GNNs to learn with heterogeneous networks [Sch+18; Zha+19a; Wan+19b; Yun+19]. However, these works face several issues: First, most of them involve the design of meta paths or variants for each type of heterogeneous graphs, requiring specific domain knowledge; Second, they either simply assume that different types of nodes/edges share the same feature and representation space or keep distinct non-sharing weights for either node type or edge type alone, making them insufficient to capture heterogeneous graphs' properties; Finally, their intrinsic design and implementation make them incapable of modeling Web-scale heterogeneous graphs.

In light of these limitations and challenges, we propose to study heterogeneous neural

networks with the goal of maintaining node- and edge-type dependent representations, avoiding customized meta paths, and being scalable to Web-scale heterogeneous graphs. In this work, we present the HGT architecture to deal with all these challenges.

To handle graph heterogeneity, we introduce the node- and edge-type dependent attention mechanism. Instead of parameterizing each type of edges, the heterogeneous mutual attention in HGT is defined by breaking down each edge $e = (s, t)$ based on its meta relation triplet, i.e., $\langle$ node type of s, edge type of e between s & t, node type of $t\rangle$. Figure 2.1 illustrates the meta relations of heterogeneous academic graphs. In specific, we use these meta relations to parameterize the weight matrices for calculating attention over each edge. As a result, nodes and edges of different types are allowed to maintain their specific representation spaces. Meanwhile, connected nodes in different types can still interact, pass, and aggregate messages without being restricted by their distribution gaps. Due to the nature of its architecture, HGT can incorporate information from high-order neighbors of different types through message passing across layers, which can be regarded as "soft" meta paths. That said, even if HGT take only its one-hop edges as input without manually designing meta paths, the proposed attention mechanism can automatically and implicitly learn and extract "meta paths" that are important for different downstream tasks.

We demonstrate the effectiveness and efficiency of the proposed HGT on the Web-scale Open Academic Graph comprised of 179 million nodes and 2 billion edges, making this the largest-scale representation learning yet performed on heterogeneous graphs. Experimental results suggest that HGT can significantly improve various downstream tasks over state-of-the-art GNN baselines by 9%–21%. We further conduct case studies to show the proposed method can indeed automatically capture the importance of implicit meta paths for different tasks.

2.2 Preliminaries and Related Work

In this section, we introduce the basic definition of heterogeneous graphs with network dynamics and review the recent development on graph neural networks (GNNs) and their heterogeneous variants. We also highlight the difference between HGT and existing attempts on heterogeneous graph neural networks.

2.2.1 Learning for Heterogeneous Graph

Heterogeneous graphs [SH12] (a.k.a., heterogeneous information networks) are an important abstraction for modeling relational data for many real-world complex systems. Formally, it is defined as:

Definition 1. Heterogeneous Graph: A heterogeneous graph is defined as a directed graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$ where each node $v \in \mathcal{V}$ and each edge $e \in \mathcal{E}$ are associated with their type mapping functions $\tau(v) : V \rightarrow \mathcal{A}$ and $\phi(e) : E \rightarrow \mathcal{R}$, respectively.

Meta Relation. For an edge $e = (s, t)$ linked from source node s to target node t, its meta relation is denoted as $\langle \tau(s), \phi(e), \tau(t) \rangle$. Naturally, $\phi(e)^{-1}$ represents the inverse of $\phi(e)$. The classical meta path paradigm [Sun+11; Sun+12; SH12] is defined as a sequence of such meta relation.

Notice that, to better model real-world heterogeneous networks, we assume that there may exist multiple types of relations between different types of nodes. For example, in OAG there are different types of relations between the *author* and *paper* nodes by considering the authorship order, i.e., "the first author of", "the second author of", and so on.

Dynamic Heterogeneous Graph. To model the dynamic nature of real-world (heterogeneous) graphs, we assign an edge $e = (s, t)$ a timestamp T, when node s connects to node t at T. If s appears for the first time, T is also assigned to s. s can be associated with multiple timestamps if it builds connections over time.

In other words, we assume that the timestamp of an edge is unchanged, denoting the time it is created. For example, when a paper published on a conference at time T, T will be assigned to the edge between the paper and conference nodes. On the contrary, different timestamps can be assigned to a node accordingly. For example, the *conference* node "WWW" can be assigned any year. WWW@1994 means that we are considering the first edition of WWW, which focuses more on internet protocol and Web infrastructure, while WWW@2020 means the upcoming WWW, which expands its research topics to social analysis, ubiquitous computing, search & IR, privacy and society, etc.

There have been significant lines of research on mining heterogeneous graphs, such as node classification, clustering, ranking and representation learning [SH12; Sun+11; Sun+12; DCS17], while the dynamic perspective of HGs has not been extensively explored and studied.

2.2.2 Graph Neural Networks

Recent years have witnessed the success of graph neural networks for relational data [KW17; Vel+18; HYL17]. Generally, a GNN can be regarded as using the input graph structure as the computation graph for message passing [Gil+17a], during which the local neighborhood information is aggregated to get a more contextual representation. Formally, it has the following form:

Definition 2. General GNN Framework: Suppose $H^l[t]$ is the node representation of node t at the (l)-th GNN layer, the update procedure from the (l-1)-th layer to the (l)-th layer is:

$$H^l[t] \leftarrow \underset{\forall s \in N(t), \forall e \in E(s,t)}{\textbf{Aggregate}} \left(\Psi \left(H^{l-1}[s]; H^{l-1}[t], e \right) \right) \tag{2.1}$$

where $N(t)$ denotes all the source nodes of node t and $E(s,t)$ denotes all the edges from node s to t.

There are two basic operators in GNNs: $\Psi(\cdot)$ and **Aggregate(·)**. $\Psi(\cdot)$ represents the

neighbor information extractor. It uses the target node's representation $H^{l-1}[t]$ and the edge e between the two nodes as query, and extract useful information from source node $H^{l-1}[s]$. **Aggregate(·)** serves as the aggregation function of the neighborhood information. The *mean, sum,* and *max* functions are often considered as the basic aggregation operators, and more sophisticated pooling and normalization functions can be also designed.

Under this framework, various (homogeneous) GNN architectures have been proposed due to its power for modeling relational data. For example, the graph convolutional network (GCN) proposed by Kipf *et al.* [KW17] averages the one-hop neighbor of each node in the graph, followed by a linear projection and non-linear activation operations. Hamilton *et al.* propose GraphSAGE that generalizes GCN's aggregation operation from *average* to *sum, max* and a *RNN unit*. Velickovi *et al.* 's graph attention network (GAT) [Vel+18] furthers this framework by introducing the attention mechanism into GNNs, which allows the model to assign different importance to nodes within the same neighborhood.

Heterogeneous GNNs Recently, studies have attempted to extend GNNs for modeling heterogeneous graphs. Schlichtkrull *et al.* [Sch+18] propose the relational graph convolutional networks (RGCN) to model knowledge graphs. RGCN keeps a distinct linear projection weight for each edge type. Zhang *et al.* [Zha+19a] present the heterogeneous graph neural networks (HetGNN) that adopts different RNNs for different node types to integrate multi-modal features. Wang *et al.* [Wan+19b] extend graph attention networks by maintaining different weights for different meta-path-defined edges. They also use high-level semantic attention to differentiate and aggregate information from different meta paths.

Though these methods have shown to be empirically better than the vanilla GCN and GAT models, they have not fully utilized the heterogeneous graphs' properties. All of them use either node type or edge type alone to determine GNN weight matrices. However, the node or edge counts of different types can vary greatly. For relations that don't have sufficient occurrences, it's hard to learn accurate relation-specific weights. To address this, we propose to consider parameter sharing for a better generalization. Given an edge

$e = (s, t)$ with its meta relation as $\langle \tau(s), \phi(e), \tau(t) \rangle$, if we use three interaction matrices to model the three corresponding elements $\tau(s), \phi(e)$, and $\tau(t)$ in the meta relation, then the majority of weights could be shared. For example, in "the first author of" and "the second author of" relationships, their source and target node types are both *author* to *paper*, respectively. In other words, the knowledge about *author* and *paper* learned from one relation could be quickly transferred and adapted to the other one. Therefore, we integrate this idea with the powerful Transformer-like attention architecture, and propose HGT.

To summarize, the key differences between HGT and existing attempts include:

1. Instead of attending on node or edge type alone, we use the meta relation $\langle \tau(s), \phi(e), \tau(t) \rangle$ to decompose the interaction and transform matrices, enabling HGT to capture both the common and specific patterns of different relationships using equal or even fewer parameters.

2. Different from most of the existing works that are based on customized meta paths, we rely on the nature of the neural architecture to incorporate high-order heterogeneous neighbor information, which automatically learns the importance of implicit meta paths.

3. Previous works don't take the dynamic nature of (heterogeneous) graphs into consideration, while we propose the relative temporal encoding technique to incorporate temporal information by using limited computational resources.

4. None of the existing heterogeneous GNNs are designed for and experimented with Web-scale graphs, we therefore propose the heterogeneous Mini-Batch graph sampling algorithm designed for Web-scale graph training, enabling experiments on the billion-scale Open Academic Graph.

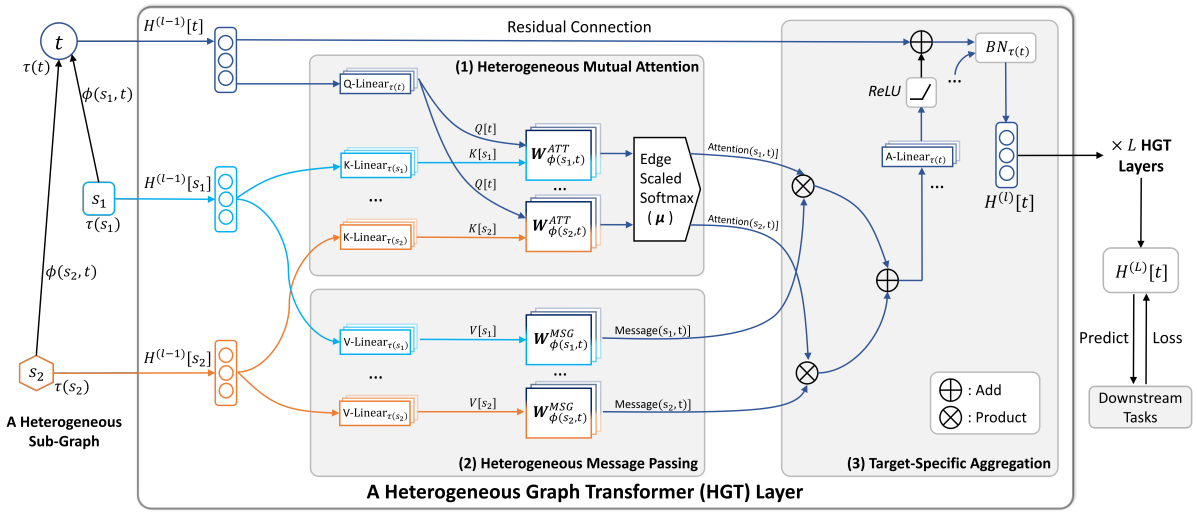


Figure 2.2: The Overall Architecture of Heterogeneous Graph Transformer. Given a sampled heterogeneous sub-graph with t as the target node, s_1 & s_2 as source nodes, the HGT model takes its edges $e_1 = (s_1, t)$ & $e_2 = (s_2, t)$ and their corresponding meta relations $< \tau(s_1), \phi(e_1), \tau(t) >$ & $< \tau(s_2), \phi(e_2), \tau(t) >$ as input to learn a contextualized representation $H^{(L)}$ for each node, which can be used for downstream tasks. Color decodes the node type. HGT includes three components: (1) meta relation-aware heterogeneous mutual attention, (2) heterogeneous message passing from source nodes, and (3) target-specific heterogeneous message aggregation.

2.3 Methodology

In this section, we present the Heterogeneous Graph Transformer (HGT). Its idea is to use the **meta relations** of heterogeneous graphs to parameterize weight matrices for the heterogeneous mutual attention, message passing, and propagation steps.

2.3.1 Heterogeneous Graphs and Meta Relation

Heterogeneous graphs [SH12] (a.k.a., heterogeneous information networks) are an important abstraction for modeling relational data and many real-world complex systems. Formally, a heterogeneous graph is defined as a directed graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$ where each node $v \in \mathcal{V}$ and each edge $e \in \mathcal{E}$ are associated with their type mapping functions $\tau(v) : V \rightarrow \mathcal{A}$ and $\phi(e) : E \rightarrow \mathcal{R}$, respectively.

Meta Relation. For an edge $e = (s, t)$ linked from source node s to target node t, its meta relation is denoted as $\langle \tau(s), \phi(e), \tau(t) \rangle$. Naturally, $\phi(e)^{-1}$ represents the inverse of $\phi(e)$. The classical meta path paradigm is defined as a sequence of such meta relations.

Notice that, to better model real-world heterogeneous networks, we assume that there may exist multiple types of relations between two nodes. For example, in OAG there could be different types of relations between the *author* and *paper* nodes by considering the authorship order, e.g., "the first author of" and "the last author of".

2.3.2 Heterogeneous Graph Transformer Architecture

Figure 2.2 shows the overall architecture of Heterogeneous Graph Transformer. Given a sampled heterogeneous sub-graph (Cf. Section 2.4), HGT extracts all linked node pairs, where target node t is linked by source node s via edge e. The goal of HGT is to aggregate information from s to get a contextualized representation for target node t. Such process can be decomposed into three components: *Heterogeneous Mutual Attention*, *Heterogeneous Message Passing* and *Target-Specific Aggregation*.

We denote the output of the (l)-th HGT layer as $H^{(l)}$, which is also the input of the $(l+1)$-th layer. By stacking L layers, we can get the node representations of the whole graph $H^{(L)}$, which can be used for end-to-end training or fed into downstream tasks.

Heterogeneous Mutual Attention. The first step is to calculate the mutual attention between source node s and target node t. We first give a brief introduction to the general attention-based GNNs as follows:

$$H^{l}[t] \leftarrow \underset{\forall s \in N(t), \forall e \in E(s,t)}{\textbf{Aggregate}} \Big(\textbf{Attention}(s, t) \cdot \textbf{Message}(s) \Big) \tag{2.2}$$

where there are three basic operators: **Attention**, which estimates the importance of each source node; **Message**, which extracts the message by using only the source node s; and **Aggregate**, which aggregates the neighborhood message by the attention weight.

Following this framework, we introduce **Heterogeneous Mutual Attention** mechanism. Given a target node t, and all its neighbors $s \in N(t)$, which might belong to different distributions, we want to calculate their mutual attention grounded by their **meta relations**, i.e., the $\langle \tau(s), \phi(e), \tau(t) \rangle$ triplets.

Inspired by the architecture design of Transformer [Vas+17], we map target node t into a Query vector, and source node s into a Key vector, and calculate their dot product as attention. The key difference is that the vanilla Transformer uses a single set of projections for all words, while in our case each meta relation should have a distinct set of projection weights. To maximize parameter sharing while still maintaining the specific characteristics of different relations, we propose to parameterize the weight matrices of the interaction operators into a source node projection, an edge projection, and a target node projection. Specifically, we calculate the h-head attention for each edge $e = (s, t)$ (See Figure 2.2 (1)) by:

$$\begin{aligned}
\textbf{Attention}_{HGT}(s, e, t) &= \underset{\forall s \in N(t)}{\text{Softmax}}\Big(\Big\Vert_{i \in [1,h]} \textit{ATT-head}^i(s, e, t) \Big) \\
\textit{ATT-head}^i(s, e, t) &= \Big(K^i(s)\ W^{ATT}_{\phi(e)}\ Q^i(t)^T \Big) \cdot \frac{\mu_{\langle \tau(s), \phi(e), \tau(t) \rangle}}{\sqrt{d}} \\
K^i(s) &= \text{K-Linear}^i_{\tau(s)}\Big(H^{(l-1)}[s] \Big) \\
Q^i(t) &= \text{Q-Linear}^i_{\tau(t)}\Big(H^{(l-1)}[t] \Big)
\end{aligned} \tag{2.3}$$

First, for the i-th attention head $\textit{ATT-head}^i(s, e, t)$, we project the $\tau(s)$-type source node s into the i-th *Key* vector $K^i(s)$ with a linear projection $\text{K-Linear}^i_{\tau(s)} : \mathbb{R}^d \to \mathbb{R}^{\frac{d}{h}}$, where h is the number of attention heads and $\frac{d}{h}$ is the vector dimension per head. Note that $\text{K-Linear}^i_{\tau(s)}$ is indexed by the source node s's type $\tau(s)$, meaning that each type of nodes has a unique linear projection to maximally model the distribution differences. Similarly, we also project the target node t with a linear projection $\text{Q-Linear}^i_{\tau(t)}$ into the $i-$th Query vector.

Next, we need to calculate the similarity between the Query vector $Q^i(t)$ and Key vector

$K^i(s)$. One unique characteristic of heterogeneous graphs is that there may exist different edge types (relations) between a node type pair, e.g., $\tau(s)$ and $\tau(t)$. Therefore, unlike the vanilla Transformer that directly calculates the dot product between the Query and Key vectors, we keep a distinct edge-based matrix $W^{ATT}_{\phi(e)} \in \mathbb{R}^{\frac{d}{h} \times \frac{d}{h}}$ for each edge type $\phi(e)$. In doing so, the model can capture different semantic relations even between the same node type pairs. Moreover, since not all the relationships contribute equally to the target nodes, we add a prior tensor $\mu \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{R}| \times |\mathcal{A}|}$ to denote the general significance of each meta relation triplet, serving as an adaptive scaling to the attention.

Finally, we concatenate h attention heads together to get the attention vector for each node pair. Then, for each target node t, we gather all attention vectors from its neighbors $N(t)$ and conduct normalization via softmax.

Heterogeneous Message Passing. Parallel to the calculation of mutual attention, we pass information from source nodes to target nodes (See Figure 2.2 (2)). Similar to the attention process, we would like to incorporate the meta relations of edges into the message passing process to alleviate the distribution differences of nodes and edges of different types. For a pair of nodes $e = (s, t)$, we calculate its multi-head **Message** by:

$$\textbf{Message}_{HGT}(s, e, t) = \Big\Vert_{i \in [1,h]} \mathit{MSG\text{-}head}^i(s, e, t) \tag{2.4}$$

$$\mathit{MSG\text{-}head}^i(s, e, t) = \text{M-Linear}^i_{\tau(s)}\Big(H^{(l-1)}[s]\Big)\ W^{MSG}_{\phi(e)}$$

To get the i-th message head $\mathit{MSG\text{-}head}^i(s, e, t)$, we first project the $\tau(s)$-type source node s into the i-th message vector with a linear projection $\text{M-Linear}^i_{\tau(s)} : \mathbb{R}^d \to \mathbb{R}^{\frac{d}{h}}$. It is then followed by a matrix $W^{MSG}_{\phi(e)} \in \mathbb{R}^{\frac{d}{h} \times \frac{d}{h}}$ for incorporating the edge dependency. The final step is to concat all h message heads to get the $\textbf{Message}_{HGT}(s, e, t)$ for each node pair.

Target-Specific Aggregation. With the heterogeneous multi-head attention and message calculated, we need to aggregate them from the source nodes to the target node (See

Figure 2.2 (3)). Note that the softmax procedure in Eq. 2.3 has made the sum of each target node t's attention vectors to one, we can thus simply use the attention vector as the weight to average the corresponding messages from the source nodes and get the updated vector $\widetilde{H}^{(l)}[t]$ as:

$$\widetilde{H}^{(l)}[t] = \bigoplus_{\forall s \in N(t)} \Big(\textbf{Attention}_{HGT}(s, e, t) \cdot \textbf{Message}_{HGT}(s, e, t) \Big).$$

This aggregates information to the target node t from all its neighbors (source nodes) of different feature distributions.

The final step is to map target node t's vector back to its type-specific distribution, indexed by its node type $\tau(t)$. To do so, we apply a linear projection $\text{A-Linear}_{\tau(t)}$ to the updated vector $\widetilde{H}^{(l)}[t]$, followed by a non-linear activation and residual connection [He+16] as:

$$H^{(l)}[t] = \sigma\Big(\text{A-Linear}_{\tau(t)} \widetilde{H}^{(l)}[t]\Big) + H^{(l-1)}[t]. \tag{2.5}$$

In this way, we get the l-th HGT layer's output $H^{(l)}[t]$ for the target node t. Due to the "small-world" property of real-world graphs, stacking the HGT blocks for L layers (L being a small value) can enable each node reaching a large proportion of nodes—with different types and relations—in the full graph. That is, HGT generates a highly contextualized representation $H^{(L)}$ for each node, which can be fed into any models to conduct downstream heterogeneous network tasks, such as node classification and link prediction.

Through the whole model architecture, we highly rely on using the **meta relation**—$\langle \tau(s), \phi(e), \tau(t) \rangle$—to parameterize the weight matrices separately. This can be interpreted as a trade-off between the model capacity and efficiency. Compared with the vanilla Transformer, our model distinguishes the operators for different relations and thus is more capable to handle the distribution differences in heterogeneous graphs. Compared with existing models that keep a distinct matrix for each meta relation as a whole, HGT's triplet parameterization can better leverage the heterogeneous graph schema to achieve parameter

sharing. On one hand, relations with few occurrences can benefit from such parameter sharing for fast adaptation and generalization. On the other hand, different relationships' operators can still maintain their specific characteristics by using a much smaller parameter set.

2.4 Training for Web-Scale Graphs

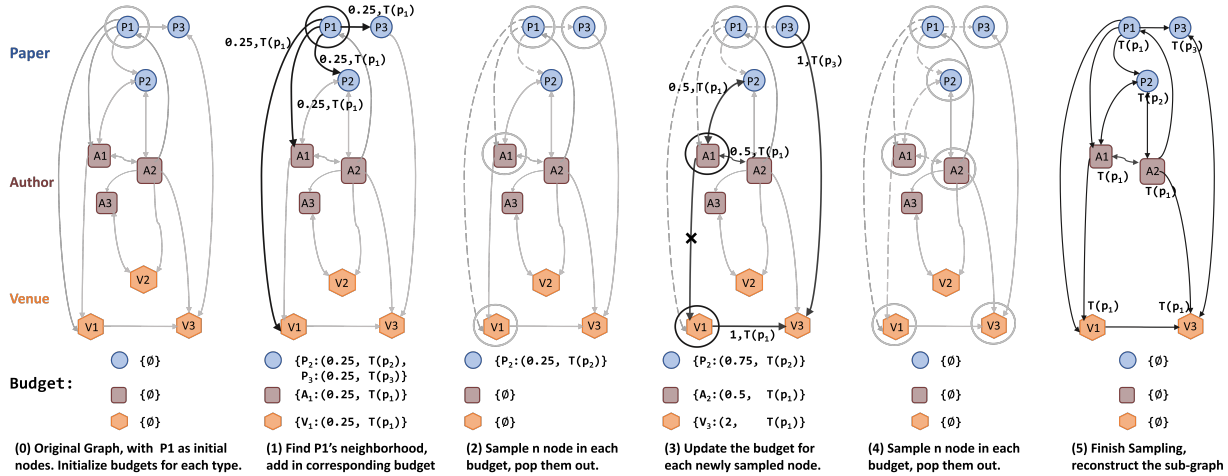


Figure 2.3: Heterogeneous Mini-Batch Graph Sampling Procedure with Inductive Timestamp Assignment.

In this section, we present an efficient Heterogeneous Mini-Batch Graph Sampling algorithm—HGSampling—to enable both HGT and traditional GNNs to handle Web-scale heterogeneous graphs with dynamic information. First, we propose an efficient Heterogeneous Mini-Batch Graph Sampling algorithm to generate informative sub-graphs. Second, we describe an inductive timestamp assignment method for keeping heterogeneous graphs' temporal information.

2.4.1 Heterogeneous Mini-Batch Graph Sampling

The original full-batch GNN [KW17] training requires the calculation of all node representations per layer, making it not scalable for Web-scale graphs. To address this issue,

different sampling-based methods [HYL17; CMX18; CZS18; Zou+] have been proposed to train GNNs on a subset of nodes. However, directly using them for heterogeneous graphs is prone to get sub-graphs that are extremely imbalanced regarding different node types, due to that the degree distribution and the total number of nodes for each type can vary dramatically.

As such, we propose a heterogeneous mini-batch graph sampling method (See Alg. 1) that is able to 1) keep a similar number of nodes and edges for each type and 2) keep the sampled sub-graph dense to minimize the information loss and reduce the sample variance. The basic idea is to keep a separate node budget $B[\tau]$ for each node type τ and to sample an equal number of nodes per type with an important sampling to reduce variance.

Given node t already sampled, we add all its direct neighbors into the corresponding budget with Algorithm 2, and add t's normalized degree to these neighbors in line 1, which will then be used to calculate the sampling probability. Such normalization is equivalent to accumulate the random walk probability of each sampled node to its neighborhood, avoiding the sampling being dominated by high-degree nodes. Intuitively, the higher such value, the more a candidate node is correlated with the currently sampled nodes, and thus should be given a higher probability to be sampled.

After the budget is updated, we then calculate the sampling probability in Algorithm 1 line 1, where we calculate the square of the cumulative normalized degree of each node s in each budget. As proved in [CMX18; Zou+], using such sampling probability can reduce the sampling variance. Then, we sample n nodes in node type τ by using the calculated probability, add them into the output node set, update its neighborhood to the budget, and remove it out of the budget in line 1-1. Repeating such procedure for L times, we get a sampled sub-graph with $L-$th depth from the initial nodes. Finally, we reconstruct the adjacency matrix among the sampled nodes. Using the above sampling algorithm, the sampled sub-graph contains a similar number of nodes per node type (based on the separate node budget), and is sufficiently dense to reduce the sampling variance (based on the normalized degree and importance sampling), and thus it is suitable for training GNNs

Algorithm 1 Heterogeneous Mini-Batch Graph Sampling

Adjacency matrix A for each $\langle \tau(s), \phi(e), \tau(t) \rangle$ relation pair; Output node Set OS; Sample number n per node type; Sample depth L.Sampled node set NS; Sampled adjacency matrix $\hat{A}$. $NS \leftarrow OS$ // Initialize sampled node set as output node set. Initialize an empty Budget B storing nodes for each node type with normalized degree. $t \in NS$ Add-In-Budget(B, t, A, NS) // Add neighbors of t to B. $l \leftarrow 1$ to L source node type $\tau \in B$ source node $s \in B[\tau]$ $prob^{(l-1)}[\tau][s] \leftarrow \frac{B[\tau][s]^2}{\|B[\tau]\|_2^2}$ // Calculate sampling probability for each source node s of node type τ. Sample n nodes $\{t_i\}_{i=1}^{n}$ from $B[\tau]$ using $prob^{(l-1)}[\tau]$. $t \in \{t_i\}_{i=1}^{n}$ $OS[\tau].add(t)$ // Add node t into Output node set. Add-In-Budget(B, t, A, NS) // Add neighbors of t to B. $B[\tau].pop(t)$ // Remove sampled node t from Budget. Reconstruct the sampled adjacency matrix $\hat{A}$ among the sampled nodes OS from A. OS and $\hat{A}$;

Algorithm 2 Add-In-Budget

Budget B storing nodes for each type with normalized degree; Added node t; Adjacency matrix A for each $\langle \tau(s), \phi(e), \tau(t) \rangle$ relation pair; Sampled node set NS.Updated Budget B. each possible source node type τ and edge type ϕ $\hat{D}_t \leftarrow 1 \,/\, len\Big(A_{\langle \tau, \phi, \tau(t) \rangle}[t]\Big)$ // get normalized degree of added node t regarding to $\langle \tau, \phi, \tau(t) \rangle$. source node s in $A_{\langle \tau, \phi, \tau(t) \rangle}[t]$ s has not been sampled ($s \notin NS$) s has no timestamp $s.time = t.time$ // Inductively inherit timestamp. $B[\tau][s] \leftarrow B[\tau][s] + \hat{D}_t$ // Add candidate node s to budget B with target node t's normalized degree. Updated Budget B

on Web-scale heterogeneous graphs.

2.4.2 Inductive Timestamp Assignment

Till now we have assumed that each node t is assigned with a timestamp $T(t)$. However, in real-world heterogeneous graphs, many nodes are not associated with a fixed time. Instead, we can assign different timestamps to it. We denote these nodes as *plain nodes*. For example, the WWW conference is held in both 1974 and 2019, and the WWW node in these two years has dramatically different research topics. Therefore, we need to decide which timestamp(s) to attach to the WWW node. Note that there also exist *event nodes* in heterogeneous graphs that have an explicit timestamp associated with them. For example, the paper node should be associated with its publication behavior and therefore attached to its publication date.

To address this issue, we propose an inductive timestamp assignment algorithm that assigns plain nodes timestamps based on the event nodes that they are linked with. The algorithm is shown in Algorithm 2 line 1. The basic idea is that we inherit the timestamp from event nodes to plain nodes. We examine whether the candidate source node is an event node. If yes, like a paper published at a specific year, we keep its timestamp for capturing temporal dependency. If no, like a conference that can be associated with any timestamp, we inductively assign the associated node's timestamp, such as the published year of its paper, to this plain node. In this way, we can adaptively assign timestamps during the sub-graph sampling procedure.

During the sampling, we can have multiple event nodes linked to one plain node, such as multiple papers published at WWW at different times. In this case, we treat the same node with different timestamps distinctly, which means that in Algorithm 2 line 1, we also use the associated timestamp as a judgment indicator. In this way, WWW@1974 and WWW@2019 can both occur in our sampled subgraph, linking to the same neighborhoods. With relative temporal encoding, the HGT model should learn to attend differently for these two WWW nodes towards all the papers published on it in different years.

An example of sampling a heterogeneous mini-batch academic graph as well as assigning timestamps is also illustrated in Figure 2.3. We start from a graph with Paper *P1* as the initial node. At each Step we update the budget by exploring the immediate neighbors of newly added nodes and then sample n (n=1) nodes for each budget. During this process, we inductively assign timestamps from target nodes to the plain source nodes, if the source nodes don't have fixed timestamps themselves (e.g., papers assign publication dates to venues). In this way, we can sample a dense sub-graph with a balanced number of nodes for each type and inductively assigned timestamps, which can be used to conduct efficient training for large-scale graphs.

2.5 Experiments

In this section, we evaluate the proposed HGT on the Open Academic Graph (OAG) [Zha+19b]—the largest publicly available heterogeneous academic dataset. We conduct the Paper-Field prediction, Paper-Venue prediction, and Author Disambiguation tasks. We also take case studies to demonstrate how HGT can automatically learn and extract meta paths that are important for downstream tasks[1].

2.5.1 Experimental Setup

Web-scale Datasets To examine HGT and its real-world applications, we use the Open Academic Graph (OAG) [Sin+15; Zha+19b] as our experimental basis. OAG consists of more than 178 million nodes and 2.236 billion edges, making them at least two–three magnitudes larger than the other datasets that are commonly used in existing heterogeneous GNN and heterogeneous graph mining studies. Besides, it is far more distinguishable than previously wide-adopted small citation graphs used in GNN studies, such as Cora, Citeseer and Pubmed [KW17; Vel+18], which only contain thousands of nodes.

Tasks and Evaluation. We evaluate the HGT model on four different real-world downstream tasks: the prediction of Paper–Field (L_1), Paper–Field (L_2), and Paper–Venue, and Author Disambiguation. The goal of the first three node classification tasks is to predict the correct L_1 and L_2 fields that each paper belongs to or the venue it is published at, respectively. We use different GNNs to get the contextual node representation of the paper and use a softmax output layer to get its classification label. For author disambiguation, we select all the authors with the same name and their associated papers. The task is to conduct link prediction between these papers and candidate authors. After getting the paper and author node representations from GNNs, we use a Neural Tensor Network to get the probability of each author-paper pair to be linked.

[1]The dataset and code are publicly available at https://github.com/acbull/pyHGT.

GNN Models		GCN	RGCN	GAT	HetGNN	HAN	$\text{HGT}_{\text{noHeter}}$	HGT
Paper–Field (L_1)	NDCG	.508±.141	.511±.128	.534±.103	.543±.084	.544±.096	.571±.089	**.595±.089**
	MRR	.556±.136	.565±.105	.610±.096	.616±.076	.622±.092	.649±.081	**.675±.082**
Paper–Field (L_2)	NDCG	.218±.074	.228±.046	.239±.049	.236±.062	.242±.051	.250±.045	**.258±.052**
	MRR	.222±.067	.232±.052	.248±.045	.250±.053	.258±.049	.262±.057	**.271±.064**
Paper–Venue	NDCG	.265±.066	.276±.051	.270±.057	.262±.071	.280±.062	.297±.058	**.306±.064**
	MRR	.258±.070	.236±.047	.260±.052	.246±.059	.278±.067	.293±.061	**.317±.048**
Author Disambiguation	NDCG	.612±.064	.619±.057	.645±.063	.649±.052	.660±.049	.668±.059	**.683±.066**
	MRR	.738±.042	.755±.048	.797±.044	.803±.058	.821±.056	.835±.043	**.847±.043**

Table 2.1: Experimental results of different methods on Open Academic Graph (OAG).

For all tasks, we use papers published before the year 2015 as the training set, papers between 2015 and 2016 for validation, and papers between 2016 and 2019 as testing. We choose NDCG and MRR, which are two widely adopted ranking metrics [Liu11; Li14], as the evaluation metrics. All models are trained for 5 times and, the mean and standard variance of test performance are reported.

Baselines. We compare HGT with several state-of-the-art GNNs, including both homogeneous—GCN [KW17] and GAT [Vel+18]—and heterogeneous GNNs—RGCN [Sch+18], HetGNN [Zha+19a], and HAN [Wan+19b]. To examine the effectiveness of the heterogeneous components in our model, we also propose the $\text{HGT}_{\text{noHeter}}$ model, which uses the same set of weights for all meta relations, as the ablation study. All baselines as well as our own model are implemented via the PyTorch Geometric (PyG) package [FL19].

Input Features. As we don't assume the feature of each node type belongs to the same distribution, we are free to use the most appropriate features to represent each type of nodes. For each paper, we use a pre-trained XLNet [Yan+19b; Wol+19] to get the representation of each word in its title. We then average them weighted by each word's attention to get the title representation for each paper. The initial feature of each author is then simply an average of his/her published papers' representations. For the field, venue, and institute nodes, we use the metapath2vec model [DCS17] to train their node embeddings by reflecting the heterogeneous network structures.

The homogeneous GNN baselines assume the node features belong to the same distribution, while our feature extraction doesn't fulfill this assumption. To make a fair comparison, we add an adaptation layer between the input features and all used GNNs. This module simply conducts different linear projections for nodes of different types. Such a procedure can be regarded to map heterogeneous data into the same distribution, which is also adopted in literature [Zha+19a; Wan+19b].

Implementation Details. We use 256 as the hidden dimension throughout the neural networks for all baselines. For all multi-head attention-based methods, we set the head number as 8. All GNNs keep 3 layers so that the receptive fields of each network are exactly the same. All baselines are optimized via the AdamW optimizer [LH19] with the Cosine Annealing Learning Rate Scheduler [LH17]. For each model, we train it for 200 epochs and select the one with the lowest validation loss as the reported model. We use the default parameters used in GNN literature and donot tune hyper-parameters.

2.5.2 Experimental Results

We summarize the experimental results of the proposed model and baselines in Table 4.1. All experiments for the four tasks are evaluated in terms of NDCG and MRR. It shows that in terms of both metrics, the proposed HGT model significantly and consistently outperforms all baselines for all tasks. Take, for example, the Paper–Field (L_1) classification task, HGT achieves performance gains over baselines by 9–19% in terms of NDCG and 9–21% in terms of MRR (i.e., the performance difference divided by the baseline performance). When compared to HetGNN and HAN—the two dedicated heterogeneous GNN baselines, on average, the relative NDCG improvements of HGT for all four tasks are 8% and 6%, respectively. Moreover, HGT has fewer parameters and comparable batch time than all the heterogeneous graph neural network baselines, including RGCN, HetGNN, and HAN. This suggests that by modeling heterogeneous edges according to their meta relation schema, we are able to have better generalization with fewer resource consumption.

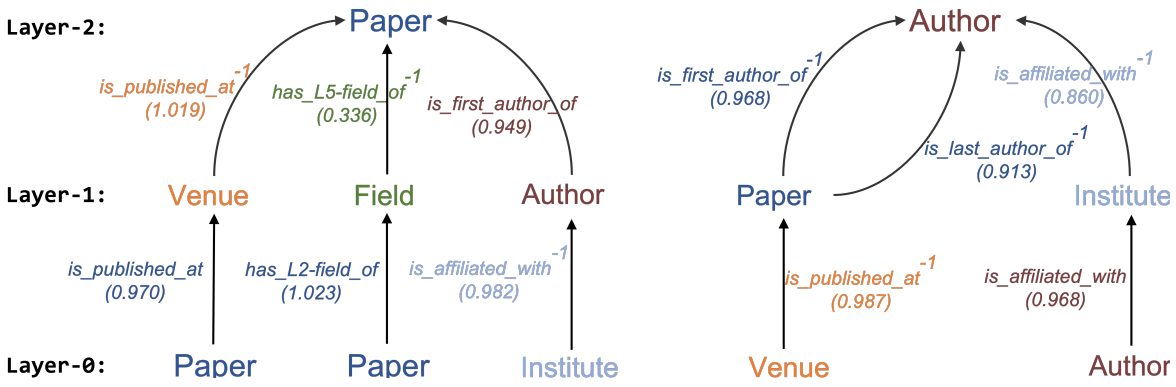


Figure 2.4: Hierarchy of the learned meta relation attention.

Ablation Study. The core component in HGT is the meta relation parameterization. To further analyze its effect, we conduct an ablation study. $\text{HGT}_{\text{noHeter}}$ only maintains a single set of parameters for all relations, which is equivalent to the vanilla Transformer applied on graphs. We can see that after removing this component, the NDCG performance drops 3.2%, demonstrating the importance of our meta relation parameterization.

Besides, we also try to implement a baseline that keeps a unique weight matrix for each relation. However, such a baseline contains too many parameters so that our experimental setting doesn't have enough GPU memory to optimize it. This also indicates that using the meta relation to parameterize weight matrices can achieve competitive performance with fewer resources.

Visualize Meta Relation Attention To illustrate how the incorporated meta relation schema can benefit the heterogeneous message passing process, we pick the schema that has the largest attention value in each of the first two HGT layers and plot the meta relation attention hierarchy tree in Figure 2.4. For example, to calculate a paper's representation, ⟨Paper, $is\_published\_at$, Venue, $is\_published\_at^{-1}$, Paper⟩, ⟨Paper, $has\_L_2\_field\_of$, Field, $has\_L_5\_field\_of^{-1}$, Paper⟩, and ⟨Institute, $is\_affiliated\_with^{-1}$, Author, $is\_first\_author\_of$, Paper⟩ are the three most important meta relation sequences, which can be regarded as meta paths *PVP*, *PFP*, and *IAP*, respectively. Note that these meta paths and their importance are automatically learned from the data without manual design. Another example of calculating an author node's representation is shown on the right.

Such visualization demonstrates that HGT is capable of implicitly learning to construct important meta paths for specific downstream tasks, without manual customization.

2.6 Summary

In this paper, we propose the HGT architecture for modeling Web-scale heterogeneous graphs. We leverage the meta relation to parameterize the weight matrices for calculating attention over each edge, empowering HGT to maintain dedicated representations for different types of nodes and edges.

CHAPTER 3

OREO-LM: Knowledge Graph Reasoning Empowered Language Model

Answering open-domain questions requires world knowledge about in-context entities. As pre-trained Language Models (LMs) lack the power to store all required knowledge, external knowledge sources, such as knowledge graphs, are often used to augment LMs. In this work, we propose knOwledge REasOning empowered Language Model (OreoLM), which consists of a novel Knowledge Interaction Layer that can be flexibly plugged into existing Transformer-based LMs to interact with a differentiable Knowledge Graph Reasoning module collaboratively. In this way, LM guides KG to walk towards the desired answer, while the retrieved knowledge improves LM. By adopting OreoLM to RoBERTa and T5, we show significant performance gain, achieving state-of-art results in the *Closed-Book* setting. The performance enhancement is mainly from the KG reasoning's capacity to infer missing relational facts. In addition, OreoLM provides reasoning paths as rationales to interpret the model's decision.

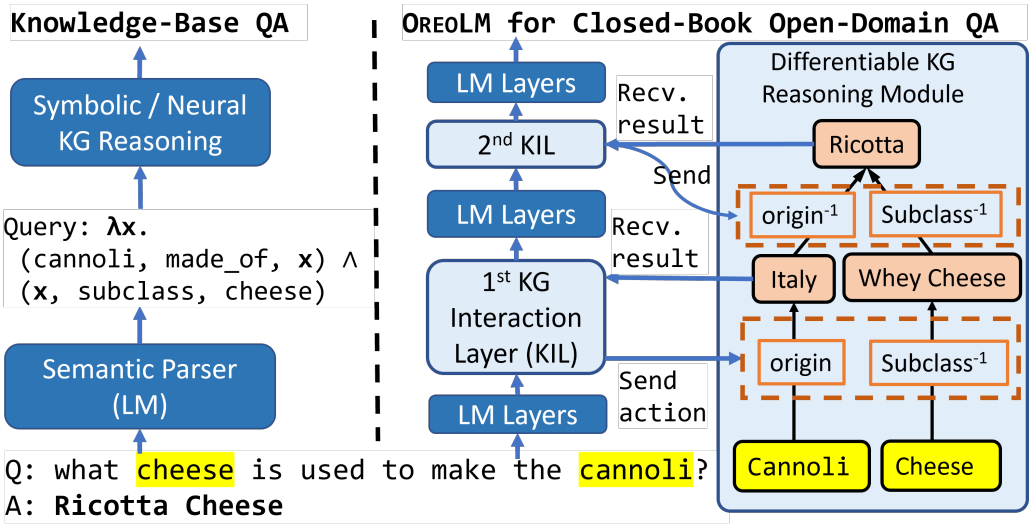


Figure 3.1: An Illustrative figure of OreoLM. Compared with previous KBQA systems that stack reasoner on top of LM, OreoLM enables interaction between the two.

3.1 Introduction

Open-Domain Question Answering (ODQA), one of the most knowledge-intensive NLP tasks, requires QA models to infer out-of-context knowledge to the given single question. Following the pioneering work by Chen, Fisch, Weston, and Bordes [Che+17], ODQA systems often assume to access an external text corpus (e.g., Wikipedia) as an external knowledge source. Due to the large scale of such textual knowledge sources (e.g., 20GB for Wikipedia), it cannot be encoded in the model parameters. Therefore, most works retrieve relevant passages as knowledge and thus named *Open-Book* models [RRS20], with an analogy of referring to textbooks during an exam. Another line of *Closed-book* models [RRS20] assume knowledge could be stored implicitly in parameters of Language Models (LM, e.g. BERT [Dev+19a] and T5 [Raf+20]). These LMs directly generate answers without retrieving from an external corpus and thus benefit from faster inference speed and simpler training. However, current LMs still miss a large portion of factual knowledge [PWS20; LSR21], and are not competitive with *Open-Book* models.

To improve the knowledge coverage of LM, one natural choice is to leverage knowledge stored in Knowledge Graph ($\mathcal{KG}$, e.g. FreeBase [Bol+08] and WikiData [VK14]), which

explicitly encodes world knowledge via relational triplets between entities. There are several good properties of $\mathcal{KG}$: 1) a $\mathcal{KG}$ triplet is a more abstract and compressed representation of knowledge than text, and thus $\mathcal{KG}$ could be stored in memory and directly enhance LM without using an additional retrieval model; 2) the structural nature of $\mathcal{KG}$ could support logical reasoning [RHL20] and infer missing knowledge through high-order paths [LMC11; Das+18]. Taking the question "what cheese is used to make the desert cannoli?" as an example, even if this relational fact is missing in $\mathcal{KG}$, we could still leverage high-order relationships, e.g., both Ricotta Cheese and Cannoli are specialties in Italy, to infer the answer "Ricotta Cheese."

In light of the good properties of $\mathcal{KG}$, there are several efforts to build Knowledge Base Question Answering (KBQA) systems. As is illustrated in Figure 3.1(a), most KBQA models use LM as a parser to map textual questions into a structured form (e.g., SQL query or subgraph), and then based on $\mathcal{KG}$, the queries could be executed by symbolic reasoning [Ber+13] or neural reasoning (e.g. Graph Neural Networks) [SBC19] to get the answer. Another recent line of research [Ver+21; Yu+20a] tries to encode the knowledge graph as the *memory* into LM parameters. However, for most methods discussed above, LM is not interacting with $\mathcal{KG}$ to correctly understand the question, and the answer is usually restricted to a node or edge in $\mathcal{KG}$.

In this paper, we propose knOwledge REasOning empowered Language Model (OreoLM), a model architecture that can be applied to Transformer-based LMs to improve *Closed-Book* ODQA. As is illustrated in Figure 3.1(b), the key component is the Knowledge Interaction Layers (KIL) inserted amid LM layers, which is like cream filling within two waffles, leading to our model's name Oreo. KIL interacts with a $\mathcal{KG}$ reasoning module, in which we maintain different reasoning paths for each entity in the question. We formulate the retrieval and reasoning process as a contextualized *random walk* over the $\mathcal{KG}$, starting from the in-context entities. Each KIL is responsible for one reasoning step. It first predicts a relation distribution for every in-context entity, and then the $\mathcal{KG}$ reasoning module traverses the graph following the predicted relation distribution. The reasoning result in each step is

summarized as a weighted averaged embedding over the retrieved entities from the traversal.

By stacking T layers of KIL, OreoLM can retrieve entities that are T-hop away from in-context entities and help LM to answer open questions that require out-of-context knowledge or multi-hop reasoning. The whole procedure is fully differentiable, and thus OreoLM learns and infers in an end-to-end manner. We further introduce how to pre-train OreoLM over unlabelled Wikipedia corpus. In addition to the salient entity span masking objective, we introduce two self-supervised objectives to guide OreoLM to learn better entity and relation representations and how to reason over them.

We test OreoLM with RoBERTa and T5 as our base LMs. By evaluating on several single-hop ODQA datasets in *closed-book* setting, we show that OreoLM outperforms existing baselines with fewer model parameters. Specifically, OreoLM helps more for questions with missing relations in $\mathcal{KG}$, and questions that require multi-hop reasoning. We further show that OreoLM can serve as a backbone for *open-book* setting and achieves comparable performance compared with the state-of-the-art QA systems with dedicated design. In addition, OreoLM has better interpretability as it can generate reasoning paths for the answered question and summarize general relational rules to infer missing relations.

This key contributions are as follows:

1. We propose OreoLM to integrate symbolic knowledge graph reasoning with neural LMs. Different from prior works, OreoLM can be seamlessly plugged into existing LMs.

2. We pretrain OreoLM with RoBERTa and T5 to on the Wikipedia corpus. OreoLM can bring significant performance gain on ODQA.

3. OreoLM offers interpretable reasoning paths for answering the question and high-order reasoning rules as rationales.

3.2 Preliminaries and Related Work

Open-Domain Question Answering (ODQA) gives QA model a single question without any context and asks the model to infer out-of-context knowledge. Following the pioneering work by Chen, Fisch, Weston, and Bordes [Che+17], most ODQA systems assume the model can access an external text corpus (e.g. Wikipedia). Due to the large scale of web corpus (20GB for Wikipedia), it could not be simply encoded in the QA model parameters, and thus most works propose a *Retrieval-Reader* pipeline, by firstly index the whole corpus and use a *retriever* model to identify which passage is relevant to the question; then the retrieved text passage concatenate with question is re-encoded by a seperate *reader* model (e.g., `LM`) to predict answer. As the knowledge is outside of model parameter, Roberts, Raffel, and Shazeer [RRS20] defines these methods as *Open-book*, with an analogy to referring textbooks during exam. *Closed-book* QA models (mostly a single `LM`) try to answer open questions without accessing external knowledge. This setting is much harder as it requires `LM` to memorize all pertinent knowledge in its parameters, and even recent `LM`s with much larger model parameters is still not competitive to state-of-the-art *Open-book* models.

Knowledge-augmented Language Models explicitly incorporate external knowledge (e.g. knowledge graph) into `LM` [Yu+20c]. Overall, these approaches can be grouped into two categories: The first one is to explicitly inject knowledge representation into language model pre-training, where the representations are pre-computed from external sources [Zha+19c; Liu+21; HSC21]. For example, ERNIE [Zha+19c] encodes the pre-trained TransE [Bor+13] embeddings as input. The second one is to implicitly model knowledge information into language model by performing knowledge-related tasks, such as entity category prediction [Yu+20a] and graph-text alignment [Ke+21]. For example, JAKET [Yu+20a] jointly pre-trained both the KG representation and language representation by adding entity category and relation type prediction self-supervised tasks.

There also exists several QA works using $\mathcal{KG}$ to help ODQA. For example, Asai,

Hashimoto, Hajishirzi, Socher, and Xiong [Asa+20] and Min, Chen, Zettlemoyer, and Hajishirzi [Min+19] expand the entity graph following wikipedia hyperlinks or triplets in knowledge base. Ding, Zhou, Chen, Yang, and Tang [Din+19] extract entities from current context via entity-linking and turn them into a cognitive graph, and a graph neural network is applied on top of it to extract answer. Dhingra, Zaheer, Balachandran, Neubig, Salakhutdinov, and Cohen [Dhi+20] and Lin, Sun, Dhingra, Zaheer, Ren, and Cohen [Lin+21] construct an entity-mention bipartite graph and then model the QA reasoning as graph traversal by filtering only the contexts that are relevant to the question. Lin, Chen, Chen, and Ren [Lin+19], Feng, Chen, Lin, Wang, Yan, and Ren [Fen+20] and Yasunaga, Ren, Bosselut, Liang, and Leskovec [Yas+21] parse the question into a sub-graph of knowledge base, and apply graph neural networks as reasoner for extracting one of the entities as the answer.

To encode knowledge (significantly smaller than the web corpus) as *memory* into `LM` parameter, a line of works try compressed knowledge including QA pairs [Che+22a; Lew+21; Yu+22c], entity embedding [Fév+20] and reasoning cases [Das+21; Das+22]. There's also several works utilizing Knowledge Graph ($\mathcal{KG}$) to augment `LM`. FILM [Ver+21] turns $\mathcal{KG}$ triplets into memory. Given a question, `LM` retrieves most relevant triplet as answer. GreaseLM [Zha+22c] propose to interact `LM` with $\mathcal{KG}$ via a interaction node.

Preliminary We denote a Knowledge Graph $\mathcal{KG} = \big(\mathcal{E}, \mathcal{R}, \mathcal{A} = \{A_r\}_{r\in\mathcal{R}}\big)$, where each $e \in \mathcal{E}$ and $r \in \mathcal{R}$ is entity node and relation label. $A_r \in \{0,1\}^{|\mathcal{E}|\times|\mathcal{E}|}$ is a sparse adjacency matrix indicating whether relation r holds between a pair of entities. The task of knowledge graph reasoning aims at answering a factoid query $(s, r, ?)$, i.e., which target entity has relation r with the source entity s. If $\mathcal{KG}$ is complete, we could simply get answers by checking the adjacency matrix, i.e., $\{\forall t : A_r[s,t] = 1\}$. For incomplete $\mathcal{KG}$ where many relational facts are missing, path-based reasoning approaches [LMC11; XHW17; Das+18] have been proposed to answer the one-hop query via finding multi-hop paths. For example, to answer the query $(s, \text{Mother}, ?)$, a path $s \xrightarrow{\text{Father}} j \xrightarrow{\text{Wife}} t$ could reach the target answer t.

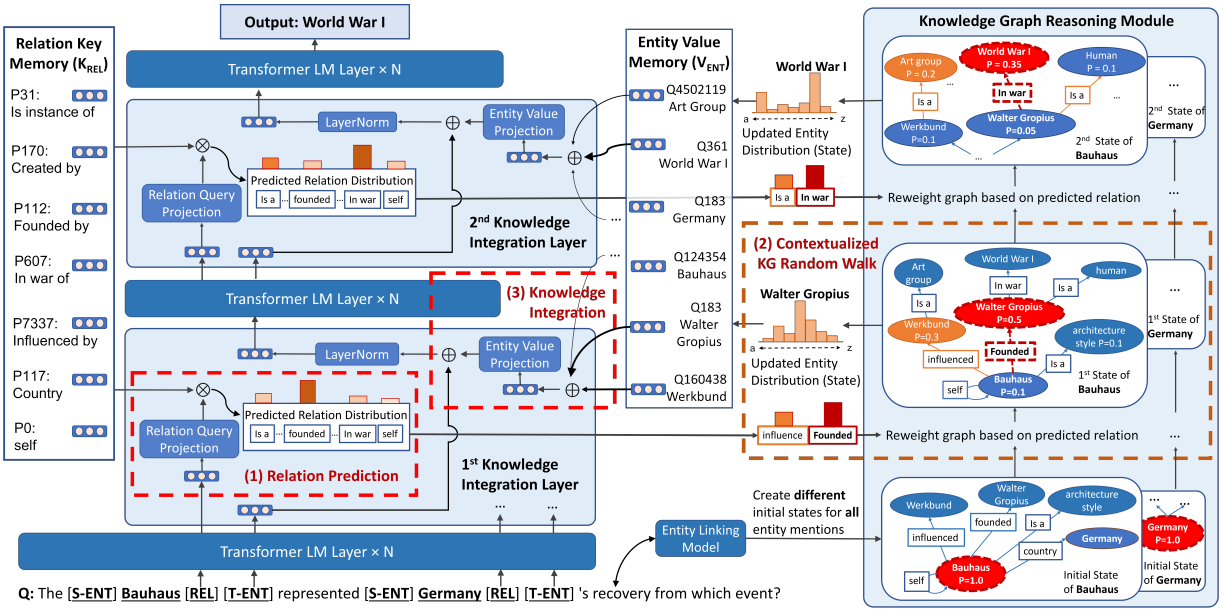


Figure 3.2: **Model architecture of OreoLM**. Three key procedures are highlighted in red dotted box: 1) **Relation Prediction** (Sec. 3.3.1.1): Knowledge Interaction Layers (KIL) predicts relation action for each entity mention. 2) **One-step State Transition** (Sec. 3.3.1.2): Based on the predicted relation, $\mathcal{KG}$ re-weights each graph and conduct contextualized random walk to update entity distribution state. 3) **Knowledge Integration** (Sec. 3.3.2): An weighted aggregated entity embedding is added into a placeholder token as retrieved knowledge.

In this paper we try to integrate symbolic $\mathcal{KG}$ reasoning into neural LMs and help it deal with ODQA problems.

3.3 Methodology

We illustrate the overall architecture of OreoLM in Figure 5.2. All the light blue blocks are our added components to support $\mathcal{KG}$ reasoning, while the dark blue Transformer layers are knowledge-injected LM. The key component of OreoLM for conducting $\mathcal{KG}$ reasoning is the Knowledge Interaction Layers (KIL), which are added amid LM layers to enable deeper interaction with the $\mathcal{KG}$.

Given a question q = "The Bauhaus represented Germany's recovery from which event?", QA model needs to extract knowledge about all n in-context entity mentions $M = \{m_i\}_{i=1}^{n}$,

e.g., the history of "Germany" at the time when "Bauhaus" is founded, to get the answer a = "World War I". Such open-domain Q&A can be abstracted as $P(a|q, M)$.

Starting from each mentioned entity m_i, we desire the model to learn to walk over the graph to retrieve relevant knowledge and form a T-length reasoning path for answering this question, where T is a hyper-parameter denote the longest reasoning path required to answer the questions. We define each reasoning path starting from the entity mention m_i as a chain of entities (states) random variables $\rho_i = \{e_i^t\}_{t=0}^T$, where each mentioned entity is the initial state, i.e., $e_i^0 = m_i$. The union of all paths for this question is defined as $\boldsymbol{\varrho} = \{\rho_i\}$, which contains the reasoning paths from each mentioned entity to answer the question.

OreoLM factorizes $\mathbf{P}\big(a|q, M\big)$ by incorporating possible paths $\boldsymbol{\varrho}$ as a latent variable, yielding:

$$
\begin{aligned}
\mathbf{P}\big(a|q, M\big) &= \sum\nolimits_{\boldsymbol{\varrho}} \mathbf{P}\big(\boldsymbol{\varrho}|q, \{m_i\}_{i=1}^n\big) \cdot \mathbf{P}\big(a|q, M, \boldsymbol{\varrho}\big) \\
&= \sum\nolimits_{\boldsymbol{\varrho}} \Big(\prod_{i=1}^{n} \mathbf{P}(\rho_i|q, m_i)\Big) \cdot \mathbf{P}\Big(a|q, \{m_i, \rho_i\}_{i=1}^n\Big) \\
&= \sum\nolimits_{\boldsymbol{\varrho}} \Big(\prod_{i=1}^{n}\prod_{t=1}^{T} \underbrace{\mathbf{P}\big(e_i^t|q, e_i^{<t}\big)}_{\mathcal{KG}\text{ Reasoning (3.3.1)}} \Big) \underbrace{\mathbf{P}\Big(a|q, \{e_i^{0:T}\}_{i=1}^n\Big)}_{\text{knowledge-injected LM (3.3.2)}}
\end{aligned}
$$

We assume (1) reasoning paths starting from different entities are generated independently; and (2) reasoning paths can be generated autoregressively.

In this way, the QA problem can be decomposed into two entangled steps: 1) $\mathcal{KG}$ Reasoning, which autoregressively walks through the graph to get a path ρ_i starting from each entity mention m_i; and 2) knowledge-injected LM, which benefits from the reasoning paths to obtain the out-context knowledge for answer prediction.

The relational path ρ_i in $\mathcal{KG}$ Reasoning requires the selection of next entity e_i^t at each step t. We further decompose it into two steps: 1.a) relation prediction, in which LM is involved to predict the next-hop relation based on the current state and context; and 1.b)

the non-parametric state transition, which is to predict the next-hop entity based on the $\mathcal{KG}$ and the predicted relation. Formally:

$$\underbrace{\mathbf{P}\big(e_i^t|q,e_i^{<t}\big)}_{\mathcal{KG}\text{ Reasoning (3.3.1)}} = \sum_r \underbrace{\mathbf{P}_{rel}\big(r_i^t|q,e_i^{<t}\big)}_{\text{relation prediction (3.3.1.1)}} \cdot \underbrace{\mathbf{P}_{walk}\big(e_i^t|r_i^t,e_i^{<t}\big)}_{\text{contextualized random walk (3.3.1.2)}}$$

We keep track of the entity distribution at each step t via the probability vector[1] $\boldsymbol{\pi}_i^{(t)} \in \mathcal{R}^{|\mathcal{E}|}$, with $\boldsymbol{\pi}_i^{(t)}[e]$ being the probability of staying at entity e, i.e., $\mathbf{P}\big(e_i^t = e|q,e_i^{<t}\big)$.

We highlight the three procedures in red dotted box in Figure 5.2. We take the first reasoning step starting from entity mention "Bauhaus" as an example. In the first red box within `KIL`, we predict which relation action should be taken for entity "Bauhaus", and send the prediction (e.g. "Founded") to $\mathcal{KG}$. In the second red box, $\mathcal{KG}$ re-weights the graph and conducts contextualized random walk to update entity distribution, where "Walter" has the highest probability. Finally, weighted by the entity distribution, an aggregated entity embedding is sent back to `KIL` and added into a placeholder token as the knowledge, so the later `LM` layer knows to focus on the retrieved "Walter". We introduce these steps in the following.

Input Initially, we first identify all N entity mentions $\{m_i\}_{i=1}^N$ in the input question q as well as the corresponding $\mathcal{KG}$ entities[2].. For each mention m_i we add three special tokens as the interface for Knowledge Interaction Layers (`KIL`) to send instruction and receive knowledge: we add a [`S-ENT`] token before, and [`REL`], [`T-ENT`] tokens after each entity mention m_i. `KIL` can be flexibly inserted into arbitrary `LM` intermediate layer. By default, we just insert each `KIL` every N Transformer-based `LM` layers, thus the input to the t-th `KIL` are contextualized embeddings of each token k as $\texttt{LM}_k^{(t)}$, including added special tokens.

[1]Throughout the paper, all vectors are row-vectors

[2]For Wikipedia pretraining, we use the ground-truth entity label as one-hot initialization for $\boldsymbol{\pi}_i^0$. For downstream tasks we use GENRE [Cao+21] to get top 5 entity links.

3.3.1 LM involved $\mathcal{KG}$ Reasoning

We first introduce the reasoning process $\mathbf{P}\big(e_i^t|q,e_i^{<t}\big)=\sum_r \mathbf{P}\big(r_i^t|q,e_i^{<t}\big)\cdot\mathbf{P}\big(e_i^t|r_i^t,e_i^{<t}\big)$.

3.3.1.1 Relation Prediction.

For each entity mention m_i, we desire to predict which relation action should take r_i^t as instruction to transit state. We define the predicted relation probability vector $\mathrm{b}_i^{(t)} = \mathbf{P}_{rel}\big(r_i^t|q,e_i^{<t}\big)\in\mathcal{R}^{|\mathcal{R}|}$ representing the relation distribution to guide walking through the graph. Denote the corresponding [REL] token as $\texttt{REL}[i]$ (and similarly for other special tokens). The contextual embedding $\texttt{LM}^{(t)}_{\texttt{REL}[i]}$ encode the relevant information in question q that hints next relation. We maintain a global relation key memory $\texttt{K}_{rel}\in\mathbb{R}^{|\mathcal{R}|\times d}$ storing each relation's d-dimentional embedding. To calculate similarity, we first get relation query $Q^{(t)}_{\texttt{REL}[i]}$ by projecting relation token's embedding into the same space of key memory via a projection head Q-Proj[3] followed by a LayerNorm (abbreviated as LN), and then calculate dot-product similarity followed by softmax:

$$Q^{(t)}_{\texttt{REL}[i]} = \mathrm{LN}^{(t)}\big(\text{Q-Proj}^{(t)}(\texttt{LM}^{(t)}_{\texttt{REL}[i]})\big), \tag{3.1}$$

$$\mathrm{b}_i^{(t)} = \mathbf{P}_{rel}\big(r_i^t|q,e_i^{<t}\big) = \mathrm{Softmax}\big(Q^{(t)}_{\texttt{REL}[i]}\ \texttt{K}_{rel}^T\big). \tag{3.2}$$

Note that the relation queries $\texttt{LM}^{(t)}_{\texttt{REL}[i]}$ are different for every mention m_i and reasoning step t depending on the context, and thus the the relation distributions $\mathrm{b}_i^{(t)}$ gives contextualized predictions based on the question q. The predicted relations are sent to the knowledge graph reasoning module as instruction to conduct state transition.

[3]We denote a non-linear MLP projection as X-Proj$(h) = W_2^X\sigma(W_1^X h + b_1) + b_2$, where X have different instantiations.

3.3.1.2 Contextualized KG Random Walk

Next, we introduce how we conduct state transition $\mathbf{P}_{walk}\left(e_i^t | r_i^t, e_i^{<t}\right)$. One classic transition algorithm is random walk, which is a special case of markov chain, i.e. the transition probability only depends on previous state. Consider a state at entity s, the probability walking to target t is $\frac{1}{deg(s)}$ if $A[s,t] = 1$. Based on it, we define the Markov transition matrix for random walk as $M_{rw} = \mathbf{D}_A^{-1} A$, where the degree matrix $\mathbf{D}_A \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is defined as the diagonal matrix with the degrees $deg(1), \ldots, deg(|\mathcal{E}|)$ on the diagonal. With random walk Markov matrix M_{rw} we can transit the state distribution as: $\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}^{(t-1)} M$, The limitation of random walk is that the transition strategy is not dependent on the question q. We thus propose a Contextualized Random Walk (w).

Based on the predicted relation distribution $\mathrm{b}_i^{(t)}$, we calculate a different weighted adjacency matrix $\widetilde{A}_i^{(t)} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ by adjusting the edge weight:

$$\widetilde{A}_i^{(t)} = \sum\nolimits_{r\in\mathcal{R}} w_r \cdot \mathrm{b}_{i,r}^{(t)} \cdot A_r, \tag{3.3}$$

$$M_{crw,i}{}^{(t)} = \mathbf{D}_{\widetilde{A}_i^{(t)}}^{-1} \widetilde{A}_i^{(t)}, \quad \forall i \in [1, N]. \tag{3.4}$$

where w_r is a learnable importance weight for relation r that helps solving downstream tasks, and $\mathrm{b}_{i,r}^{(t)}$ is the probability corresponding to relation r in $\mathrm{b}_i^{(t)}$. With the transition matrix $M_{crw,i}{}^{(t)}$, the state transition is defined as $\boldsymbol{\pi}_i^{(t)} = \boldsymbol{\pi}_i^{(t-1)} M_{crw,i}^{(t)}$.

wallows each reasoning path ρ_i to have its transition matrix. However, as the total number of entity nodes $|\mathcal{E}|$ could be huge (e.g., 5M for WikiData), we cannot afford to update the entire adjacency matrix for every in-batch mention. We thus adopt a scatter-gather pipeline to implement graph walking as shown in Algorithm 3. We first gather the entity and relation probability to each edge, and then scatter the probability to target nodes. This allows us to simultaneously conduct message passing with modified adjacency weight $\widetilde{A}_i^t$ for all entity mention m_i in parallel.

The complexity is # of in-batch entities times # of edges in T-hop subgraph starting from these entities, i.e., $\mathcal{O}(n \times \#\text{edge})$, and thus this operation is not expensive.

Algorithm 3 Pytorch Pseudocode of CRW

```
def ContextualizedRandomWalk(
    i_init, KG,   # initial entity index and Graph
    w_deg, w_rel, # inv(degree) and relation weights
    p_ent, p_rel  # entity and predicted relation dis-
                  # tribution tensor @ t-th step.
): -> FloatTensor
    # Get <src, rel, tgt> edge list of k-hop subgraph
    i_src, i_rel, i_tgt = k_hop_subgraph(i_init, KG)
    # Gather entity and relation probability to edge
    p_src  = (p_ent * w_deg)[:, i_src] # N x n_edge
    p_rel  = (p_rel * w_rel)[:, i_rel] # N x n_edge
    p_edge = l1_normalize(p_src * p_rel, dim=1)
    # Scatter edge probability to target node
    p_ent  = scatter_add(src=p_edge, idx=i_tgt, dim=1)
    return p_ent  #(t+1)-th step's entity distribution
```

3.3.2 Knowledge-Injected LM

After we get the updated entity distribution $\boldsymbol{\pi}_i^{(t)}$, we want to inject such information back to the LM without harming its overall structure. We maintain a global entity embedding value memory $\mathtt{V}_{ent} \in \mathbb{R}^{|\mathcal{E}| \times d}$ storing entity embeddings. We only consider the entities within the sampled local subgraph in each batch. We thus get an entity index list $\mathbf{I}$ as the query to sparsely retrieve a set of candidate entity embeddings and then aggregate them weighted by entity distribution and embedding table. We then use a Value Projection block to map the aggregated entity embedding into the space of LM, and then directly add the transformed embedding back to the output of T-ENT.

$$V_i^{(t)} = \text{V-Proj}^{(t)}\big(\boldsymbol{\pi}_i^{(t)} \cdot \mathtt{V}_{ent}[\mathbf{I}]\big), \tag{3.5}$$

$$\widehat{\mathtt{LM}}^{(t)}_{\mathtt{T\text{-}ENT}[i]} = \text{LN}^{(t)}\big(\mathtt{LM}^{(t)}_{\mathtt{T\text{-}ENT}[i]} + V_i^{(t)}\big). \tag{3.6}$$

Then, we just take all $\widehat{\mathtt{LM}}^{(t)}_{\mathtt{T\text{-}ENT}}$ as input to next Transformer-based LM layer to learn the interaction between the retrieved knowledge with in-context words via self-attention.

By repeating the KIL for T times, the final representation $\widehat{\mathtt{LM}}^{T}$ is conditioned on the reasoning paths $\rho_i = e_i^{0:T}$, which reaches entities that are T-hop away from initial entity m_i in the question. Finally, we can predict the answer of open questions $\mathbf{P}\big(a|q, \{e_i^{0:T}\}_{i=1}^n\big)$ by taking knowledge-injected representation $\widehat{\mathtt{LM}}^{T}$ for span extraction, entity prediction or

direct answer generation.

3.3.3 Pre-Train OreoLM to Conduct Reasoning

The design of OreoLM allows end-to-end training given QA datasets. However, due to the small coverage of knowledge facts for existing QA datasets, we need to pretrain OreoLM on a large-scale corpus to get good entity embeddings.

Salient Span Masking One straightforward approach is to use Salient Span Masking (SSM) objective [Guu+20] masks out entities or noun tokens requiring specific out-of-context knowledge. We mainly mask out entities for guiding OreoLM to reason. Instead of randomly masking entity mentions, we explicitly sample a set of entity IDs and mask every mentions linking to these entities. This could prevent the model copy the entity from the context to fill in the blank. We also follow [Yan+19c] to mask out consecutive token spans. We then calculate the cross-entropy loss on each salient span masked (SSM) token as $\mathcal{L}_{SSM}$.

3.3.3.1 Weakly Supervised Training of KIL

Ideally, OreoLM can learn all the entity knowledge and how to access the knowledge graph by solely optimizing $\mathcal{L}_{SSM}$. However, without a good initialization of entity and relation embeddings, KIL makes a random prediction, and the retrieved entities by $\mathcal{KG}$ reasoning are likely to be unrelated to the question. In this situation, KIL does not receive meaningful gradients to update the parameters, and LM learns to ignore the knowledge. To avoid this cold-start problem and provide entity and relation embedding a good initialization, We utilize the following two external signals as self-supervised guidance.

Entity Linking Loss To initialize the large entity embedding tables in $\mathtt{V}_{ent}$, we use other entities that are not masked as supervision. Similar to Févry, Baldini Soares, FitzGerald, Choi, and Kwiatkowski [Fév+20], we force the output embedding of [S-ENT] token before

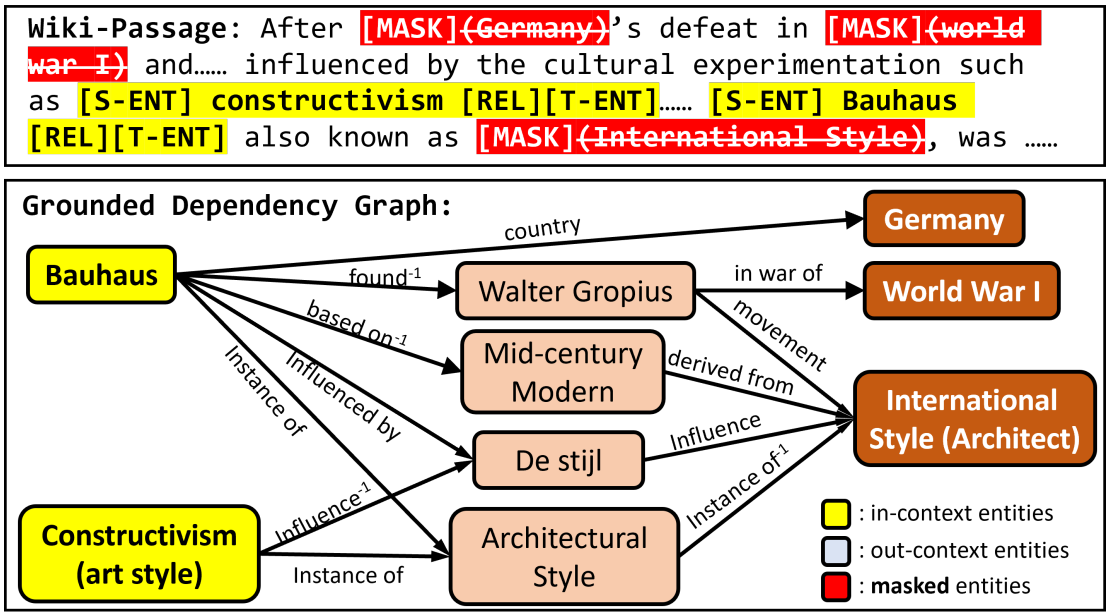


Figure 3.3: Pre-training sample w/ golden reasoning path.

the first KIL followed by a projection head E-Proj to be close to its corresponding entity embedding:

$$
\begin{aligned}
E_{\texttt{S-ENT}[i]} &= \mathrm{LN}\big(\text{E-Proj}(\texttt{LM}^{(1)}_{\texttt{S-ENT}[i]})\big), \\
\mathbf{P}^{(0)}_{ent}\big(e|m_i, q\big) &= \mathrm{Softmax}\big(E_{\texttt{S-ENT}[i]}\ \texttt{V}_{ent}[\mathbf{I}]^T\big), \\
\mathcal{L}_{ent} &= \sum\nolimits_{m_i} -\log \mathbf{P}^{(0)}_{ent}\big(e|m_i, q\big) \cdot \boldsymbol{\pi}^0_i[\mathbf{I}].
\end{aligned}
$$

Similar to Section 3.3.2, we only consider entities within the batch, denoted by index $\mathbf{I}$. This contrastive loss guides each entity's embedding $\texttt{V}_{ent}[e]$ closer to all its previously mentioned contextualized embedding, and thus memorizes those context as a good initialization for later knowledge integration.

Weakly Supervised Relation Path Loss Entity mentions within each Wikipedia passage are naturally grounded to WikiData $\mathcal{KG}$. Therefore, after we mask out several entities, we can utilize the $\mathcal{KG}$ to get all reasoning paths from other in-context entities to the masked entities as weakly supervised relation labels.

Name	Number	dimension	#param (M)
Number of Entity	4,947,397	128	633
Number of Relation	2,008	768	1.5
Number of Edges	45,217,947	-	47

Table 3.1: Statistics and parameter of $\mathcal{KG}$ Memory.

Formally, we define a **Grounded Dependency Graph** $\mathcal{DG}$, which contains all reasoning paths within T-step from other in-context entities to masked entities, and then define $R_{\mathcal{DG}}(m_i, t)$ as the set of all relations over every edges for entity mention m_i at t-th hop. Based on it, we define the weakly supervised relation label $q_i^{(t)} \in \mathbb{R}^{|\mathcal{R}|}$ as the probabilistic vector which uniformly distributed on each relation in set. Note that we call uniformly-weighted $q_i^{(t)}$ as weakly supervised because 1) some paths lead to multiple entities rather than only the target masked entity; 2) the correct relation is dependent on the context. Therefore, $q_i^{(t)}$ only provides all potential candidates for reachability, and more fine-grained signals for reasoning should be learned from unsupervised $\mathcal{L}_{SSM}$. We adopt a list-wise ranking loss to guide the model to assign a higher score on these relations than others.

$$\mathcal{L}_{rel} = \sum\nolimits_{m_i} \sum\nolimits_{t=1}^{T} -\log \mathbf{P}_{rel}^{(t)}\big(r|m_i, q\big) \cdot q_i^{(t)}.$$

Overall, $\mathcal{L}_{ent}$ and $\mathcal{L}_{rel}$ provide OREOLM with good initialization of the large $\mathcal{KG}$ memory. Afterward, via optimizing $\mathcal{L}_{SSM}$, the reasoning paths that provide informative knowledge receive a positive gradient, guiding OREOLM to reason.

3.4 Experiments

The proposed `KIL` layers can be pugged into most Transformer-based Language Models without hurting its original structure. In this paper, we experiment with both encoder-based `LM`, i.e. RoBERTa-base ($d = 768, l = 12$), and encoder-decoder `LM`, i.e. T5-base ($d = 768, l = 12$) and T5-large ($d = 1024, l = 24$). For all `LM`s, add 1 `KIL` layer or 2 `KIL` layers to the encoder layers. The statistics of $\mathcal{KG}$ are shown in Table 3.1. Altogether,

Models	#param	NQ	WQ	TQA	ComplexWQ	HotpotQA
T5 (Base)	0.22B	25.9	27.9	29.1	11.6	22.8
+ **OreoLM** (T=1)	0.23B + 0.68B	28.3	30.6	32.4	20.8	24.1
+ **OreoLM** (T=2)	0.24B + 0.68B	28.9	31.2	33.7	23.7	26.3
T5 (Large)	0.74B	28.5	30.6	35.9	16.7	25.3
+ **OreoLM** (T=1)	0.75B + 0.68B	30.6	32.8	39.1	24.5	28.2
+ **OreoLM** (T=2)	0.76B + 0.68B	**31.0**	**34.3**	**40.0**	**27.1**	**31.4**
T5-3B [RRS20]	3B	30.4	33.6	43.4	-	27.8
T5-11B [RRS20]	11B	32.6	37.2	50.1	-	30.2

Table 3.2: **Closed-Book Generative QA** performance of Encoder-Decoder LM on Single- and Multi-hop Dataset.

it takes about 0.67B parameter for $\mathcal{KG}$ memory, which is affordable to load as model parameter. We pre-train all LMs using the combination of $\mathcal{L}_{SSM}$, $\mathcal{L}_{ent}$ and $\mathcal{L}_{rel}$ for 200k steps on 8 V100 GPUs, with a batch size of 128 and default optimizer and learning rate in the original paper, taking approximately one week to finish pre-training of T5-large model, and 1-2 days for base model.

3.4.1 Evaluate for *Closed-Book* QA

OreoLM is designed for improving *Closed-Book* QA, so we first evaluate it in this setting.

Generative QA Task Following the hyperparameters and setting in [RRS20], we directly fine-tune the T5-base and T5-large augmented by our OreoLM on the three single-hop ODQA datasets: Natural Question **(NQ)** [Kwi+19], WebQuestions **(WQ)** [Ber+13] and TriviaQA **(TQA)** [Jos+17]. To test OreoLM's ability to solve complex questions, we also evaluate on two multi-hop QA datasets, i.e. **Complex WQ** [TB18] and **HotpotQA** [Yan+18].

Experimental results are shown in Table 3.2. We use Exact Match accuracy as the metric for all the datasets. On the three single-hop ODQA datasets, OreoLM with 2 KIL blocks achieves 3.3 absolute accuracy improvement to T5-base, and 3.4 improvement to T5-large. Compared with T5 model with more model parameters (e.g., T5-3B and T5-11B),

our T5-large augmented by OreoLM could outperform T5-3B on NQ and WQ datasets. In addition, OreoLM could use the generated reasoning path to interpret the model's prediction.

For the two multi-hop QA datasets, the performance improvement brought by OreoLM is more significant, i.e., 7.8 to T5-base and 8.2 to T5-large. Notably, by comparing the T5-3B and T5-11B's performance on HotpotQA (we take results from [Che+22a]), T5-large augmented by OreoLM achieves 1.2 higher than T5-11B. This shows that OreoLM is indeed very effective for improving *Closed-Book* QA performance, especially for complex questions.

Entity Prediction Task Encoder-based `LM` (i.e. RoBERTa) in most cases cannot be directly used for *Closed-Book* QA, but more serve as reader to extract answer span. However, Verga, Sun, Baldini Soares, and Cohen [Ver+21] propose a special evaluation setting as *Closed-Book Entity Prediction*. They add a single [`MASK`] token after the question, and use its output embedding to classify WikiData entity ID. This restricts that answers must be entities that are covered by WikiData, which they call *WikiData-Answerable* questions. We follow Verga, Sun, Baldini Soares, and Cohen [Ver+21] to use such reduced version of WebQuestionsSP **(WQ-SP)** [Yih+15] and TriviaQA **(TQA)** as evaluation dataset, and finetune the RoBERTa (base) model augmented by OreoLM to classify entity ID. We mainly compare OreoLM with EaE [Fév+20] and FILM [Ver+21], which are two $\mathcal{KG}$ memory augmented `LM`. We also run experiments on KEPLER [Wan+21a], a RoBERTa model pre-trained with knowledge augmented task.

Experimental results are shown in Table 3.3. Similar to the observation reported by Verga, Sun, Baldini Soares, and Cohen [Ver+21], adding $\mathcal{KG}$ memory for this entity prediction task could significantly improve over vanilla `LM`, as most of the factual knowledge required to predict entities are stored in $\mathcal{KG}$. By comparing with FILM [Ver+21], which is the state-of-the-art model in this setup, OreoLM with reasoning step ($T = 2$) outperforms FILM by 2.9, with smaller memory consumption.

Models	#param (B)	WQ-SP	TQA
EaE [Fév+20]	0.11 + 0.26	62.4	24.4
FILM [Ver+21]	0.11 + 0.72	78.1	37.3
KEPLER [Wan+21a]	0.12	48.3	24.1
RoBERTa (Base)	0.12	43.5	21.3
+ **OreoLM** (T=1)	0.12 + 0.68	80.1	39.7
+ **OreoLM** (T=2)	0.13 + 0.68	**80.9**	**40.3**
Ablation Studies			
RoBERTa + Concat KB + $\mathcal{L}_{SSM}$	0.12	47.1	22.6
+ **OreoLM** (T=2) **w/o PT**	0.13 + 0.68	46.9	22.7
w. $\mathcal{L}_{SSM}$	0.13 + 0.68	51.9	26.8
w. $\mathcal{L}_{SSM}$ + $\mathcal{L}_{ent}$	0.13 + 0.68	68.4	35.7

Table 3.3: **Closed-Book Entity Prediction** performance of Encoder `LM` on WikiData-Answerable Dataset.

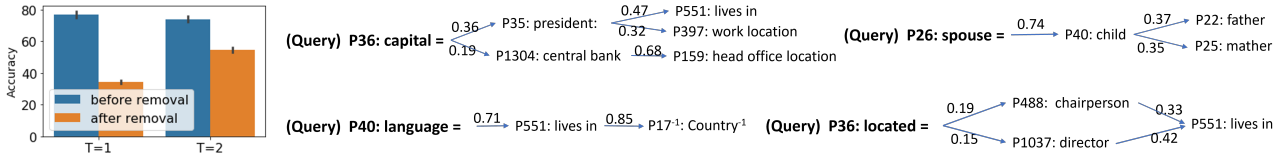


Figure 3.4: **Testing the reasoning capacity of OreoLM to infer missing relations**. On the **left**, the barplot shows the transfer performance on EQ before and after removing relation edges, OreoLM ($T = 2$) is less influenced. On the **right** shows reasoning paths (rules) automatically generated by OreoLM for each missing relation.

3.4.2 Analyze $\mathcal{KG}$ Reasoning Module

In our previous studies, we find that using a higher reasoning step, i.e. $T = 2$, generally performs better than $T = 1$. We hypothesize that the $\mathcal{KG}$ we use has many missing one-hop facts, and high-order reasoning helps recover them and empowers the model to answer related questions. To test whether OreoLM indeed can infer missing facts, we use **EntityQuestions (EQ)** [Sci+21], which is a synthetic dataset by mapping each WikiData triplet to natural questions. We take RoBERTa-base model augmented by OreoLM trained on NQ as entity predictor and directly test its transfer performance on EQ dataset without further fine-tuning.

To test whether OreoLM could recover missing relation, we mask **all** the edges corre-

Models	#param (B)	NQ	TQA
Graph-Retriever [Min+19]	0.11	34.7	55.8
REALM [Guu+20]	0.33 + 16	40.4	-
DPR [Kar+20] + BERT	0.56 + 16	41.5	56.8
+ **OreoLM** (DPR, T=2)	0.57 + 17	43.7	58.5
FiD (Base) = DPR + T5 (Base)	0.44 + 16	48.2	65.0
+ **OreoLM** (T5, T=2)	0.45 + 17	49.3	67.1
+ **OreoLM** (DPR & T5, T=2)	0.46 + 17	51.1	68.4
FiD (Large) = DPR + T5 (Large)	0.99 + 16	51.4	67.6
+ **OreoLM** (T5, T=2)	0.99 + 17	52.4	68.9
+ **OreoLM** (DPR & T5, T=2)	1.00 + 17	**53.2**	69.5
KG-FiD (Base) [Yu+22a]	0.44 + 16	49.6	66.7
KG-FiD (Large) [Yu+22a]	0.99 + 16	**53.2**	69.8
EMDR^2 [Sac+21a]	0.44 + 16	52.5	**71.4**

Table 3.4: **Open-Book QA** Evaluation.

sponding to each relation separately and make the prediction again. The average results before and after removing edges are shown on the left part of Figure 3.4. When we remove all the edges to each relation, OreoLM with $T = 1$ drops significantly, while $T = 2$ could still have good accuracy. To understand why OreoLM ($T = 2$) is less influenced, in the right part of Figure 3.4, we generate a reasoning path for each relation by averaging the predicted probability score at each reasoning step and pick the relation with the top score. For example, to predict the "Capital" of a country, the model learns to find the living place of the president, or the location of a country's central bank. Both are very reasonable guesses. Many previous works [XHW17] could also learn such rules in an ad-hoc manner and require costly searching or reinforcement learning. In contrast, OreoLM could learn such reasoning capacity for all relations end-to-end during pre-training.

Ablation Studies We conduct several ablation studies to evaluate which model design indeed contributes to the model. As shown in the bottom blocks in Table 3.3, we first remove the $\mathcal{KG}$ reasoning component and provide RoBERTa base model via concatenated KB triplets and train such a model using $\mathcal{L}_{SSM}$ over the same WikiDataset. Such a

model's results are close to the KEPLER results but much lower than other models with explicit knowledge memory. We further investigate the role of pre-training tasks. Without pre-training, the OreoLM only performs slightly better than RoBERTa baseline, due to the cold-start problem of entity and relation embedding. We further show that removing $\mathcal{L}_{ent}$ and $\mathcal{L}_{ent}$ could significantly influence final performance. The current combination is the best choice to train OreoLM to reason.

3.4.3 Evaluate for *Open-Book* QA

Though OreoLM is designed for *Closed-Book* QA, the learned model can serve as backbone for *Open-Book* QA. We take DPR and FiD models as baseline. For DPR retriever, we replace the question encoder to RoBERTa + OreoLM, fixing the passage embedding and only finetune on each downstream QA dataset. For FiD model, we replace the T5 + OreoLM. We also changed the retriever with our tuned DPR. Results in Table 3.4 show that by augmenting both retriever and generator, OreoLM improves a strong baseline like FiD, for about 3.1% for Base and 1.8% for Large, and it outperforms the very recent KG-FiD model for 1.6% in base setting, and achieve comparative performance in a large setting. Note that though our results is still lower than some recent models (e.g., EMDR^2), these methods are dedicated architecture or training framework for *Open-Book* QA. We may integrate OreoLM with these models to further improve their performance.

3.5 Summary

We presented OreoLM, a novel model that incorporates symbolic $\mathcal{KG}$ reasoning with existing LMs. We showed that OreoLM can bring significant performance gain to open-domain QA benchmarks, both for closed-book and open-book settings, as well as encoder-only and encoder-decoder models. Additionally, OreoLM produces reasoning paths that helps interpret the model prediction. In future, we'd like to improve OreoLM by training to conduct more reasoning steps, supporting locial reasoning, and apply OreoLM to a

broader range of knowledge-intensive NLP tasks.

Part II

Self-Supervised Learning from Symbolic Knowledge

CHAPTER 4

GPT-GNN: Generative Pre-Training of Graph Neural Networks

The second attempt was to utilize symbolic knowledge as a self-supervised pre-training signal. The goal of the pre-training is to empower GNNs to capture the intrinsic structural and semantic properties of the graph so that it can easily generalize to any downstream tasks on this graph with a few fine-tuning steps. To achieve this goal, we propose GPT-GNN, which models the graph distribution by directly learning to reconstruct the attributed graph. We factorize the likelihood of graph generation into two components: 1) attribute generation, and 2) edge generation. By modeling both components, GPT-GNN captures the inherent dependency between node attributes and graph structure during the generative process. We also propose an efficient large-scale GNN pre-training framework to optimize the generation loss, with which we only need to run GNN once without information leakage. Comprehensive experiments on the billion-scale academic graph and Amazon recommendation data demonstrate that GPT-GNN significantly outperforms state-of-the-art base GNN models without pre-training by up to 9.1% across different downstream tasks. In addition, the performance of GPT-GNN with only 10% data is comparable to direct supervised learning with 100% data. This shows the effectiveness of pre-training, especially when the label is scarce.

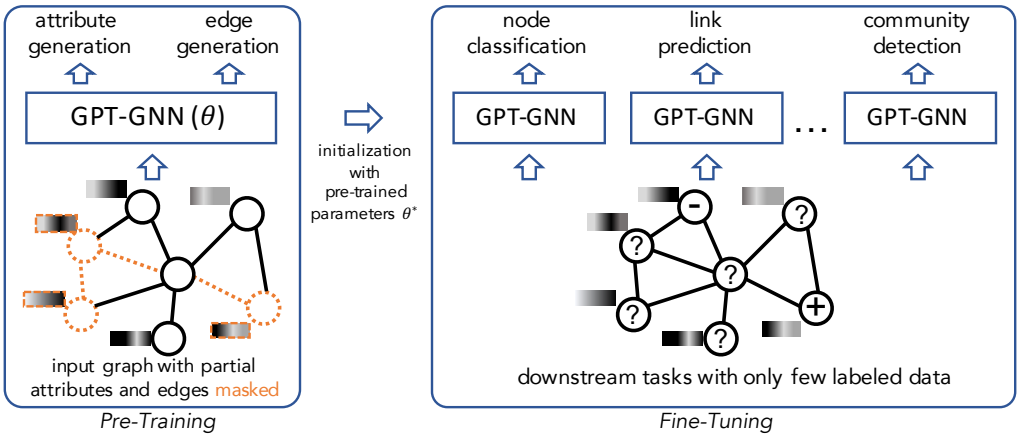


Figure 4.1: The pre-training and fine-tuning flow of GPT-GNN: First, a GNN is pre-trained with the self-supervised learning task—attribute and structure generations. Second, the pre-trained model and its parameters are then used to initialize models for downstream tasks on the input graph or graphs of the same domain.

4.1 Introduction

The breakthroughs in graph neural networks (GNNs) have revolutionized graph mining from structural feature engineering to representation learning [Bru+13; Gil+17b; KW17]. Commonly, GNNs take a graph with attributes as input and apply convolutional filters to generate node-level representations layer by layer. Often, a GNN model is trained with supervised information in an end-to-end manner for one task on the input graph. That said, for different tasks on the same graph, it is required to have enough and different sets of labeled data to train dedicated GNNs corresponding to each task. Usually, it is arduously expensive and sometimes infeasible to access sufficient labeled data for those tasks, particularly for large-scale graphs. Take, for example, the author disambiguation task in academic graphs [Tan+08], it has still faced the challenge of the lack of ground-truth to date.

Similar issues had also been experienced in natural language processing (NLP). Recent advances in NLP address them by training a model from a large unlabeled corpus and transferring the learned model to downstream tasks with only a few labels—the idea of pre-training. For example, the pre-trained BERT language model [Dev+19b] is able to

learn expressive contextualized word representations by reconstructing the input text—next sentence and masked language predictions, and thus it can significantly improve the performance of various downstream tasks. Additionally, similar observations have also been demonstrated in computer vision [OLV18; He+19; Che+20a].

Inspired by these developments, we propose to pre-train graph neural networks for graph mining. The goal of the pre-training is to empower GNNs to capture the structural and semantic properties of a input graph, so that it can easily generalize to any downstream tasks with a few fine-tuning steps on the graphs within the same domain. To achieve this goal, we propose to model the graph distribution by learning to reconstruct the input attributed graph.

To pre-train GNNs based on graph reconstruction, one straightforward option could be to directly adopt the neural graph generation techniques [KW16; You+18; Lia+19]. However, they are not suitable for pre-training GNNs by design. First, most of them focus only on generating graph structure without attributes, which does not capture the underlying patterns between node attributes and graph structure—the core of convolutional aggregation in GNNs. Second, they are designed to handle small graphs to date, limiting their potential to pre-train on large-scale graphs.

In this work, we design a self-supervised attributed graph generation task for GNN pre-training, with which both the structure and attributes of the graph are modeled. Based on this task, we present the GPT-GNN framework for generative pre-training of graph neural networks (Cf. Figure 4.1). The pre-trained GNN on the input graph can be then used as the initialization of models for different downstream tasks on the same type of graphs. Specifically, our contributions are illustrated below.

First, we design an attributed graph generation task to model both node attributes and graph structure. We decompose the graph generation objective into two components: Attribute Generation and Edge Generation, whose joint optimization is equivalent to maximizing the probability likelihood of the whole attributed graph. In doing this, the pre-trained model can capture the inherent dependency between node attributes and graph

structure.

Second, we propose an efficient framework GPT-GNN to conduct generative pre-training with the aforementioned task. GPT-GNN can calculate the attribute and edge generation losses of each node simultaneously, and thus only need to run the GNN once for the graph. Additionally, GPT-GNN can handle large-scale graphs with sub-graph sampling and mitigate the inaccurate loss brought by negative sampling with an adaptive embedding queue.

Finally, we pre-train GNNs on two large-scale graphs—the Open Academic Graph (OAG) of 179 million nodes & 2 billion edges and Amazon recommendation data of 113 million nodes. Extensive experiments show that the GPT-GNN pre-training framework can significantly benefit various downstream tasks. For example, by applying the pre-trained model on OAG, the node classification and link prediction performance is on average lifted by 9.1% over the state-of-the-art GNN models without pre-training. In addition, we show that GPT-GNN can consistently improve the performance of different base GNNs under various settings.

4.2 Preliminaries and Related Work

The goal of pre-training is to allow a model (usually neural networks) to initialize its parameters with pre-trained weights. In this way, the model can leverage the commonality between the pre-training and downstream tasks. Recently pre-training has shown superiority in boosting the performance of many downstream applications in computer vision and natural language processing. In the following, we first introduce the preliminaries about GNNs and then review pre-training approaches in graphs and other domains.

Preliminaries of Graph Neural Networks Recent years have witnessed the success of GNNs for modeling graph data [KW17; Vel+18; HYL17; Zin+c]. A GNN can be regarded as using the input graph structure as the computation graph for message passing [Gil+17b],

during which the local neighborhood information is aggregated to get a more contextual representation. Formally, suppose $H_t^{(l)}$ is the node representation of node t at the (l)-th GNN layer, the update procedure from the (l-1)-th layer to the (l)-th layer is:

$$H_t^{(l)} \leftarrow \underset{\forall s \in N(t), \forall e \in E(s,t)}{\textbf{Aggregate}} \left(\left\{ \textbf{Extract}\left(H_s^{(l-1)}; H_t^{(l-1)}, e\right) \right\} \right), \tag{4.1}$$

where $N(t)$ denotes all the source nodes of node t and $E(s,t)$ denotes all the edges from node s to t.

There are two basic operators for GNNs, which are **Extract(·)** and **Aggregate(·)**. Among them, **Extract(·)** represents the neighbor information extractor. It uses the target node's representation $H_t^{(l-1)}$ and the edge e between the two nodes as query, and extract useful information from source node $H_s^{(l-1)}$. **Aggregate(·)** serves as the aggregation function of the neighborhood information. The *mean, sum,* and *max* functions are often considered as the basic aggregation operators, while sophisticated pooling and normalization functions can also be designed. Under this framework, various GNN architectures have been proposed. For example, the graph convolutional network (GCN) proposed by Kipf *et al.* [KW17] averages the one-hop neighbor of each node in the graph, followed by a linear projection and then a non-linear activation. Hamilton *et al.* [HYL17] propose GraphSAGE that generalizes GCN's aggregation operation from *average* to *sum, max* and a *RNN unit*.

Also, there are a bunch of works incorporating the attention mechanism into GNNs. In general, the attention-based models implement the **Extract(·)** operation by estimating the importance of each source node, based on which a weighted aggregation is applied. For example, Velickovi *et al.* [Vel+18] propose the graph attention network (GAT), which adopts an additive mechanism to calculate attention and uses the same weight for calculating messages. Recently, Hu *et al.* propose the heterogeneous graph transformer (HGT) [Zin+c] that leverages multi-head attentions for different relation types to get type-dependent attentions. The proposed pre-training framework OreoLM can apply to all of these GNN models.

Pre-Training for Graphs Previous studies have proposed to utilize pre-training to learn node representations, which largely belong to two categories. The first category is usually termed as network/graph embedding, which directly parameterizes the node embedding vectors and optimizes them by preserving some similarity measures, such as the network proximity [Tan+15] or statistics derived from random walks [GL16; DCS17; Qiu+18]. However, the embeddings learned in this way cannot be used to initialize other models for fine-tuning over other tasks. In contrast, we consider a transfer learning setting, where the goal is to pre-train a generic GNN that can deal with different tasks.

With the increasing focus on GNNs, researchers have explored the direction of pre-training GNNs on unannotated data. Kipf *et al.* propose Variational Graph Auto-Encoders [KW16] to reconstruct the graph structure. Hamilton *et al.* propose GraphSAGE [HYL17], which can optimize via an unsupervised loss by using random walk based similarity metric. Velickovic *et al.* introduce Graph Infomax [Vel+19], which maximizes the mutual information between node representations obtained from GNNs and a pooled graph representation. Although these methods show enhancements over purely-supervised learning settings, the learning tasks can be achieved by forcing nearby nodes to have similar embeddings, ignoring the rich semantics and higher-order structure of the graph. Our work proposes to pre-train GNNs by the permutated generative objective, which is a harder graph task and thus can guide the model to learn more complex semantics and structure of the input graph.

In addition, there are attempts to pre-train GNNs to extract graph-level representations. Sun *et al.* present InfoGraph [Sun+20], which maximizes the mutual information between graph-level representations obtained from GNNs and the representations of sub-structures. Hu *et al.* [Hu+20] introduce different strategies to pre-train GNNs at both node and graph levels and show that combining them together can improve the performance on graph classification tasks. Our work is different with them as our goal is to pre-train GNNs over a single (large-scale) graph and conduct the node-level transfer.

Pre-Training for Vision and Language Pre-training has been widely used in computer vision (CV) and natural language processing (NLP). In CV, early pre-training techniques [Gir+14; Don+14b; Pat+16] mostly follow the paradigm of first pre-training a model on large-scale supervised datasets (such as ImageNet [Den+09]) and then fine-tuning the pre-trained model on downstream tasks [Gir+14] or directly extracting the representations as features [Don+14b]. Recently, some self-supervised tasks [OLV18; He+19; Che+20a] have also been utilized to pre-train vision models. In NLP, Early works have been focused on learning (shallow) word embeddings [Mik+13a; PSM14] by leveraging the co-occurrence statistics on the text corpus. More recently, significant progresses have been made on contextualized word embeddings, such as BERT [Dev+19b], XLNET [Yan+19b] and GPT [Rad+19]. Take BERT as an example, it pre-trains a text encoder with two self-supervised tasks in order to better encode words and their contexts. These pre-training approaches have been shown to yield state-of-the-art performance in a wide range of NLP tasks and thus used as a fundamental component in many NLP systems.

4.3 Methodology

In this section, we formalize the attributed graph generation task and introduce the generative pre-training framework (GPT-GNN).

4.3.1 The GNN Pre-Training Problem

The input to GNNs is usually an attributed graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, where $\mathcal{V}$ and $\mathcal{E}$ denote its node and edge sets, and $\mathcal{X}$ represents the node feature matrix. A GNN model learns to output node representations under the supervision of a specific downstream task, such as node classification. Sometimes there exist multiple tasks on a single graph, and most GNNs require sufficient dedicated labeled data for each task. However, it is often challenging to obtain sufficient annotations, in particular for large-scale graphs, hindering the training of a well-generalized GNN. Therefore it is desirable to have a pre-trained GNN model that can

generalize with few labels. Conceptually, this model should 1) capture the intrinsic structure and attribute patterns underlying the graph and 2) thus benefit various downstream tasks on this graph.

Formally, our goal of GNN pre-training concerns the learning of a general GNN model f_θ purely based on single (large-scale) graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ without labeled data such that f_θ is a good initialization for various (unseen) downstream tasks on the same graph or graphs of the same domain. To learn such a general GNN model without labeled data on the graph, a natural question arises here is: *how to design an unsupervised learning task over the graph for pre-training the GNN model?*

4.3.2 The Generative Pre-Training Framework

Recent advances in self-supervised learning for NLP [Dev+19b; Yan+19b] and CV [OLV18; He+19; Che+20a] have shown that unlabeled data itself contains rich semantic knowledge, and thus a model that can capture the data distribution is able to transfer onto various downstream tasks. Inspired by this, we propose GPT-GNN, which pre-trains a GNN by reconstructing/generating the input graph's structure and attributes.

Formally, given an input graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ and a GNN model f_θ, we model the likelihood over this graph by this GNN as $p(G; \theta)$—representing how the nodes in G are attributed and connected. GPT-GNN aims to pre-train the GNN model by maximizing the graph likelihood, i.e., $\theta^* = \max_\theta p(G; \theta)$.

Then, the first question becomes how to properly model $p(G; \theta)$. Note that most existing graph generation methods [You+18; Lia+19] follow the auto-regressive manner to factorize the probability objective, i.e., the nodes in the graph come in an order, and the edges are generated by connecting each new arriving node to existing nodes. Similarly, we denote a permutation vector π to determine the node ordering, where i^π denotes the node id of i-th position in permutation π. Consequently, the graph distribution $p(G; \theta)$ is equivalent to

the expected likelihood over all possible permutations, i.e.,

$$p(G;\theta) = \mathbb{E}_{\pi}\big[p_{\theta}(X^{\pi}, E^{\pi})\big],$$

where $X^{\pi} \in \mathbb{R}^{|\mathcal{V}|\times d}$ denotes permutated node attributes and E is a set of edges, while E_i^{π} denotes all edges connected with node i^{π}. For simplicity, we assume that observing any node ordering π has an equal probability and also omit the subscript π when illustrating the generative process for one permutation in the following sections. Given a permutated order, we can factorize the log likelihood autoregressively—generating one node per iteration—as:

$$\log p_{\theta}(X, E) = \sum_{i=1}^{|\mathcal{V}|} \log p_{\theta}(X_i, E_i \mid X_{<i}, E_{<i}). \tag{4.2}$$

At each step i, we use all nodes that are generated before i, their attributes $X_{<i}$, and the structure (edges) between these nodes $E_{<i}$ to generate a new node i, including both its attribute X_i and its connections with existing nodes E_i.

Essentially, the objective in Eq. 4.2 describes the autoregressive generative process of an attributed graph. The question becomes: *how to model the conditional probability* $p_{\theta}(X_i, E_i|X_{<i}, E_{<i})$*?*

4.3.3 Factorizing Attributed Graph Generation

To compute $p_{\theta}(X_i, E_i|X_{<i}, E_{<i})$, while capturing the dependency between the attributes and structure for each node, we define a variable o to denote the index vector of all the observed edges within E_i. Thus, $E_{i,o}$ denotes the observed edges. Similarly, $\neg o$ denotes the index of all the masked edges, which are to be generated. With this, we can rewrite the conditional probability as an expected likelihood over all observed edges:

$$
\begin{aligned}
& p_\theta(X_i, E_i \mid X_{<i}, E_{<i}) \\
=& \sum\nolimits_o p_\theta(X_i, E_{i,\neg o} \mid E_{i,o}, X_{<i}, E_{<i}) \cdot p_\theta(E_{i,o} \mid X_{<i}, E_{<i}) \\
=& \mathbb{E}_o\Big[p_\theta(X_i, E_{i,\neg o} \mid E_{i,o}, X_{<i}, E_{<i})\Big] \\
=& \mathbb{E}_o\Big[\underbrace{p_\theta(X_i \mid E_{i,o}, X_{<i}, E_{<i})}_{\text{1) generate attributes}} \cdot \underbrace{p_\theta(E_{i,\neg o} \mid E_{i,o}, X_{\leq i}, E_{<i})}_{\text{2) generate edges}}\Big].
\end{aligned} \tag{4.3}
$$

This factorization design is able to model the dependency between node i's attributes X_i and its associated connections E_i. The first term $p_\theta(X_i \mid E_{i,o}, X_{<i}, E_{<i})$ denotes the generation of attributes for node i. Based on the observed edges $E_{i,o}$, we gather the target node i's neighborhood information to generate its attributes X_i. The second term $p_\theta(E_{i,\neg o} \mid E_{i,o}, X_{\leq i}, E_{<i})$ denotes the generation of masked edges. Based on both the observed edges $E_{i,o}$ and the generated attributes X_i, we generate the representation of the target node i and predict whether each edge within $E_{i,\neg o}$ exists.

So far, we factorize the attributed graph generation process into a node attribute generation step and an edge generation step. The question we need to answer here is: *How to efficiently pre-train GNNs by optimizing both attribute and edge generation tasks?*

4.3.4 Efficient Attribute and Edge Generation

For the sake of efficiency, it is desired to compute the loss of attribute and edge generations by running the GNN only once for the input graph. In addition, we expect to conduct attribute generation and edge generation simultaneously. However, edge generation requires node attributes as input, which can be leaked to attribute generation. To avoid information leakage, we design to separate each node into two types:

1. Attribute Generation Nodes. We mask out the attributes of these nodes by replacing their attributes with a dummy token and learn a shared vector X^{init} to represent

it[1]. This is equivalent to the trick of using the [Mask] token in the masked language model [Dev+19b].

2. Edge Generation Nodes. For these nodes, we keep their attributes and put them as input to the GNN.

We then input the modified graph to the GNN model and generate the output representations. We use h^{Attr} and h^{Edge} to represent the output embeddings of Attribute Generation and Edge Generation Nodes, respectively. As the attributes of Attribute Generation Nodes are masked out, h^{Attr} in general contains less information than h^{Edge}. Therefore, when conduct the GNN message passing, we only use Edge Generation Nodes' output h^{Edge} as outward messages. The representations of the two sets of nodes are then used to generate attributes and edges with different decoders.

For Attribute Generation, we denote its decoder as $Dec^{Attr}(\cdot)$, which takes h^{Attr} as input and generates the masked attributes. We define a distance function as a metric between the generated attributes and the real ones, such as perplexity for text or L2-distance for vectors. Thus, we calculate the attribute generation loss via:

$$\mathcal{L}_i^{Attr} = Distance\big(Dec^{Attr}(h_i^{Attr}), X_i\big). \tag{4.4}$$

By minimizing the distance between the generated and masked attributes, it is equivalent to maximize the likelihood to observe each node attribute, i.e., $p_\theta(X_i \mid E_{i,o}, X_{<i}, E_{<i})$.

For Edge Generation, we assume that the generation of each edge is independent with others, so that we can factorize the likelihood:

$$p_\theta(E_{i,\neg o} \mid E_{i,o}, X_{\leq i}, E_{<i}) = \prod_{j^+ \in E_{i,\neg o}} p_\theta(j^+ \mid E_{i,o}, X_{\leq i}, E_{<i}). \tag{4.5}$$

Next, after getting the Edge Generation node representation h^{Edge}, we model the

[1] X^{init} has the same dimension as X_i and can be learned during pre-training.

likelihood that node i is connected with node j by $Dec^{Edge}(h_i^{Edge}, h_j^{Edge})$, where Dec^{Edge} is a pairwise score function. Finally, we adopt the negative contrastive estimation to calculate the likelihood for each linked node j^+. We prepare all the unconnected nodes as S_i^- and calculate the contrastive loss via

$$\mathcal{L}_i^{Edge} = -\sum_{j^+ \in E_{i,\neg o}} \log \frac{\exp\left(Dec^{Edge}(h_i^{Edge},\ h_{j^+}^{Edge})\right)}{\sum_{j \in S_i^- \cup \{j^+\}} \exp\left(Dec^{Edge}(h_i^{Edge},\ h_j^{Edge})\right)} \tag{4.6}$$

By optimizing $\mathcal{L}^{Edge}$, it is equivalent to maximizing the likelihood of generating all the edges, and thus the pre-trained model is able to capture the intrinsic structure of the graph.

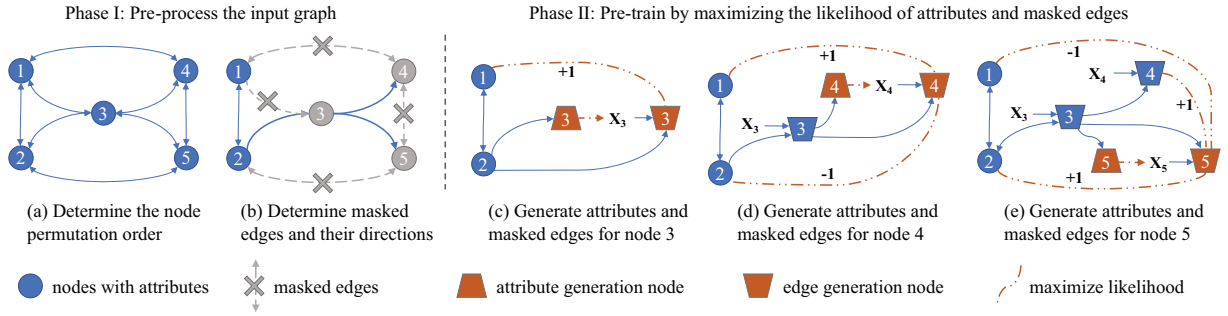


Figure 4.2: An illustrative example of the proposed attributed graph generation procedure.

Figure 5.2 illustrates the attributed graph generation process. Specifically: (a) We determine the node permutation order π for the input graph. (b) We randomly select a portion of the target node's edges as observed edges $E_{i,o}$ and the remaining as masked edges $E_{i,\neg o}$ (grey dashed lines with cross). We delete masked edges in the graph. (c) We separate each node into the Attribute Generation and Edge Generation nodes to avoid information leakage. (d) After the pre-processing, we use the modified adjacency matrix to calculate the representations of node 3,4 and 5, including both their Attribute and Edge Generation Nodes. Finally, as illustrated in (d)–(e), we train the GNN model via the attribute prediction and masked edge prediction task for each node in parallel.

4.4 Evaluation

To evaluate the performance of GPT-GNN, we conduct experiments on the Open Academic Graph (OAG) and Amazon Recommendation datasets. To evaluate the generalizability of GPT-GNN, we consider different transfer settings—time transfer and field transfer—which are of practical importance.

4.4.1 Experimental Setup

We conduct experiments on both heterogeneous and homogeneous graphs. For heterogeneous graphs, we use the Open Academic Graph and Amazon Review Recommendation data. For homogeneous graphs, we use the Reddit dataset [HYL17] and the paper citation network extracted from OAG. All datasets are publicly available and the details can be found in Appendix A.

Open Academic Graph (OAG) [Wan+20; Zha+19b; Tan+08] contains more than 178 million nodes and 2.236 billion edges. It is the largest publicly available heterogeneous academic dataset to date. Each paper is labeled with a set of research topics/fields (e.g., Physics and Medicine) and the publication date ranges from 1900 to 2019. We consider the prediction of Paper–Field, Paper–Venue, and Author Name Disambiguation (Author ND) as three downstream tasks [Zin+c; Don+20]. The performance is evaluated by MRR—a widely adopted ranking metric [Liu11].

Amazon Review Recommendation Dataset (Amazon) [NLM19] contains 82.8 million reviews, 20.9 million users, and 9.3 million products. The reviews are published from 1996 to 2018. Each review consists of a discrete rating score from 1 to 5 and a specific field, including book, fashion, etc. For downstream tasks, we predict the rating score as a five-class classification task within the Fashion, Beauty, and Luxury fields. We use micro F1-score as the evaluation metric.

On the OAG and Amazon datasets, we use the state-of-the-art heterogeneous GNN—Heterogeneous Graph Transformer (HGT) [Zin+c]—as the base model for GPT-GNN.

Furthermore, we also use other (heterogeneous) GNNs as the base model to test our generative pre-training framework. For all base models, we set the hidden dimension as 400, the head number as 8, and the number of GNN layers as 3. All of them are implemented using the PyTorch Geometric (PyG) package [FL19]. We optimize the model via the AdamW optimizer [LH19] with the Cosine Annealing Learning Rate Scheduler [LH17] with 500 epochs and select the one with the lowest validation loss as the pre-trained model. We set the adaptive queue size to be 256. During downstream evaluation, we fine-tune the model using the same optimization setting for 200 epochs as that in pre-training. We train the model on the downstream tasks for five times and report the mean and standard deviation of test performance.

There exist several works that propose unsupervised objectives over graphs, which can potentially be used to pre-train GNNs. We thus compare the proposed GPT-GNN framework with these baselines:

1. *GAE* [KW16], which denotes graph auto-encoders, focuses on a traditional link prediction task. It randomly masks out a fixed proportion of the edges and asks the model to reconstruct these masked edges.

2. *GraphSAGE (unsp.)* [HYL17] forces connected nodes to have similar output node embeddings. Its main difference with GAE lies in that it does not mask out the edges during pre-training.

3. *Graph Infomax* [Vel+19] tries to maximize the local node embeddings with global graph summary embeddings. Following its setting for a large-scale graph, for each sampled subgraph, we shuffle the graph to construct negative samples.

In addition, we also evaluate the two pre-training tasks in GPT-GNN by using each one of them alone, that is, attribute generation: *GPT-GNN (Attr)*; and edge generation: *GPT-GNN (Edge)*.

	Downstream Dataset	OAG			Amazon		
	Evaluation Task	Paper–Field	Paper–Venue	Author ND	Fashion	Beauty	Luxury
	No Pre-train	.336±.149	.365±.122	.794±.105	.586±.074	.546±.071	.494±.067
Field Transfer	GAE	.403±.114	.418±.093	.816±.084	.610±.070	.568±.066	.516±.071
	GraphSAGE (unsp.)	.368±.125	.401±.096	.803±.092	.597±.065	.554±.061	.509±.052
	Graph Infomax	.387±.112	.404±.097	.810±.084	.604±.063	.561±.063	.506±.074
	Contrastive	.381±.104	.398±.102	.804±.081	.598±.059	.567±.058	.504±.067
	GPT-GNN (Attr)	.396±.118	.423±.105	.818±.086	.621±.053	.576±.056	.528±.061
	GPT-GNN (Edge)	.401±.109	.428±.096	.826±.093	.616±.060	.570±.059	.520±.047
	GPT-GNN	**.407±.107**	**.432±.098**	**.831±.102**	**.625±.055**	**.577±.054**	**.531±.043**
Time Transfer	GAE	.384±.117	.412±.101	.812±.095	.603±.065	.562±.063	.510±.071
	GraphSAGE (unsp.)	.352±.121	.394±.105	.799±.093	.594±.067	.553±.069	.501±.064
	Graph Infomax	.369±.116	.398±.102	.805±.089	.599±.063	.558±.060	.503±.063
	Contrastive	.363±.101	.391±.107	.801±.086	.595±.054	.554±.063	.502±.062
	GPT-GNN (Attr)	.382±.114	.414±.098	.811±.089	.614±.057	**.573±.053**	.522±.051
	GPT-GNN (Edge)	.392±.105	.421±.102	.821±.088	.608±.055	.567±.038	.513±.058
	GPT-GNN	**.400±.108**	**.429±.101**	**.825±.093**	**.617±.059**	.572±.059	**.525±.057**
Combined Transfer	GAE	.371±.124	.403±.108	.806±.102	.596±.065	.554±.063	.505±.061
	GraphSAGE (unsp.)	.349±.130	.393±.118	.797±.097	.589±.071	.545±.068	.498±.064
	Graph Infomax	.360±.121	.391±.102	.800±.093	.591±.068	.550±.058	.501±.063
	Contrastive	.357±.109	.389±.104	.797±.089	.588±.058	.547±.061	.500±.064
	GPT-GNN (Attr)	.364±.115	.409±.103	.809±.094	.608±.062	.569±.057	.517±.057
	GPT-GNN (Edge)	.386±.116	.414±.104	.815±.105	.604±.058	.565±.057	.514±.047
	GPT-GNN	**.393±.112**	**.420±.108**	**.818±.102**	**.610±.054**	**.572±.063**	**.521±.049**

Table 4.1: Performance of different downstream tasks on OAG and Amazon by using different pre-training frameworks with the heterogeneous graph transformer (HGT) [Zin+c] as the base model. 10% of labeled data is used for fine-tuning. We report the results under different transfer settings with 10% fine-tuning data. Our proposed Generative Pre-training framework can enhance the downstream evaluation performance for 9.1% and 5.7% to OAG and Amazon respectively, and it can consistently outperform all the other baselines under different settings.

4.4.2 Pre-Training and Fine-Tuning Setup

The goal of pre-training is to transfer knowledge learned from numerous unlabeled nodes of a large graph to facilitate the downstream tasks with a few labels. Specifically, we first pre-train a GNN and use the pre-trained model weights to initialize models for downstream tasks. We then fine-tune the models with the downstream task specific decoder on the training (fine-tuning) set and evaluate the performance on the test set.

Broadly, there are two different setups. The first one is to pre-train and fine-tune on exactly the same graph. The second one is relatively more practical, which is to pre-train on one graph and fine-tune on unseen graphs of the same type as the pre-training one. Specifically, we consider the following three graph transfer settings between the pre-training

and fine-tuning stages:

1. *Time Transfer*, where we use data from different time spans for pre-training and fine-tuning. For both OAG and Amazon, we use data before 2014 for pre-training and data since 2014 for fine-tuning.

2. *Field Transfer*, where we use data from different fields for pre-training and evaluating. In OAG, we use papers in the field of computer science (CS) for downstream fine-tuning and use all papers in the remaining fields (e.g., Medicine) for pre-training. In Amazon, we pre-train on products in Arts, Crafts, and Sewing, and fine-tune on products in Fashion, Beauty, and Luxury.

3. *Time + Field Transfer*, where we use the graph of particular fields before 2014 to pre-train the model and use the data from other fields since 2014 for fine-tuning. Intuitively, this combined transfer setting is more challenging than the transfer of time or field alone.

During fine-tuning, for both datasets, we choose nodes from 2014 to 2016 for training, 2017 for validation, and since 2018 for testing. To meet the assumption that training data is usually scarce, we only provide 10% of the labels for training (fine-tuning) by default, while the ablation study over different data percentages is also conducted. During pre-training, we randomly select a subset of the data as the validation set.

4.4.3 Experimental Results

We summarize the performance of downstream tasks with different pre-training methods on OAG and Amazon in Table 4.1. As discussed above, we setup three different transfer settings between pre-training and fine-tuning stages: Field Transfer, Time Transfer, and Field + Time Combined Transfer, as organized in three different blocks in the Table.

Overall, the proposed GPT-GNN framework significantly enhances the performance for all downstream tasks on both datasets. On average, GPT-GNN achieves relative

performance gains of 13.3% and 5.7% over the base model without pre-training on OAG and Amazon, respectively. Moreover, it consistently outperforms other pre-training frameworks, such as Graph Infomax, across different downstream tasks for all three transfer settings on both datasets.

Different transfer settings. Observed from Table 4.1, the performance gain lifted by pre-training under the field transfer is higher than that under the time transfer, and the time + field combined transfer is the most challenging setting as evident in the least performance gain brought by pre-training. Nonetheless, under the combined transfer, GPT-GNN still achieves 11.7% and 4.6% performance gains on both datasets, respectively. Altogether, the results suggest that *the proposed generative pre-training strategy enables the GNN model to capture the generic structural and semantic knowledge of the input graph, which can be used to fine-tune on the unseen part of the graph data.*

We analyze the effectiveness of the two pre-training tasks in GPT-GNN—attribute generation and edge generation—by examining which of them is more beneficial for the pre-training framework and, by extension, downstream tasks. In Table 4.1, we report the performance of GPT-GNN by using attribute generation and edge generation alone, that is, GPT-GNN (Attr) and GPT-GNN (Edge). On OAG, the average performance gains by GPT-GNN (Attr) and GPT-GNN (Edge) are 7.4% and 10.3%, respectively, suggesting that Edge Generation is a more informative pre-training task than Attribute Generation in GPT-GNN. However, we have an opposite observation for Amazon, on which the performance improved by Attribute Generation is 5.2% in contrast to the 4.1% improvement lifted by Edge Generation. *This suggests that the GPT-GNN framework benefits differently from attribute and edge generations on different datasets. However, combining the two pre-training tasks together produces the best performance on both cases.*

We further compare the Edge Generation task against other edge-based pre-training methods—GAE and GraphSage (unsp.)—in Table 4.1. On OAG, the performance improvements brought by GPT-GNN's edge generation, GAE, and GraphSage over no pre-training

Model	HGT	GCN	GAT	RGCN	HAN
No Pre-train	**.336**	.317	.308	.296	.322
GPT-GNN	**.407**	.349	.362	.351	.384
Relative Gain	21.1%	10.1%	17.5%	18.6%	19.3%

Table 4.2: Compare the pre-training Gain with different GNN architectures. Evaluate on OAG, Paper-Field Task, under Combined Transfer setting with 10% training data.

are 10.3%, 7.4%, and 4.0%, respectively. On Amazon, the gains are 5.2%, 3.1%, and 1.3%, respectively. From the comparisons, we have the following observations. First, both GAE and GPT-GNN's edge generation offer better results than GraphSage on both datasets, demonstrating that masking on edges is an effective strategy for self-supervised graph representation learning. Without edge masking, the model simply retains a similar embedding for connected nodes, as the label we would like to predict (whether two nodes are linked) has already been encoded in the input graph structure. Such information leakage will downgrade the edge prediction task to a trivial problem. Second, the proposed Edge Generation task consistently outperforms GAE. The main advantage of GPT-GNN's edge generation comes from that it learns to generate missing edges autoregressively and thus can capture the dependencies between the masked edges, which are discarded by GAE. *In summary, the results suggest that the proposed graph generation tasks can give informative self-supervision for GNN pre-training.*

Ablation studies on the base GNN. We investigate whether the other GNN architectures can benefit from the proposed pre-training framework. Therefore, in addition to HGT [Zin+c], we try GCN [KW17], GAT [Vel+18], RGCN [Sch+18], and HAN [Wan+19b] as the base model. Specifically, we pre-train them on OAG and then use the paper-field prediction task under the combined transfer setting with 10% of training data for fine-tuning and testing. Model-independent hyper-parameters, such as the hidden dimension size and optimization, are set the same. The results are reported in in Table 4.2. We can observe that the proposed GPT-GNN pre-training framework can enhance the downstream performance for all the GNN architectures.

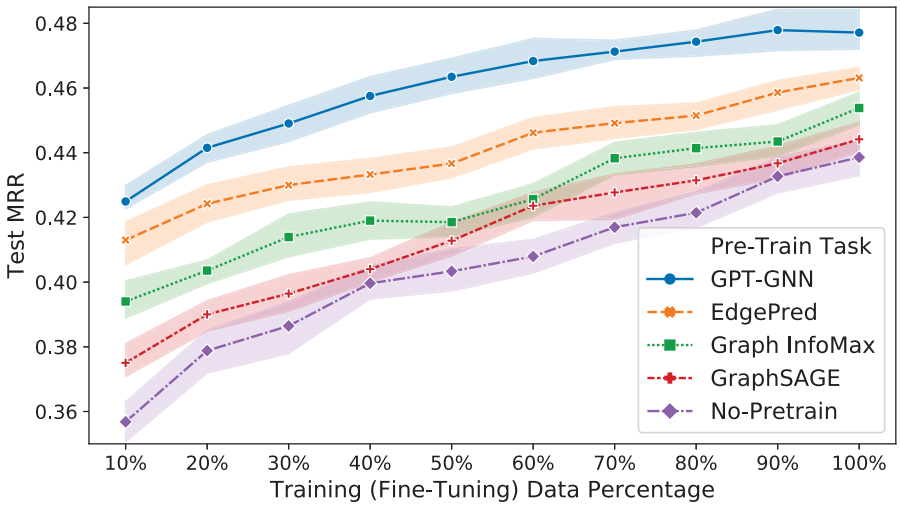


Figure 4.3: Compare pre-training tasks with different training data size. Evaluated by the paper–field prediction task on OAG under the field transfer setting.

Training data size. In Figure 4.3, we examine whether the proposed GPT-GNN method can generalize well with different training data size during fine-tuning, i.e., from 10% to 100%. First, we can observe that GPT-GNN and other pre-training frameworks consistently improve the downstream task performance with more labeled training data. Second, it is clear that GPT-GNN performs the best among all pre-training tasks/frameworks. Finally, we can see that with the pre-trained model, fine-tuning with only 10–20% of data (the two leftmost blue circles) generates comparative performance to the supervised learning with all 100% of training data (the rightmost purple diamond), demonstrating the superiority of GNN pre-training, especially when the label is scarce.

4.5 Summary

In this work, we study the problem of graph neural network pre-training. We present GPT-GNN—a generative GNN pre-training framework. We design the graph generation factorization to guide the base GNN model to autoregressively reconstruct both the attributes and structure of the input graph. Furthermore, we propose to separate the attribute and edge generation nodes to avoid information leakage. In addition, we introduce the adaptive

node representation queue to mitigate the gap between the likelihoods over the sampled graph and the full graph. The pre-trained GNNs with fine-tuning over few labeled data can achieve significant performance gains on various downstream tasks across different datasets with different transfer settings.

CHAPTER 5

REVEAL: Retrieval-Augmented Visual Language Pre-Training

In this paper, we propose an end-to-end Retrieval-Augmented Visual Language Model (ReVeaL) that learns to encode world knowledge into a large-scale memory, and to retrieve from it to answer knowledge-intensive queries. ReVeaL consists of four key components: the memory, the encoder, the retriever and the generator. The large-scale memory encodes various sources of multimodal world knowledge (e. ., image-text pairs, question answering pairs, knowledge graph triplets, etc) via a unified encoder. The retriever finds the most relevant knowledge entries in the memory, and the generator fuses the retrieved knowledge with the input query to produce the output. A key novelty in our approach is that the memory, encoder, retriever and generator are all pre-trained end-to-end on a massive amount of data. Furthermore, our approach can use a diverse set of multimodal knowledge sources, which is shown to result in significant gains. We show that ReVeaL achieves state-of-the-art results on visual question answering and image captioning.

Figure 5.1: We augment a visual-language model with the ability to retrieve multiple knowledge entries from a diverse set of knowledge sources, which helps generation. Both retriever and generator are trained jointly, end-to-end, by optimizing a language modeling objective.

5.1 Introduction

Recent large-scale models such as T5 [Raf+20], GPT-3 [Bro+20], PaLM [Cho+22], CoCa [Yu+22b], Flamingo [Ala+22], BEIT-3 [Wan+22a] and PaLI [Che+22b] have demonstrated the ability to store substantial amounts of world knowledge, when scaled to tens of billions of parameters and trained on vast text and image corpora. These models achieve state-of-the-art results in downstream tasks such as image captioning, visual question answering and open vocabulary recognition. Yet, these models have a number of drawbacks: (i) they require massive scale, of parameters, data and computation, and (ii) they need to be re-trained every time the world knowledge is updated.

To address these issues, we adopt a different approach. Instead of statically compiling world knowledge into model weights, we transform the knowledge into a key-value memory through neural representation learning. Our model learns to utilize the memory for answering knowledge-intensive queries. By decoupling the knowledge memorization from reasoning, we enable our model to leverage various external sources of knowledge (e.g., Wikipedia passages and images [Sri+21], the WikiData knowledge graph [VK14], Web image-text pairs [Cha+21] and visual question answering data [Goy+17]). This enables the model parameters to focus on understanding the query and conducting reasoning, rather than being dedicated to memorization.

Retrieval-augmented models have attracted a fair amount of attention in the fields of NLP [Guu+20; Iza+22] and computer vision [Lon+22; Gui+21]. Typically, these models often use a pre-existing single-modality backbone to encode and retrieve information from the knowledge corpus. Such approaches do not leverage all available modalities in the query and knowledge corpora, and hence they might not find the information that is most helpful for generating the model output. A key novelty in our approach is that we encode and store various sources of multimodal world knowledge into a unified memory, which the retriever can access via multimodal query encodings, to find the most relevant information from across complementary sources. Our multimodal memory and retriever are pre-trained end-to-end together together with the rest of the model, on a massive amount of data and using diverse knowledge sources.

A key challenge of pre-training the multimodal retriever end-to-end is the lack of direct supervision. There is no ground-truth indicating which knowledge entries are most helpful for answering knowledge-intensive queries. Some of the existing works in NLP [Guu+20; Lew+20; Sac+21b] propose to acquire training signal by assessing the usefulness of each retrieved knowledge entry independently for helping language modelling. This approach is inefficient, as it involves estimating hundreds of retrieved knowledge entries independently, and also inaccurate as it discards the dependency between different knowledge entries in the retrieval set. In contrast, we propose to get this training signal while simultaneously considering multiple retrieved knowledge entries, by introducing an attentive fusion layer that injects retrieval score into the attention calculation procedure. This enables the retrieval module to be differentiable and jointly pre-trained with the rest of the model.

In summary, our key contributions are as follows:

1. We are the first to propose an end-to-end pre-training paradigm that learns to index into a large-scale memory to solve knowledge-intensive visual-language tasks.

2. Our method can construct a large-scale memory by encoding various sources of multimodal world knowledge, including Wikipedia passage, web images with alt-text

captions, and knowledge graph triplets.

3. REVEAL achieves state-of-the-art performance on several knowledge-intensive visual question answering and image captioning datasets. Notably on the OKVQA benchmark, REVEAL achieves a new state-of-the-art, 59.1% accuracy, while using order of magnitude fewer parameters than previous works.

5.2 Related Work and Background

Knowledge-based Visual Question Answering. To evaluate a model's ability to comprehend multimodal world knowledge not easily inferred from input data, several knowledge-based Visual Question Answering (VQA) datasets have been introduced. KB-VQA [Wan+17] and FVQA [Wan+18] design questions that can be answered by retrieving relevant triplets from domain-specific structured knowledge graphs. OK-VQA [Mar+19] improves these datasets by necessitating the use of external knowledge, which goes beyond what can be directly observed in the input images. More recently, A-OKVQA [Sch+22] offers further improvements to OK-VQA by exclusively selecting questions that demand both external knowledge and commonsense reasoning about the image scenes. To tackle knowledge-based VQA tasks, many approaches have been proposed to incorporate external knowledge into visual-language models. One line of research uses explicit knowledge from structured knowledge graphs [NLS18; Gar+20; Wan+22b; Hu+22a] or unstructured text corpora [Mar+21; Luo+21; Wu+22]. The key component for these works is the knowledge retriever. Some works [NLS18; Gar+20; Mar+21; Wu+22] utilize off-the-shelf vision detection models to generate image tags for knowledge retrieval, while others train the retrieval model via distant supervision [Luo+21] or auxiliary tasks (e.g. entity linking) [Gui+21]. Another research direction aims to incorporate implicit knowledge from pre-trained Large Language Models, such as GPT-3 [Bro+20] or PaLM [Cho+22]. These approaches utilize off-the-shelf image caption models to convert images into text, feed them into a language model, and use the generated text output as augmented knowledge [Yan+21;

Gui+21; Lin+22]. Our work follows the first direction, augmenting a vision-language model with an explicit knowledge retriever. The main distinction is that we propose an end-to-end training framework to jointly learn the answer generator and retriever, rather than using a fixed or predefined knowledge retrieval.

End-to-End Training of Retrieval-Augmented Models. Given the advantage of knowledge retrieval, a key question is how to get learning signal to train the retrieval model. For tasks with annotated retrieval ground-truth, retrieval training can be conducted via standard contrastive learning [Kar+20]. However, most tasks do not provide clear indications of which knowledge entries are relevant for generating answers. To this end, a series of studies have investigated retrieval training using supervision derived from downstream tasks. REALM [Guu+20] trains a single-document retriever by concatenating each retrieved result with the query, to calculate the final loss independently. A similar approach has been used by EMDR^2 [Sac+21b] for multi-document retrieval training. FID-KD [IG21] proposes to use the aggregated attention score calculated by the generator as a distillation signal to train the retriever. Atlas [Iza+22] further introduces a perplexity distillation loss and a leave-one-out variant. Our REVEAL proposes to inject the retrieval scores directly into an attentive fusion module, enabling to train the retriever to directly optimize downstream tasks as well as pre-training objectives.

5.3 Method

We propose a Retrieval-Augmented Visual Language Model (REVEAL), which learns to use knowledge from different sources for solving knowledge-intensive tasks. For both pre-training and fine-tuning, our goal is to learn the distribution $P(y \mid x)$ to generate a textual output y conditioned on a multimodal input query x. REVEAL contains four components: knowledge encoding, memory, retrieval and generation. Given an input query x, we first retrieve K possibly helpful entries $M = \{m_1, \cdots, m_K\}$ from the memory corpora $\mathcal{M}$. Each m is a memory entry containing the encoded single key embedding and a sequence

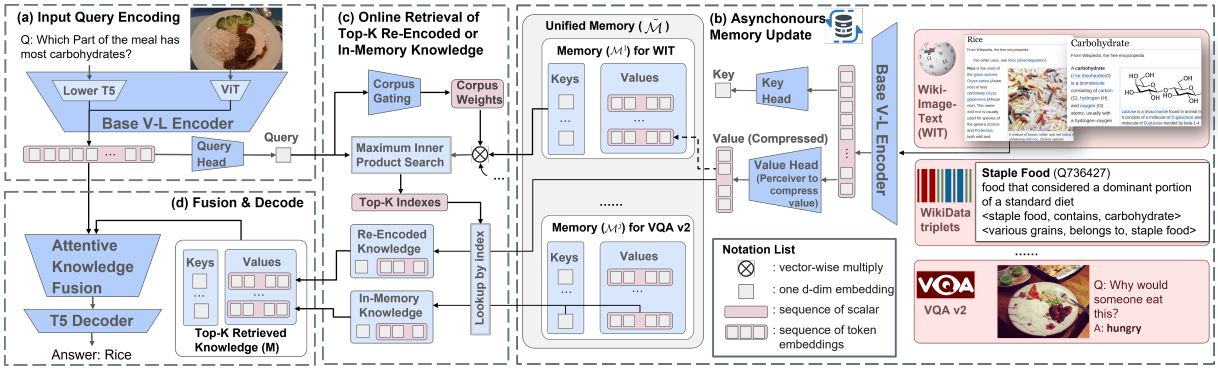


Figure 5.2: **The overall workflow of ReVeaL** consists of four main steps: **(a)** encode a multimodal input into a sequence of token embeddings and a summarized query embedding; **(b)** encode each knowledge entry from different corpus into unified key and value embedding pairs, where key is used to index the memory and value contains full information of the knowledge; **(c)** retrieve top-K most similar knowledge items from different knowledge sources, and return the pre-computed in-memory value embeddings and re-encoded value; and **(d)** fuse the top-K knowledge items via attentive knowledge fusion layer by injecting the retrieval score as a prior during attention calculation. This facilitates ReVeaL's key novelty: the memory, encoder, retriever and the generator can be jointly trained in an end-to-end manner.

of value embeddings (we will describe how to encode knowledge items into memory entries in Sec. 5.3.2). With it, the retriever can use embedding similarity to find relevant memory entries. We model this retrieval process as sampling from distribution $p(M \mid x)$. Then, we condition on both the retrieved set M and the original input query x to generate the output y, modeled as $p(y \mid x, M)$. To obtain the overall likelihood of generating y, we treat M as a latent variable from the entire memory $\tilde{\mathcal{M}}$ and marginalize over it yielding:

$$p(y \mid x) = \sum_{M \subset \tilde{\mathcal{M}}} \underbrace{p(M \mid x)}_{retrieval} \cdot \underbrace{p(y \mid x, M)}_{generation}. \tag{5.1}$$

However, this marginal probability involves an intractable summation over all size-K subsets of the memory corpora $\tilde{\mathcal{M}}$. We approximate this instead by using the top-K entries in memory with the highest probability under $p(M \mid x)$. This is reasonable if most of the unrelated memory entries do not contribute to the generation. Note that we use an online memory that is updated as the knowledge encoder is trained end-to-end with the rest of the model.

Figure 5.2 illustrates the overall workflow of ReVeaL, and we describe each component

in this section. In particular, in Sec. 5.3.1 we describe how the query is encoded. In Sec. 5.3.2 we go over how the multimodal knowledge memory is constructed and updated during pre-training. Next, we describe how we retrieve the memory entries that are most relevant to the input query in Sec. 5.3.3. Finally, in Sec. 5.3.4 we describe the generator that fuses the query and retrieved knowledge and decodes them into the generated text.

5.3.1 Query Encoding

Figure 5.2 (a) depicts how the input image-text query is encoded. We use a base visual-language encoder $b(\cdot)$ to turn the query input and each knowledge item (with potentially different modalities e.g. text-only, image-only or image-text pairs) into a sequence of embeddings (tokens). We adopt a Vision Transformer (ViT) [Dos+21] to encode the images and we use a lower-layer[1] T5 encoder [Raf+20] to encode the texts. We add a projection layer on top of the ViT model to map the image tokens into the same space as the text tokens. We then concatenate the two modalities together. We use an upper-layer T5 module as both the query Head $\phi_{\texttt{Query}}(\cdot)$ and the key Head $\phi_{\texttt{Key}}(\cdot)$ to compute the query embedding and memory keys. We take the output of the first `[CLS]` tokens followed by a linear projection and L2-normalization to summarize the input into a d-dimensional embedding.

5.3.2 Memory

Figure 5.2 (b) shows how memory is constructed and updated by encoding knowledge items. Our approach differs from previous works primarily by leveraging a diverse set of multimodal knowledge corpora (WikiData knowledge graph, Wikimedia passages and images, Web image-text pairs). Throughout the paper, we denote each corpus as $\mathcal{C}^j = \{z_1^j, \ldots, z_N^j\}$, in which each $z_i^j \in \mathcal{C}^j$ is a knowledge item that could be an image-text pair, text only, image only, or a knowledge graph triplet. We denote the unified knowledge corpus as

[1]We denote the last l layers of a T5 encoder as 'upper-layer', and the remaining ones including the token embedding layer as 'lower-layer'.

$\tilde{\mathcal{C}} = \mathcal{C}^1 \cup \mathcal{C}^2 \cdots \cup \mathcal{C}^S$ that combines $|\tilde{\mathcal{C}}| = S$ different knowledge corpora. We encode the external knowledge corpora into a unified memory $\tilde{\mathcal{M}} = [\mathcal{M}^1, \ldots, \mathcal{M}^{|\tilde{\mathcal{C}}|}]$. Each knowledge item z_i is encoded into a key/value pair $m_i = (\texttt{Emb}_{\texttt{Key}}(z_i), \texttt{Emb}_{\texttt{Value}}(z_i))$ in memory. Each key $\texttt{Emb}_{\texttt{Key}}(z) = \phi_{\texttt{Key}}\big(b(z)\big) \in \mathbb{R}^d$ is a d-dimensional embedding vector encoded via Key Head. Each value is a sequence of token embeddings representing the full information of knowledge item z. We follow a similar procedure as in [Guu+20] to precompute key/value embeddings of knowledge items from different sources and index them in a unified knowledge memory. We continuously re-compute the memory key/value embeddings as the model parameters get updated during the pre-training phase. We update the memory $\tilde{\mathcal{M}}$ asynchronously at every 1000 training steps.

Scaling Memory by Compression A naive solution for encoding the memory value is to keep the whole sequence of tokens for each knowledge item. Then, the generator could fuse the input query and the top-K retrieved memory values by concatenating all their tokens together and feeding them into a Transformer Encoder-Decoder pipeline [Lew+20]. This approach has two issues: (1) storing hundreds of millions of knowledge items in memory is impractical given that each memory value would consist of hundreds of tokens; (2) transformer encoder has quadratic complexity with respect to the total number of tokens times K for self-attention.

Therefore, we propose to use the Perceiver architecture [Jae+21] as the Value Head to encode and compress knowledge items. The Perceiver model uses a transformer decoder $\psi(\cdot)$ with learnable c-length latent embeddings to compress the full token sequence into an arbitrary length c, such that $\texttt{Emb}_{\texttt{Value}}(z) = \psi(b(z)) \in \mathbb{R}^{c \times d}$ (In our experiments we use $c = 32$). This lets us retrieve top-K memory entries for K as large as a hundred. To make the compressed embeddings generated by Perceiver more expressive, we add two additional regularizations. The first one is a disentangled regularization [Hu+22b] that forces every two output tokens to be linearly de-correlated $\mathcal{L}_{\text{decor}} = \sum_{i,j=1}^{K} \Big\| \text{Covariance}\big(\psi\big(b(z_i)\big), \psi\big(b(z_j)\big)\big) \Big\|_F^2$, and the second one is an alignment regularization that minimizes the distance of L2-Norm between the query and compressed knowledge embedding: $\mathcal{L}_{\text{align}} = \left| 1 - \frac{\sum_z \|\psi\big(b(z)\big)\|_2}{\sum_x \|b(x)\|_2} \right|$.

5.3.3 Retriever

Figure 5.2 (c) shows REVEAL's retrieval procedure. Given the input query x, the retriever's task is to find top-K memory entries M with the highest probability $p(M \mid x)$ which we approximate as $p(M \mid x) = \prod_{m \in M} p(m \mid x)$ by retrieving each entry independently. Note that we retrieve from a large-scale unified memory $\tilde{\mathcal{M}} = [\mathcal{M}^1, \ldots, \mathcal{M}^{|\tilde{\mathcal{C}}|}]$ that is constructed from a diverse set of knowledge sources. To help the query to better choose the most appropriate knowledge sources, we learn a gating function that models the probability of retrieving from each memory corpus. With the corpus gating, for $m_i^j \in \mathcal{M}^j$ we re-weight $p(m^j \mid x)$ by the computed corpus gating score:

$$p(m_i^j \mid x) = p(\mathcal{M}^j \mid x) \cdot p(m_i^j \mid x; \mathcal{M}^j) \tag{5.2}$$

$$= Gate_{\mathcal{M}^j}(x) \cdot \frac{\exp\left(Rel(x, m_i^j)/\tau\right)}{\sum_{m_k^j \in \mathcal{M}^j} \exp\left(Rel(x, m_k^j)/\tau\right)} \tag{5.3}$$

where $Gate_{\mathcal{M}^j}(x) = \text{Softmax}(W \cdot \texttt{Emb}_{\texttt{Query}}(x) + b)[j]$ is a softmax gating that assigns a score to each memory corpus M^j, with W and b as function parameters. $Rel(x, m_i^j)$ models relevance score between query x and each memory entry via embedding dot product, such that $Rel(x, m_i^j) = \texttt{Emb}_{\texttt{Query}}(x)^T \cdot \texttt{Emb}_{\texttt{Key}}(z_i^j)$. where z_i is the knowledge item corresponding to the memory entry m_i and τ is the temperature parameter.

After identifying the top-K memory entries, the retriever passes the pre-computed in-memory key and value embeddings to the generator. In the meantime, to support end-to-end training of the encoders, we also re-encode a small portion (i.e., 10%) of the retrieved knowledge items z_i from scratch. In this way, the memory encoders could be updated with moderate computational cost. We concatenate the re-encoded knowledge with in-memory ones to construct the final top-K retrieved key/value embeddings.

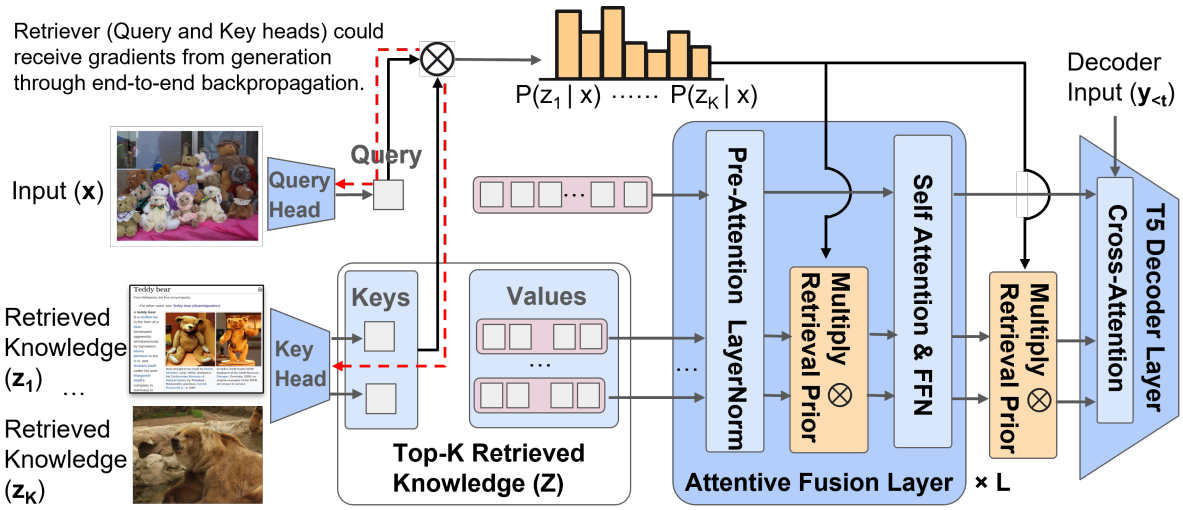


Figure 5.3: Detailed procedure of attentive knowledge fusion module. We inject retrieval probability as a prior to knowledge token embeddings, so the retriever can receive gradients via back-propagating over {self/cross}-attention part.

5.3.4 Generator

Figure 5.2 (d) shows how the query and the retrieved knowledge items are fused to generate the output answer. All K retrieved memory values are concatenated with the query embedding, which is feasible due to the Perceiver module utilized as the value head $\psi(\cdot)$, compressing each knowledge item into a short sequence. We denote the concatenated query embedding and memory values as $X = [b(x), \psi(b(z_1)), \ldots, \psi(b(z_K))] \in \mathbb{R}^{(I+c\cdot K)\times d}$, where I is the number of tokens of the input query x and c is the number of compressed tokens. To guide the generator towards attending to the most important items in X and facilitate backpropagation of gradients to the retriever, we propose an attentive fusion module $f(\cdot)$ capable of incorporating the retriever score as a prior for calculating cross-knowledge attention. The detailed procedure is illustrated in Figure 5.3. We firstly compute a latent soft attention mask over X as $\texttt{Mask}_{\texttt{att}} = [1, p(z_1|x), \ldots, p(z_K|x)]$. Finally, we pass the fused representation $f(X, \texttt{Mask}_{\texttt{att}})$ into a T5 decoder module $g(\cdot)$ to generate the textual output.

Knowledge Source	Corpus Size	Type of Text	Avg. Text Length
WIT [Sri+21]	5,233,186	Wikipedia Passage	258
CC12M [Cha+21]	10,009,901	Alt-Text Caption	37
VQA-V2 [Goy+17]	123,287	Question Answer	111
WikiData [VK14]	4,947,397	Linearlized Triplets	326

Table 5.1: Statistics of the knowledge sources used.

Model Name	T5 Variant	Image Encoder	# params.	GFLOPs
ReVeaL-Base	T5-Base	ViT-B/16	0.4B	120
ReVeaL-Large	T5-Large	ViT-L/16	1.4B	528
ReVeaL	T5-Large	ViT-g/14	2.1B	795

Table 5.2: Model configuration of different ReVeaL variants.

5.4 Generative Pre-Training

The existing VQA datasets are not large enough for training a complex multi-component model like ours from scratch. Therefore, we pre-train our model on a massive image-text corpus. In Sec. 5.4.1 we go over the details of our pre-training data and objective. Then in Sec. 5.4.2 we introduce the various sources of knowledge used in our experiments. Finally, in Sec. 5.4.3 we describe the pre-training implementation details.

5.4.1 Pre-Training Objective

We pre-train our model on the Web-Image-Text dataset [Zha+22b], a large-scale corpus containing 3 billion image alt-text caption pairs collected from the public Web. Since the dataset is noisy, we add a filter to remove data points whose captions are shorter than 50 characters. This yields roughly 1.3 billion image caption pairs for pre-training.

We denote the pre-training Web-Image-Text dataset [Zha+22b] as $\mathcal{D}$. We use the text generation objective used in Wang et al. [Wan+22c]) to pre-train our model on $\mathcal{D}$. Given an image-text example $x = (\texttt{img}, \texttt{txt})$ from $\mathcal{D}$, we randomly sample a prefix length T_p. We feed $x_{<T_p}$ that contains the text prefix and image to the model as input and our objective is to generate $x_{\geq T_p}$ containing the rest of the text as output. The training goal is to condition

on $x_{<T_p}$ and autoregressively generate the remaining text sequence $x_{\geq T_p}$:

$$\begin{aligned} \mathcal{L}_{\text{PrefixLM}} &= -\mathbb{E}_{x\sim\mathcal{D}}\big[\log p(x_{\geq T_p} \mid x_{<T_p})\big] \\ &= -\mathbb{E}_{x\sim\mathcal{D}}\Big[\sum_{i\geq T_p} \log p(x_i \mid x_{<i})\Big]. \end{aligned} \tag{5.4}$$

Warm Starting the Model In order to pre-train all components of our model end-to-end, we need to warm start the retriever at a good state. Otherwise, if starting with random weights, the retriever would often return irrelevant memory items that would never generate useful training signals.

To avoid this cold-start problem, we propose to construct an initial retrieval dataset with pseudo ground-truth knowledge to give the pre-training a reasonable head start. We create a modified version of the Wikipedia-Image-Text (WIT) [Sri+21] dataset for this purpose. Each image-caption pair in WIT also comes with a corresponding Wikipedia passage (words surrounding the text). We put together the surrounding passage with the query image and use it as the pseudo ground-truth knowledge that corresponds to the input query. As the passage provides rich information about the image and caption, it definitely is useful for initializing the model. To avoid the model from relying on low-level image features for retrieval, we apply random data augmentation to the input query image. Given this modified dataset that contains pseudo retrieval ground-truth, we train the query and memory key embeddings by optimizing the following contrastive loss:

$$\mathcal{L}_{contra} = -\text{logSoftmax}(\texttt{Emb}_{\texttt{Query}}(x)^T \texttt{Emb}_{\texttt{Key}}(\hat{z}))$$

where $\hat{z}$ represents the pseudo ground-truth knowledge entry corresponding to the input query x.

VQA Model Name	Knowledge Sources	Accuracy (%)	Memory (GB)
MUTAN+AN [Mar+19]	Wikipedia + ConceptNet	27.8	-
ConceptBERT [Gar+20]	Wikipedia	33.7	-
KRISP [Mar+21]	Wikipedia + ConceptNet	38.4	-
Visual Retriever-Reader [Luo+21]	Google Search	39.2	-
MAVEx [Wu+22]	Wikipedia+ConceptNet+Google Images	39.4	-
KAT-Explicit [Gui+21]	Wikidata	44.3	1.5
PICa-Base [Yan+21]	Frozen GPT-3	43.3	350
PICa-Full [Yan+21]	Frozen GPT-3	48.0	350
KAT [Gui+21] (Single)	Wikidata + Frozen GPT-3	53.1	1.5 + 352+ 500
KAT [Gui+21] (Ensemble)	Wikidata + Frozen GPT-3	54.4	4.6 + 352 + 500
ReVIVE [Lin+22] (Single)	Wikidata + Frozen GPT-3	56.6	1.5 + 354 + 500
ReVIVE [Lin+22] (Ensemble)	Wikidata+Frozen GPT-3	58.0	4.6 + 354 + 500
REVEAL-Base	WIT + CC12M + Wikidata + VQA-2	55.2	0.8 + 7.5 + 744
REVEAL-Large	WIT + CC12M + Wikidata + VQA-2	58.0	2.8 + 10 + 993
REVEAL	WIT + CC12M + Wikidata + VQA-2	**59.1**	4.2 + 10 + 993

Table 5.3: **Visual Question Answering** results on OK-VQA, compared with existing methods that use different knowledge sources. For the memory cost, we assume all models use bfloat16. Green means on-device model parameters that are learnable, Blue means on-device memory of frozen model parameters, and Red means CPU/disk storage cost that are not involved in computation.

5.4.2 Knowledge Sources

We use the following four sources of knowledge in our experiments: **Wikipedia-Image-Text (WIT)** [Sri+21] consists of the images in Wikipedia, as well as their alt-text captions and contextualized text passages. **Conceptual (CC12M)** [Cha+21] contains web images paired with alt-text captions. It includes many long-tail entities. **VQA-v2** [Goy+17] is a visual question answering dataset. We merge all question-answer pairs per image into a single passage. **WikiData** [VK14] is a structural knowledge graph encoding relations between Wikipedia entities. We linearize all relational triplets per entity into a textual passage following the procedure of [Ogu+20]. We have listed the statistical details of these knowledge sources in Table 5.1.

5.4.3 Implementation Details

Incorporating all the components introduced above, REVEAL can be directly pre-trained over large-scale image caption datasets after proper initialization. As our model architecture is based on T5 and ViT, we use pre-trained ViT checkpoints from [Zha+22a] and pre-trained T5 checkpoints from [Raf+20] to initialize the encoder parameters. The query head, key head and attentive fusion layers are initialized from upper T5, while the base text encoder

is initialized from lower T5. The combination of these modules can be found in Table 5.2 for three three model variants, REVEAL-Base, REVEAL-Large and REVEAL, of which the largest REVEAL model has around 2 billion parameters.

Distributed Online Retrieval. Finding the top-k most-relevant knowledge entries is a standard Maximum Inner Product Search (MIPS) problem. There are approximate search algorithms [SL14; Che+22c] that scale sub-linearly with the size of the knowledge corpus $|C|$. We use TPU-KNN [Che+22c] to conduct distributed MIPS search, by splitting and storing the memory embeddings across all training devices. The query is synced to each device, which retrieves approximate top-K results from its own memory. Then these results are combined to compute the global top-K retrieved items.

Pre-Training Pipeline. We first train the multimodal retriever on our modified version of the Wikipedia Image Text (WIT) dataset via $\mathcal{L}_{contra}$. We use the Adafactor optimizer without momentum ($\beta_1 = 0$, $\beta_2 = 0.999$), with weight decay of 0.001[2], and with a peak learning rate of $6e4$, to train for 10 epochs. We use this checkpoint to warm-start our generative pre-training. We set the number of retrieved knowledge entries as $K = 10$ during pre-training, and use adafactor with a peak learning rate of $1e{-}3$ and inverse squared root learning rate scheduler with 10,000 linear warm-up steps. We use $\mathcal{L}_{PrefixLM}$ as the main objective, adding $\mathcal{L}_{contra}$, $\mathcal{L}_{decor}$ and $\mathcal{L}_{align}$ weighted by 0.01. We use a batch size of 4096 across 256 CloudTPUv4 chips and train for about 5 days.

5.5 Experimental Results

We evaluate our proposed method on knowledge-based VQA in Sec. 5.5.1 and image captioning in Sec. 5.5.2. We then conduct ablation studies in Sec. 5.5.3 to analyze the impact of each model component on overall performance.

[2]The remaining experiments use the same optimizer configuration.

VQA Model Name	Accuracy (%)
ViLBERT [Lu+19]	30.6
LXMERT [TB19]	30.7
ClipCap [MHB21]	30.9
KRISP [Mar+21]	33.7
GPV-2 [Kam+22]	48.6
REVEAL-Base	50.4
REVEAL-Large	51.5
REVEAL	**52.2**

Table 5.4: **Visual Question Answering** results on A-OKVQA.

5.5.1 Evaluating on Knowledge-Based VQA

One of the most knowledge intensive visual-language tasks is knowledge-based visual question answering (VQA), exemplified by the OK-VQA [Mar+19] and A-OKVQA [Sch+22] benchmarks. To finetune our pre-trained model on these VQA tasks, we use the same generative objective where the model takes in an image question pair as input and generates the text answer as output. There are a few differences between the fine-tuning and the pre-training stages: 1) we set the number of retrieved knowledge entries to $K = 50$, so the model is able to retrieve sufficient supporting evidence; 2) we freeze the whole base V-L encoder to stabilize training; and 3) we use a batch size of 128, with the Adafactor optimizier, a peak learning rate of 1e-4. We use the soft VQA accuracy metric [Ant+15] to evaluate the model's generated answer.

Our results on OKVQA and A-OKVQA datasets are shown in Table 5.3 and Table 5.4 respectively. For OKVQA, earlier attempts that incorporate a fixed knowledge retriever report results that are below 45%. Recently a series of works utilize large language models (e.g. GPT-3) as implicit knowledge sources, which achieve much better performance with the trade-off of a huge computational cost. REVEAL achieves higher performance than those methods without relying on such large language models[3]. Compared with the previous

[3]As shown in the last column of Table 5.3, REVEAL stores external knowledge as value embeddings on disk, occupying 993GB of space. The key embeddings consume 10GB space and are kept in TPU memory for fast lookup. On the other hand, KAT and REVIVE need to load the entire 350GB GPT-3 model in the GPU/TPU memory. Furthermore, storing WikiData on disk consumes 500GB of disk memory.

Model Name	MSCOCO	NoCaps	# params.
Flamingo [Ala+22]	138.1	-	80B
VinVL [Zha+21a]	140.9	105.1	0.4B
SimVLM [Wan+22c]	143.3	112.2	1.5B
CoCa [Yu+22b]	143.6	122.4	2.1B
ReVeaL-Base	141.1	115.8	0.4B
ReVeaL-Large	144.5	121.3	1.4B
ReVeaL	**145.4**	**123.0**	2.1B

Table 5.5: **Image Captioning** results on MSCOCO (Karpathy-test split) and NoCaps (val set). Evaluated using the CIDEr metric.

state-of-the-art, KAT and ReVIVE, which also utilizes T5-Large as a generator, ReVeaL achieves accuracy of 59.1%, which is +6.0% higher than the single KAT [Gui+21] model and +2.5% higher than ReVIVE [Lin+22].

On A-OKVQA, ReVeaL achieves 52.2% accuracy, which is +3.6% higher than the previous best, GPV-2 [Kam+22]. We also show two examples of these datasets in Figure 5.4. All these results show that, with proper end-to-end retrieval training and a diverse set of knowledge sources, ReVeaL can learn to retrieve meaningful knowledge entries, and achieve promising results without relying on a large language model.

5.5.2 Evaluating on Image Captioning

We also evaluate ReVeaL on image captioning benchmarks: MSCOCO Captions [Che+15a] and NoCaps [Agr+19]. We follow the evaluation protocol used in [Yu+22b]. We directly fine-tune our generator model on the MSCOCO training split via cross-entropy generative objective. We measure our performance on the MSCOCO test split and NoCaps val set with the CIDEr metric [VZP15]. The results of these two datasets are shown in Table 5.5. Note that ReVeaL achieves better results than strong recent baselines such as SimVLM [Wan+22c] and CoCa [Yu+22b] on both benchmarks. Notably, ReVeaL −Large with 1.4B parameters outperforms the 2.1B-parameter CoCa model and is significantly better than 80B-parameter Flamingo model [Ala+22].

Figure 5.4: **VQA Examples**. REVEAL is able to use knowledge from different sources to correctly answer the question. We show more examples in Figure 1-3 of Supplementary Material, indicating that our model can retrieve and use items from diverse knowledge sources to correctly solve different input query.

5.5.3 Analyzing Effects of Key Model Components

In the following we study which design choices contribute most to the model's performance. We focus on three research questions: (1) Does utilizing multiple knowledge sources enhance performance? (2) Does the proposed attentive fusion surpass existing end-to-end retrieval training methods? (3) Can we add knowledge by only updating the memory without modifying model parameters?

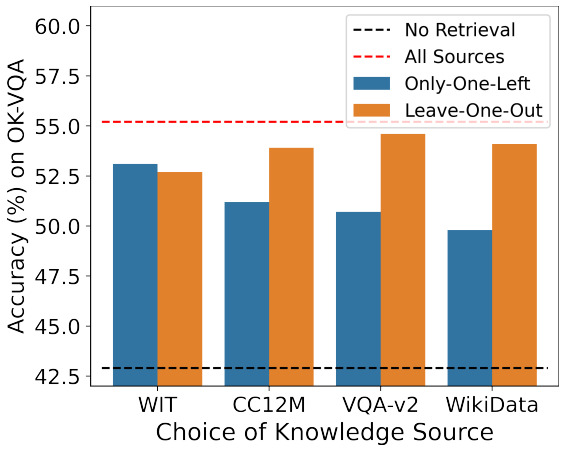


Figure 5.5: OKVQA Accuracy of ReVeaL using 1) **Only-One-Left**: only use a single knowledge source; 2) **Leave-One-Out**: use all without this knowledge source.

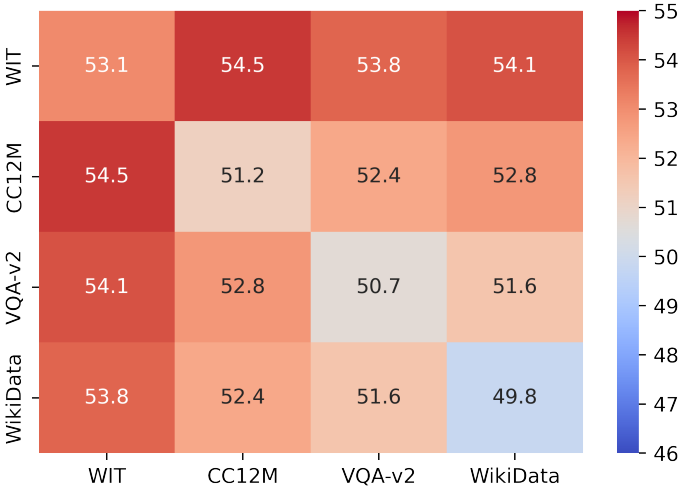


Figure 5.6: OKVQA Accuracy of ReVeaL using all **Pair of Knowledge Sources**. Results show that combining multiple sources could consistently improve performance.

Analyzing multiple knowledge sources. A major distinction of ReVeaL compared to previous retrieval-augmented approaches is its capacity to utilize a diverse set of knowledge sources during inference. To assess the relative importance of each data source and the efficacy of retrieving from various corpora, we conduct two ablation studies: 1) **Only-One-Left**: employing a single knowledge source to evaluate the outcomes; and 2) **Leave-One-Out**: excluding one knowledge source from the complete set $\mathcal{C}$. These ablation studies are executed using the ReVeaL $_{\text{Base}}$, evaluated on the OKVQA validation set under the aforementioned conditions. As shown in Figure 5.5, among the four knowledge sources utilized in this paper, WIT is the most informative, with the highest accuracy when used in isolation (53.1The remaining three corpora, CC12M, VQA-v2, and WikiData, do not offer the same level of informativeness as WIT when utilized independently. However, excluding any of these corpora from the complete dataset results in performance decreases of 1.3%, 0.6%, and 1.1%, respectively. This observation implies that these knowledge sources effectively complement one another, contributing valuable information to enhance performance. To further substantiate this hypothesis, we perform an additional experiment involving pairs of knowledge sources, as illustrated in Figure 5.6. Notably, even when paired with an informative knowledge source such as WIT, incorporating an extra corpus

consistently leads to performance improvements.

Analyzing different retrieval training methods. Another core component of REVEAL is the attentive fusion layer, which supports efficient joint training of the retriever and generator. We investigate its performance compared to two existing retrieval training method categories: 1) a frozen retriever based on ALIGN [Jia+21] representations ; 2) end-to-end retrieval training methods including **Attention Distill** [IG21], **EMDR**^{2} [Yan+19a], and **Perplexity Distill** [Iza+22].

We use the pre-trained REVEAL-Base model, fix the generator and randomly initialize the retriever (query head and key head). We utilize our modified version of WIT dataset with pseudo ground-truth retrieved labels as the evaluation corpus. We evaluate retrieval performance by checking whether the correct passage appears in top-10/100 results. For the ALIGN model, we directly evaluate the retrieval results from the pre-trained checkpoint, while for other models, we perform retrieval-augmented training on the WIT dataset. To prevent the model from relying on image similarity for accurate results, we only use text passages as knowledge entries and discard images. Subsequently, we finetune the model on OKVQA and report its accuracy. The results are presented in Table 5.6. We observe that directly using pre-trained encoder does not perform well, even with a strong model like ALIGN. Moreover, among the various end-to-end retrieval training approaches, our attentive fusion method attains better accuracy in both retrieval and OKVQA tasks. Importantly, our technique exhibits a computational cost (quantified by GFLOPs) comparable to that of attention distillation, yet significantly lower than EMDR^2 and Perplexity distillation. This indicates that our proposed method is more efficient and effective for pre-training retrieval-augmented visual-language models.

Analyzing Knowledge Modification. One advantage of utilizing knowledge memory is that we could easily add or update knowledge entries without re-training model's parameters. To validate this, we conducted ablation studies in which we removed a specific percentage of knowledge entries from the corpora and assessed the performance of the REVEAL-Base model on the OKVQA dataset. Subsequently, we add the removed knowledge

Retrieval Method	Acc@10	Acc@100	OKVQA Acc.	GFLOPs
ALIGN [Jia+21] (fixed)	0.638	0.793	44.7	-
Attention Distill [IG21]	0.674	0.835	45.9	119
EMDR^2 [Yan+19a]	0.691	0.869	46.5	561
Perplexity Distill [Iza+22]	0.704	0.886	46.7	561
Ours (Attentive Fusion)	0.726	0.894	47.3	120

Table 5.6: **Analysis of Retrieval Training Method**: We train ReVeaL-Base (frozen generator, only train randomly initialized retriever) to retrieve from the WIT dataset (only text passage without image), and show the retrieval accuracy at the first 10 or 100 results, as well as fine-tuned OKVQA accuracy.

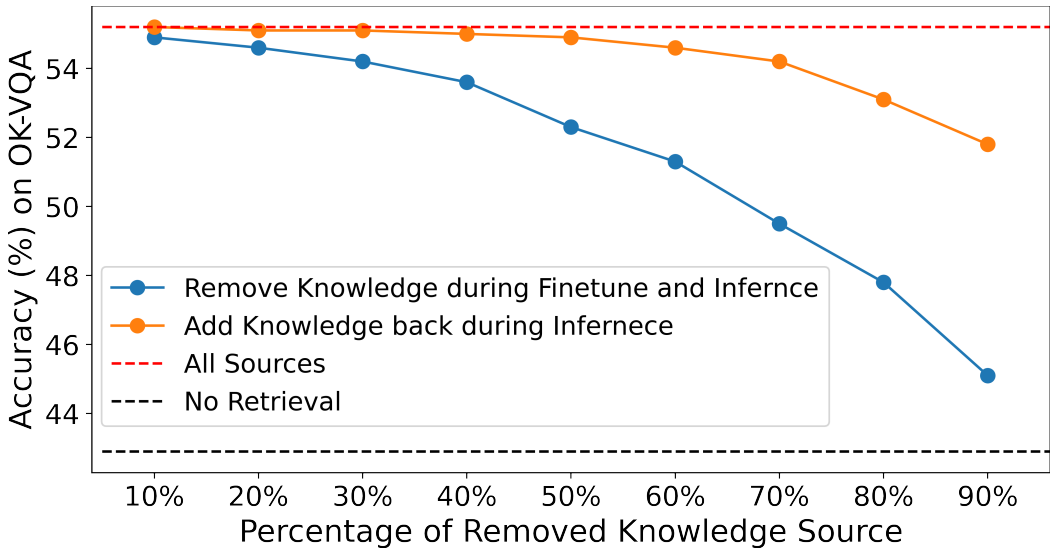


Figure 5.7: **Study of Knowledge Update**. The blue curve shows result by removing certain percentage of knowledge during both fine-tuning and inference stage. The orange curve shows results by still first removing the knowledge, and then adding the knowledge back during inference, which simulates the knowledge update.

back into the corpora, allowing the trained model to make predictions using the complete set of corpora. This approach ensured that the removed knowledge was not seen by the model during fine-tuning, enabling us to test its ability to accurately retrieve and utilize that knowledge for problem-solving.

The results are illustrated in Figure 5.7, with the blue curves representing the inference outcomes without the removed knowledge and the orange curve depicting the results after adding the removed knowledge back. A notable performance improvement was observed upon reintroducing the knowledge (orange curve) compared to the outcomes with the

removed knowledge (blue curve). Specifically, for the model fine-tuned with only 10% of the knowledge, the reintroduction of the removed knowledge resulted in an accuracy of 51.8 (+6.7 higher than when removed). This finding demonstrates that the REVEAL model can swiftly adapt to new knowledge by merely updating the memory, obviating the need for re-training model parameters.

5.6 Summary

This paper presents an end-to-end Retrieval-augmented Visual Language model (REVEAL), which contains a knowledge retriever that learns to utilize a diverse set of knowledge sources with different modality. The retriever is trained jointly with the generator to return multiple knowledge entries. We pre-train REVEAL on a massive image-text corpus with four diverse knowledge corpora, and achieves state-of-the-art results on knowledge-intensive visual question answering and image caption tasks. In the future we'd explore the ability of this model to be used for attribution, and applying it to broader class of multimodal tasks.

Part III

Generalize across Domains via Symbolic Knowledge

CHAPTER 6

Few-Shot Representation for Out-Of-Vocabulary Word

Existing approaches for learning word embeddings often assume there are sufficient occurrences for each word in the corpus, such that the representation of words can be accurately estimated from their contexts. However, in real-world scenarios, out-of-vocabulary (a.k.a. OOV) words that do not appear in training corpus emerge frequently. It is challenging to learn accurate representations of these words with only a few observations. In this paper, we formulate the learning of OOV embeddings as a few-shot regression problem, and address it by training a representation function to predict the oracle embedding vector (defined as embedding trained with abundant observations) based on limited observations. Specifically, we propose a novel hierarchical attention-based architecture to serve as the neural regression function, with which the context information of a word is encoded and aggregated from K observations. Furthermore, our approach can leverage Model-Agnostic Meta-Learning (MAML) for adapting the learned model to the new corpus fast and robustly. Experiments show that the proposed approach significantly outperforms existing methods in constructing accurate embeddings for OOV words, and improves downstream tasks where these embeddings are utilized.

6.1 Introduction

Distributional word embedding models aim to assign each word a low-dimensional vector representing its semantic meaning. These embedding models have been used as key components in natural language processing systems. To learn such embeddings, existing approaches such as skip-gram models [Mik+13b] resort to an auxiliary task of predicting the context words (words surround the target word). These embeddings have shown to be able to capture syntactic and semantic relations between words.

Despite the success, an essential issue arises: most existing embedding techniques assume the availability of abundant observations of each word in the training corpus. When a word occurs only a few times during training (i.e., in the few-shot setting), the corresponding embedding vector is not accurate [Coh+17]. In the extreme case, some words are not observed when training the embedding, which are known as out-of-vocabulary (OOV) words. These words are often rare and might only occurred for a few times in the testing corpus. Therefore, the insufficient observations hinder the existing context-based word embedding models to infer accurate OOV embeddings. This leads us to the following research problem: How can we learn accurate embedding vectors for OOV words during the inference time by observing their usages for only a few times?

Existing approaches for dealing with OOV words can be categorized into two groups. The first group of methods derives embedding vectors of OOV words based on their morphological information [Boj+17; Kim+16; PGE17]. This type of approaches has a limitation when the meaning of words cannot be inferred from its subunits (e.g., names, such as Vladimir). The second group of approaches attempts to learn to embed an OOV word from a few examples. In a prior study [Coh+17; HB17], these demonstrating examples are treated as a small corpus and are used to fine-tune OOV embeddings. Unfortunately, fine-tuning with just a few examples usually leads to overfitting. In another work [Kho+18], a simple linear function is used to infer embedding of an OOV word by aggregating embeddings of its context words in the examples. However, the simple linear averaging can fail to capture

the complex semantics and relationships of an OOV word from its contexts.

Unlike the existing approaches mentioned above, humans have the ability to infer the meaning of a word based on a more comprehensive understanding of its contexts and morphology. Given an OOV word with a few example sentences, humans are capable of understanding the semantics of each sentence, and then aggregating multiple sentences to estimate the meaning of this word. In addition, humans can combine the context information with sub-word or other morphological forms to have a better estimation of the target word. Inspired by this, we propose an attention-based hierarchical context encoder (HiCE), which can leverage both sentence examples and morphological information. Specifically, the proposed model adopts multi-head self-attention to integrate information extracted from multiple contexts, and the morphological information can be easily integrated through a character-level CNN encoder.

In order to train HiCE to effectively predict the embedding of an OOV word from just a few examples, we introduce an episode based few-shot learning framework. In each episode, we suppose a word with abundant observations is actually an OOV word, and we use the embedding trained with these observations as its oracle embedding. Then, the HiCE model is asked to predict the word's oracle embedding using only the word's K randomly sampled observations as well as its morphological information. This training scheme can simulate the real scenarios where OOV words occur during inference, while in our case we have access to their oracle embeddings as the learning target. Furthermore, OOV words may occur in a new corpus whose domain or linguistic usages are different from the main training corpus. To deal with this issue, we propose to adopt Model-Agnostic Meta-Learning (MAML) [FAL17] to assist the fast and robust adaptation of a pre-trained HiCE model, which allows HiCE to better infer the embeddings of OOV words in a new domain by starting from a promising initialization.

We conduct comprehensive experiments based on both intrinsic and extrinsic embedding evaluation. Experiments of intrinsic evaluation on the Chimera benchmark dataset demonstrate that the proposed method, HiCE, can effectively utilize context information and

outperform baseline algorithms. For example, HiCE achieves 9.3% relative improvement in terms of Spearman correlation compared to the state-of-the-art approach, *à la carte*, regarding 6-shot learning case. Furthermore, with experiments on extrinsic evaluation, we show that our proposed method can benefit downstream tasks, such as named entity recognition and part-of-speech tagging, and outperform existing methods significantly.

The contributions of this work can be summarized as follows.

1. We formulate the OOV word embedding learning as a K-shot regression problem and propose a simulated episode-based training schema to predict oracle embeddings.

2. We propose an attention-based hierarchical context encoder (HiCE) to encode and aggregate both context and sub-word information. We further incorporate MAML for fast adapting the learned model to the new corpus by bridging the semantic gap.

3. We conduct experiments on multiple tasks, and through quantitative and qualitative analysis, we demonstrate the effectiveness of the proposed method in fast representation learning of OOV words for down-stream tasks.

6.2 Related Work

OOV Word Embedding Previous studies of handling OOV words were mainly based on two types of information: 1) context information and 2) morphology features.

The first family of approaches follows the distributional hypothesis [Fir57] to infer the meaning of a target word based on its context. If sufficient observations are given, simply applying existing word embedding techniques (e.g., word2vec) can already learn to embed OOV words. However, in a real scenario, mostly the OOV word only occur for a very limited times in the new corpus, which hinders the quality of the updated embedding [LMB17; HB17]. Several alternatives have been proposed in the literature. Lazaridou, Marelli, and Baroni [LMB17] proposed additive method by using the average embeddings of context words [LMB17] as the embedding of the target word. Herbelot and

Baroni [HB17] extended the skip-gram model to *nonce2vec* by initialized with additive embedding, higher learning rate and window size. Khodak, Saunshi, Liang, Ma, Stewart, and Arora [Kho+18] introduced *à la carte*, which augments the additive method by a linear transformation of context embedding.

The second family of approaches utilizes the morphology of words (e.g., morphemes, character n-grams and character) to construct embedding vectors of unseen words based on sub-word information. For example, Luong, Socher, and Manning [LSM13] proposed a morphology-aware word embedding technique by processing a sequence of morphemes with a recurrent neural network. Bojanowski, Grave, Joulin, and Mikolov [Boj+17] extended skip-gram model by assigning embedding vectors to every character n-grams and represented each word as the sum of its n-grams. Pinter, Guthrie, and Eisenstein [PGE17] proposed MIMICK to induce word embedding from character features with a bi-LSTM model. Although these approaches demonstrate reasonable performance, they rely mainly on morphology structure and cannot handle some special type of words, such as transliteration, entity names, or technical terms.

Our approach utilizes both pieces of information for an accurate estimation of OOV embeddings. To leverage limited context information, we apply a complex model in contrast to the linear transformation used in the past, and learn to embed in a few-shot setting. We also show that incorporating morphological features can further enhance the model when the context is extremely limited (i.e., only two or four sentences).

Few-shot learning The paradigm of learning new tasks from a few labelled observations, referred to as few-shot learning, has received significant attention. The early studies attempt to transfer knowledge learned from tasks with sufficient training data to new tasks. They mainly follow a pre-train then fine-tune paradigm [Don+14a; Ben12; Zop+16]. Recently, meta-learning is proposed and it achieves great performance on various few-shot learning tasks. The intuition of meta-learning is to learn generic knowledge on a variety of learning tasks, such that the model can be adapted to learn a new task with only a few training

samples. Approaches for meta-learning can be categorized by the type of knowledge they learn. (1) Learn a metric function that embeds data in the same class closer to each other, including Matching Networks [Vin+16], and Prototypical Networks [SSZ17]. The nature of metric learning makes it specified on classification problems. (2) Learn a learning policy that can fast adapt to new concepts, including a better weight initialization as MAML [FAL17] and a better optimizer [RL17]. This line of research is more general and can be applied to different learning paradigms, including both classification and regression.

There have been emerging research studies that utilize the above meta-learning algorithms to NLP tasks, including language modelling [Vin+16], text classification [Yu+18], machine translation [Gu+18], and relation learning [Xio+18; Gao+19]. In this paper, we propose to formulate the OOV word representation learning as a few-shot regression problem. We first show that pre-training on a given corpus can somehow solve the problem. To further mitigate the semantic gap between the given corpus with a new corpus, we adopt model-agnostic meta-learning (MAML) [FAL17] to fast adapt the pre-trained model to new corpus.

Contextualized Embedding The HiCE architecture is related to contextualized representation learning [Pet+18; Dev+]. However, their goal is to get a contextualized embedding based on a given sentence, with word or sub-word embeddings as input. In contrast, our work utilizes multiple contexts to learn OOV embeddings. This research direction is orthogonal to their goal. In addition, the OOV embeddings learned by ours can be served as inputs to ELMO and BERT, helping them to deal with OOV words.

6.3 Methodology

In this section, we first formalize the problem of OOV embedding learning as a few-shot regression problem. Then, we present our embedding prediction model, a hierarchical context encoder (HiCE) for capturing the semantics of context as well as morphological

features. Finally, we adopt a state-of-the-art meta-learning algorithm, MAML, for fast and robust adaptation to a new corpus.

6.3.1 The Few-Shot Regression Framework

Problem formulation We consider a training corpus D_T, and a given word embedding learning algorithm (e.g., Word2Vec) that yields a learned word embedding for each word w, denoted as $T_w \in \mathbb{R}^d$. Our goal is to infer embeddings for OOV words that are not observed in the training corpus D_T based on a new testing corpus D_N.

D_N is usually much smaller than D_T and the OOV words might only occur for a few times in D_N, thus it is difficult to directly learn their embedding from D_N. Our solution is to learn an neural regression function $F_\theta(\cdot)$ parameterized with θ on D_T. The function $F_\theta(\cdot)$ takes both the few contexts and morphological features of an OOV word as input, and outputs its approximate embedding vector. The output embedding is expected to be close to its *"oracle" embeddings* vector that assumed to be learned with plenty of observations.

To mimic the real scenarios of handling OOV words, we formalize the training of this model in a few-shot regression framework, where the model is asked to predict OOV word embedding with just a few examples demonstrating its usage. The neural regression function $F_\theta(\cdot)$ is trained on D_T, where we pick N words $\{w_t\}_{t=1}^N$ with sufficient observations as the target words, and use their embeddings $\{T_{w_t}\}_{t=1}^N$ as *oracle embeddings*. For each target word w_t, we denote S_t as all the sentences in D_T containing w_t. It is worth noting that we exclude words with insufficient observations from target words due to the potential noisy estimation for these words in the first place.

In order to train the neural regression function $F_\theta(\cdot)$, we form episodes of few-shot learning tasks. In each episode, we randomly sample K sentences from S_t, and mask out w_t in these sentences to construct a masked supporting context set $\mathbf{S}_t^K = \{s_{t,k}\}_{k=1}^K$, where $s_{t,k}$ denotes the k-th masked sentence for target word w_t. We utilize its character sequence as features, which are denoted as C_t. Based on these two types of features, the model F_θ is

learned to predict the oracle embedding. In this paper, we choose cosine similarity as the proximity metric, due to its popularity as an indicator for the semantic similarity between word vectors. The training objective is as follows.

$$\hat{\theta} = \arg\max_{\theta} \sum_{w_t} \sum_{\mathbf{S}_t^K \sim \mathbf{S}_t} \cos\left(F_\theta(\mathbf{S}_t^K, C_t), T_{w_t}\right), \tag{6.1}$$

where $\mathbf{S}_t^K \sim \mathbf{S}_t$ means that the K sentences containing target word w_t are randomly sampled from all the sentences containing w_t. Once the model $F_{\hat{\theta}}$ is trained (based on D_T), it can be used to predict embedding of OOV words in D_N by taking all sentences containing these OOV words and their character sequences as input.

6.3.2 Hierarchical Context Encoding (HiCE)

Here we detail the design of the neural regression function $F_\theta(\cdot)$. Based on the previous discussion, $F_\theta(\cdot)$ should be able to analyze the complex semantics of context, to aggregate multiple pieces of context information for comprehensive embedding prediction, and to incorporate morphological features. These three requirements cannot be fulfilled using simple models such as linear aggregation [Kho+18].

Recent progress in contextualized word representation learning [Pet+18; Dev+] has shown that it is possible to learn a deep model to capture richer language-specific semantics and syntactic knowledge purely based on self-supervised objectives. Motivated by their results, we propose a hierarchical context encoding (HiCE) architecture to extract and aggregate information from contexts, and morphological features can be easily incorporated. Using HiCE as $F_\theta(\cdot)$, a more sophisticated model to process and aggregate contexts and morphology can be learned to infer OOV embeddings.

Self-Attention Encoding Block Our proposed HiCE is mainly based on the self-attention encoding block proposed by Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin [Vas+17]. Each encoding block consists of a self-attention layer

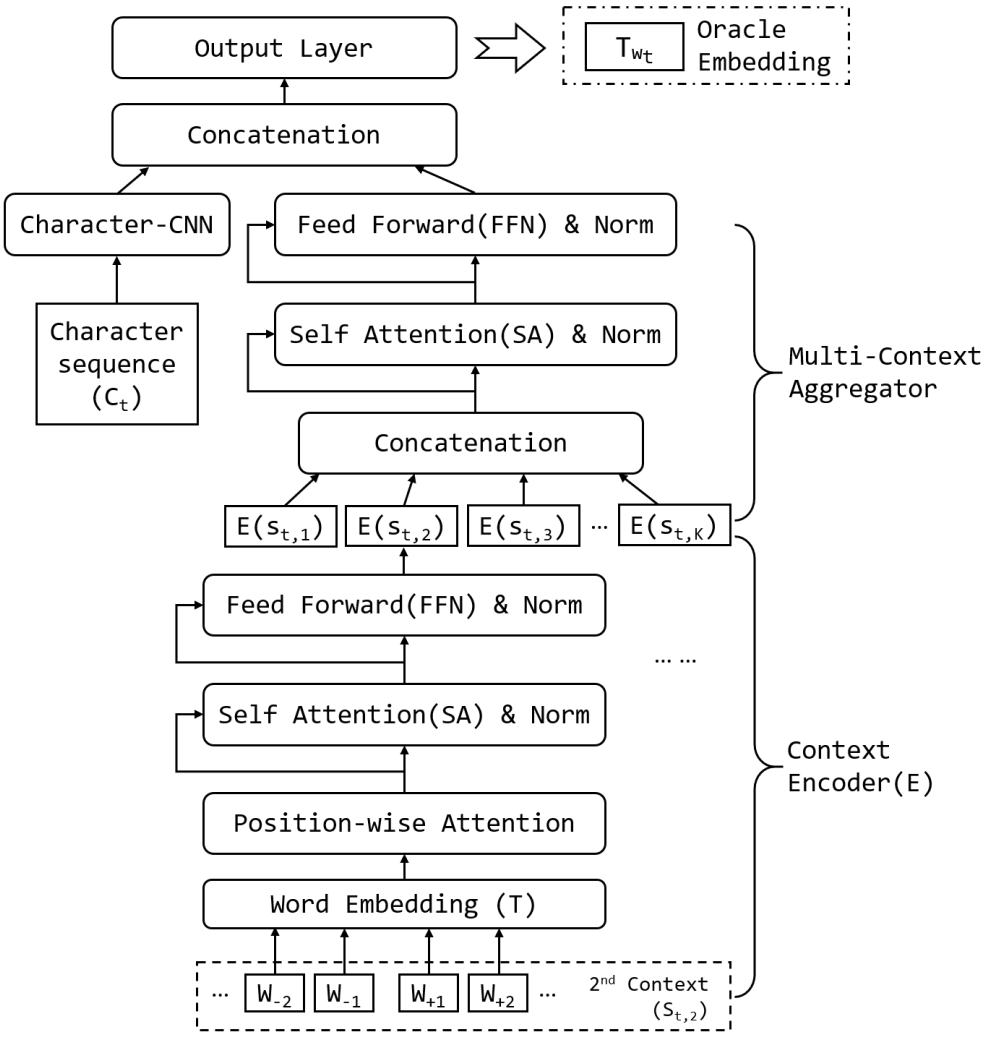


Figure 6.1: The proposed hierarchical context encoding architecture (HiCE) for learning embedding representation for OOV words.

and a point-wise, fully connected layer. Such an encoding block can enrich the interaction of the sequence input and effectively extract both local and global information.

Self-attention (SA) is a variant of attention mechanism that can attend on a sequence by itself. For each head i of the attention output, we first transform the sequence input matrix x into query, key and value matrices, by a set of three different linear projections W_i^Q, W_i^K, W_i^V. Next we calculate matrix product $xW_i^Q(xW_i^K)^T$, then scale it by the square root of the dimension of the sequence input $\frac{1}{\sqrt{d_x}}$ to get mutual attention matrix of the sequence. Finally we aggregate the value matrices using the calculated attention matrix,

and get $a_{self,i}$ as the self attention vector for head i:

$$a_{self,i} = softmax\left(\frac{xW_i^Q\ (xW_i^K)^T}{\sqrt{d_x}}\right) xW_i^V.$$

Combining all these self-attentions $\{a_{self,i}\}_{i=1}^{h}$ by a linear projection W^O, we have a $SA(x)$ with totally h heads, which can represent different aspects of mutual relationships of the sequence x:

$$SA(x) = Concat(a_{self,1}, ..., a_{self,h})W^O.$$

The self-attention layer is followed by a fully connected feed-forward network (FFN), which applies a non-linear transformation to each position of the sequence input x.

For both SA and FFN, we apply residual connection [He+16] followed by layer normalization [BKH16]. Such a design can help the overall model to achieve faster convergence and better generalization.

In addition, it is necessary to incorporate position information for a sequence. Although it is feasible to encode such information using positional encoding, our experiments have shown that this will lead to bad performance in our case. Therefore, we adopt a more straightforward position-wise attention, by multiplying the representation at *pos* by a positional attention digit a_{pos}. In this way, the model can distinguish the importance of different relative locations in a sequence.

HiCE Architecture As illustrated in Figure 6.1, HiCE consists of two major layers: the *Context Encoder* and the *Multi-Context Aggregator*.

For each given word w_t and its K masked supporting context set $\mathbf{S}_t^K = \{s_{t,1}, s_{t,2}, ..., s_{t,K}\}$, a lower-level *Context Encoder* (E) takes each sentence $s_{t,k}$ as input, followed by position-wise attention and a self-attention encoding block, and outputs an encoded context embedding $E(s_{t,k})$. On top of it, a *Multi-Context Aggregator* combines multiple encoded contexts, i.e., $E(s_{t,1}), E(s_{t,2}), ..., E(s_{t,K})$, by another self-attention encoding block. Note that the order of

contexts can be arbitrary and should not influence the aggregation, we thus do not apply the position-wise attention in *Multi-Context Aggregator*.

Furthermore, the morphological features can be encoded using character-level CNN following [Kim+16], which can be concatenated with the output of *Multi-Context Aggregator*. Thus, our model can leverage both the contexts and morphological information to infer OOV embeddings.

6.3.3 Fast and Robust Adaptation with MAML

So far, we directly apply the learned neural regression function $F_{\hat{\theta}}$ trained on D_T to OOV words in D_N. This can be problematic when there exists some linguistic and semantic gap between D_T and D_N. For example, words with the same form but in different domains [SLS18] or at different times [HLJ16] can have different semantic meanings. Therefore, to further improve the performance, we aim to adapt the learned neural regression function $F_{\hat{\theta}}$ from D_T to the new corpus D_N. A naïve way to do so is directly fine-tuning the model on D_N. However, in most cases, the new corpus D_N does not have enough data compared to D_T, and thus directly fine-tuning on insufficient data can be sub-optimal and prone to overfitting.

To address this issue, we adopt Model Agnostic Meta-Learning (MAML) [FAL17] to achieve fast and robust adaption. Instead of simply fine-tuning $F_{\hat{\theta}}$ on D_N, MAML provides a way of learning to fine-tune. That is, the model is firstly trained on D_T to get a more promising initialization, based on which fine-tuning the model on D_N with just a few examples could generalize well.

More specifically, in each training episode, we first conduct gradient descent using sufficient data in D_T to learn an updated weight θ^*. For simplification, we use $\mathcal{L}$ to denote the loss function of our objective function (6.1). The update process is as:

$$\theta^* = \theta - \alpha \nabla_\theta \mathcal{L}_{D_T}(\theta).$$

We then treat θ^* as an initialized weight to optimize θ on the limited data in D_N. The final update in each training episode can be written as follows.

$$\begin{aligned}\theta' &= \theta - \beta\nabla_\theta\mathcal{L}_{D_N}(\theta^*) \\ &= \theta - \beta\nabla_\theta\mathcal{L}_{D_N}(\theta - \alpha\nabla_\theta\mathcal{L}_{D_T}(\theta)),\end{aligned} \tag{6.2}$$

where both α and β are hyper-parameters of two-stage learning rate. The above optimization can be conducted with stochastic gradient descent (SGD). In this way, the knowledge learned from D_T can provide a good initial representation that can be effectively fine-tuned by a few examples in D_N, and thus achieve fast and robust adaptation.

Note that the technique presented here is a simplified variant of the original MAML, which considers more than just two tasks compared to our case, i.e., a source task (D_T) and a target task (D_N). If we require to train embeddings for multiple domains simultaneously, we can also extend our approach to deal with multiple D_T and D_N.

6.4 Experiments

In this section, we present two types of experiments to evaluate the effectiveness of the proposed HiCE model. One is an intrinsic evaluation on a benchmark dataset, and the other is an extrinsic evaluation on two downstream tasks: (1) named entity recognition and (2) part-of-speech tagging.

6.4.1 Experimental Settings

As aforementioned, our approach assumes an initial embedding T trained on an existing corpus D_T. As all the baseline models learn embedding from Wikipedia, we train HiCE on WikiText-103 [Mer+17] with the initial embedding provided by Herbelot and Baroni [HB17][1].

[1] clic.cimec.unitn.it/~aurelie.herbelot/wiki_all.model.tar.gz

WikiText-103 contains 103 million words extracted from a selected set of articles. From WikiText-103, we select words with an occurrence count larger than 16 as training words. Then, we collect the masked supporting contexts set S_t for each training word w_t with its oracle embedding T_{w_t}, and split the collected words into a training set and a validation set. We then train the HiCE model[2] in the previous introduced episode based K-shot learning setting, and select the best hyper-parameters and model using the validation set. After we obtain the trained HiCE model, we can either directly use it to infer the embedding vectors for OOV words in new corpus D_N, or conduct adaptation on D_N using MAML algorithm as shown in Eq. (6.2).

Baseline Methods We compare HiCE with the following baseline models for learning OOV word embeddings.

1. **Word2Vec:** The local updating algorithm of Word2Vec. The model employs the 'Skip-gram' update to learn a new word embedding by predicting its context word vectors. We implement this baseline model with gensim[3].

2. **FastText:** FastText is a morphological embedding algorithm that can handle OOV by summing n-gram embeddings. To make fair comparison, we train FastText on WikiText-103, and directly use it to infer the embeddings of OOV words in new datasets. We again use the implementation in gensim[3].

3. **Additive:** Additive model [LMB17] is a purely non-parametric algorithm that averages the word embeddings of the masked supporting contexts S_t. Specifically:

$$e_t^{additive} = \frac{1}{|S_t|} \sum\nolimits_{c \in S_t} \frac{1}{|c|} \sum\nolimits_{w \in c} e_w.$$

Also, this approach can be augmented by removing the stop words beforehand.

[2] github.com/acbull/HiCE

[3] radimrehurek.com/gensim/

4. **nonce2vec:** This algorithm [HB17] is a modification of original gensim Word2Vec implementation, augmented by a better initialization of additive vector, higher learning rates and large context window, etc. We directly used their open-source implementation[4].

5. **à la carte:** This algorithm [Kho+18] is based on an additive model, followed by a linear transformation A that can be learned through an auxiliary regression task. Specifically:

$$e_t^{\grave{a}\ la\ carte} = \frac{A}{|S_t|} \sum\nolimits_{c \in S_t} \sum\nolimits_{w \in c} A e_w^{additive}$$

We conduct experiments by using their open-source implementation[5].

6.4.2 Intrinsic Evaluation: Evaluate OOV Embeddings on the Chimera Benchmark

First, we evaluate HiCE on Chimera [LMB17], a widely used benchmark dataset for evaluating word embedding for OOV words.

Dataset The Chimera dataset simulates the situation when an embedding model faces an OOV word in a real-world application. For each OOV word (denoted as "chimera"), a few example sentences (2, 4, or 6) are provided. The dataset also provides a set of probing words and the human-annotated similarity between the probing words and the OOV words. To evaluate the performance of a learned embedding, Spearman correlation is used in [LMB17] to measure the agreement between the human annotations and the machine-generated results.

[4] github.com/minimalparts/nonce2vec

[5] github.com/NLPrinceton/ALaCarte

Methods	2-shot	4-shot	6-shot
Word2vec	0.1459	0.2457	0.2498
FastText	0.1775	0.1738	0.1294
Additive	0.3627	0.3701	0.3595
nonce2vec	0.3320	0.3668	0.3890
à la carte	0.3634	0.3844	0.3941
HiCE w/o Morph	0.3710	0.3872	0.4277
HiCE + Morph	**0.3796**	0.3916	0.4253
HiCE + Morph + Fine-tune	0.1403	0.1837	0.3145
HiCE + Morph + MAML	0.3781	**0.4053**	**0.4307**
Oracle Embedding	0.4160	0.4381	0.4427

Table 6.1: Performance on the Chimera benchmark dataset with different numbers of context sentences, which is measured by Spearman correlation. Baseline results are from the corresponding papers.

Experimental Results Table 6.1 lists the performance of HiCE and baselines with different numbers of context sentences. In particular, our method (HiCE+Morph+MAML)[6] achieves the best performance among all the other baseline methods under most settings. Compared with the current state-of-the-art method, *à la carte*, the relative improvements (i.e., the performance difference divided by the baseline performance) of HiCE are 4.0%, 5.4% and 9.3% in terms of 2,4,6-shot learning, respectively. We also compare our results with that of the oracle embedding, which is the embeddings trained from D_T, and used as ground-truth to train HiCE. This results can be regarded as an upper bound. As is shown, when the number of context sentences (K) is relatively large (i.e., $K = 6$), the performance of HiCE is on a par with the upper bound (Oracle Embedding) and the relative performance difference is merely 2.7%. This indicates the significance of using an advanced aggregation model.

Furthermore, we conduct an ablation study to analyze the effect of morphological features. By comparing HiCE with and without Morph, we can see that morphological features are helpful when the number of context sentences is relatively small (i.e., 2 and 4

[6]Unless other stated, HiCE refers to HiCE + Morph + MAML.

shot). This is because morphological information does not rely on context sentences, and can give a good estimation when contexts are limited. However, in 6-shot setting, their performance does not differ significantly.

In addition, we analyze the effect of MAML by comparing HiCE with and without MAML. We can see that adapting with MAML can improve the performance when the number of context sentences is relatively large (i.e., 4 and 6 shot), as it can mitigate the semantic gap between source corpus D_T and target corpus D_N, which makes the model better capture the context semantics in the target corpus. Also we evaluate the effect of MAML by comparing it with fine-tuning. The results show that directly fine-tuning on target corpus can lead to extremely bad performance, due to the insufficiency of data. On the contrary, adapting with MAML can leverage the source corpus's information as regularization to avoid over-fitting.

Methods	Named Entity Recognition (F1-score)		POS Tagging (Acc)
	Rare-NER	Bio-NER	Twitter POS
Word2vec	0.1862	0.7205	0.7649
FastText	0.1981	0.7241	0.8116
Additive	0.2021	0.7034	0.7576
nonce2vec	0.2096	0.7289	0.7734
à la carte	0.2153	0.7423	0.7883
HiCE w/o Morph	0.2394	0.7486	0.8194
HiCE + Morph	0.2375	0.7522	0.8227
HiCE + Morph + MAML	**0.2419**	**0.7636**	**0.8286**

Table 6.2: Performance on Named Entity Recognition and Part-of-Speech Tagging tasks. All methods are evaluated on test data containing OOV words. Results demonstrate that the proposed approach, HiCE + Morph + MAML, improves the downstream model by learning better representations for OOV words.

6.4.3 Extrinsic Evaluation: Evaluate OOV Embeddings on Downstream Tasks

To illustrate the effectiveness of our proposed method in dealing with OOV words, we evaluate the resulted embedding on two downstream tasks: (1) named entity recognition (NER) and (2) part-of-speech (POS) tagging.

Named Entity Recognition NER is a semantic task with a goal to extract named entities from a sentence. Recent approaches for NER take word embedding as input and leverage its semantic information to annotate named entities. Therefore, a high-quality word embedding has a great impact on the NER system. We consider the following two corpora, which contain abundant OOV words, to mimic the real situation of OOV problems.

1. Rare-NER: This NER dataset [Der+17] focus on unusual, previously-unseen entities in the context of emerging discussions, which are mostly OOV words.

2. Bio-NER: The JNLPBA 2004 Bio-entity recognition dataset [CK04] focuses on technical terms in the biology domain, which also contain many OOV words.

Both datasets use entity-level F1-score as an evaluation metric. We use the WikiText-103 as D_T, and these datasets as D_N. We select all the OOV words in the dataset and extract their context sentences. Then, we train different versions of OOV embeddigns based on the proposed approaches and the baseline models. Finally, the inferred embedding is used in an NER system based on the Bi-LSTM-CRF [Lam+16] architecture to predict named entities on the test set. We posit a higher-quality OOV embedding results in better downstream task performance.

As we mainly focus on the quality of OOV word embeddings, we construct the test set by selecting sentences which have at least one OOV word. In this way, the test performance will largely depend on the quality of the OOV word embeddings. After the pre-processing, Rare-NER dataset contains 6,445 OOV words and 247 test sentences, while Bio-NER contains 11,748 OOV words and 2,181 test sentences. Therefore, Rare-NER has a high ratio of OOV words per sentence.

Part-of-Speech Tagging Besides NER, we evaluate the syntactic information encoded in HiCE through a lens of part-of-speech (POS) tagging, which is a standard task with a goal to identify which grammatical group a word belongs to. We consider the Twitter social media POS dataset [Rit+11], which contains many OOV entities. The dataset is

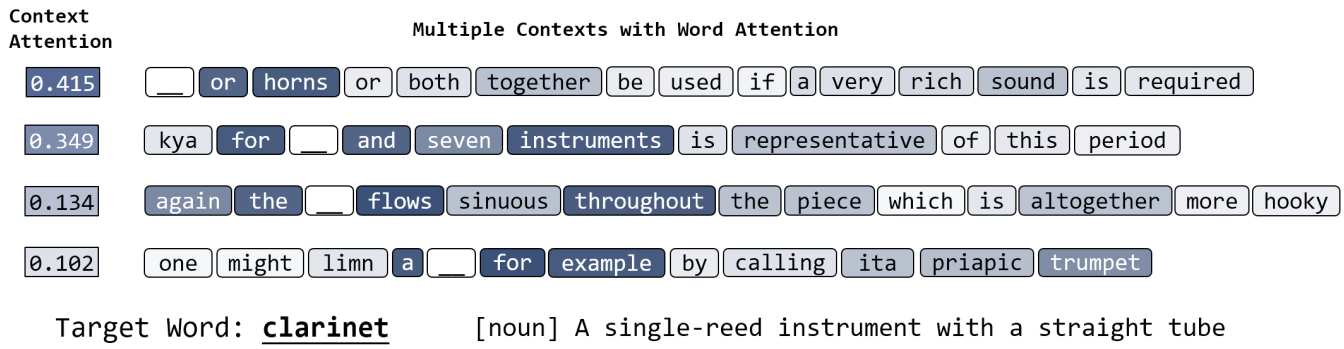


Figure 6.2: Visualization of attention distribution over words and contexts.

comprised of 15,971 English sentences collected from Twitter in 2011. Each token is tagged manually into 48 grammatical groups, consisting of Penn Tree Bank Tag set and several Twitter-specific classes. The performance of a tagging system is measured by accuracy. Similar to the previous setting, we use different updating algorithms to learn the embedding of OOV words in this dataset, and show different test accuracy results given by learned Bi-LSTM-CRF tagger. The dataset contains 1,256 OOV words and 282 test sentences.

OOV	Contexts	Methods	Top-5 similar words (via cosine similarity)
scooter	We all need vehicles like bmw c1 scooter that allow more social interaction while using them ...	Additive FastText **HiCE**	the, and, to, of, which cooter, pooter, footer, soter, sharpshooter cars, motorhomes, bmw, motorcoaches, microbus
cello	The instruments I am going to play in the band service are the euphonium and the cello ...	Additive FastText **HiCE**	the, and, to, of, in celli, cellos, ndegéocello, cellini, cella piano, orchestral, clarinet, virtuoso, violin
potato	It started with a green salad followed by a mixed grill with rice chips potato ...	Additive FastText **HiCE**	and, cocoyam, the, lychees, sapota patatoes, potamon, potash, potw, pozzato vegetables, cocoyam, potatoes, calamansi, sweetcorn
scarf	I wore my hat, scarf and gloves today i'm lucky I wore it ...	Additive FastText **HiCE**	the, and, trivalved, a, pealike scarfe, scarp, scar, scarpe, scarpa appliques, edgings, drawstring, bustier, dungarees

Table 6.3: For each OOV in Chimera benchmark, infer its embedding using different methods, then show top-5 words with similar embedding to the inferred embedding. HiCE can find words with most similar semantics.

Results Table 6.2 illustrates the results evaluated on the downstream tasks. HiCE outperforms the baselines in all the settings. Compared to the best baseline *à la carte*, the relative improvements are 12.4%, 2.9% and 5.1% for Rare-NER, Bio-NER, and Twitter POS, respectively. As aforementioned, the ratio of OOV words in Rare-NER is high. As a result, all the systems perform worse on Rare-NER than Bio-NER, while HiCE reaches the largest improvement than all the other baselines. Besides, our baseline embedding is trained on Wikipedia corpus (WikiText-103), which is quite different from the bio-medical texts and social media domain. The experiment demonstrates that HiCE trained on D_T is already able to leverage the general language knowledge which can be transferred through different domains, and adaptation with MAML can further reduce the domain gap and enhance the performance.

6.4.4 Qualitative Evaluation of HiCE

To illustrate how does HiCE extract and aggregate information from multiple context sentences, we visualize the attention weights over words and contexts. We demonstrate an example in Figure 6.2, where we choose four sentences in chimera dataset, with "clarinet" (a woodwind instrument) as the OOV word. From the attention weight over words, we can see that the HiCE puts high attention on words that are related to instruments, such as "horns", "instruments", "flows", etc. From the attention weight over contexts, we can see that HiCE assigns the fourth sentence the lowest context attention, in which the instrument-related word "trumpet" is distant from the target placeholder, making it harder to infer the meaning by this context. This shows HiCE indeed distinguishes important words and contexts from the uninformative ones.

Furthermore, we conduct a case study to show how well the inferred embedding for OOV words capture their semantic meaning. We randomly pick three OOV words with 6 context sentences in Chimera benchmark, use additive, fastText and HiCE to infer the embeddings. Next, we find the top-5 similar words with the highest cosine similarity. As is shown in Table 6.3, Additive method can only get embedding near to neutral words as

"the", "and", etc, but cannot capture the specific semantic of different words. FastText can find words with similar subwords, but representing totally different meaning. For example, for OOV "scooter" (a motor vehicle), FastText finds "cooter" as the most similar word, which looks similar in character-level, but means a river turtle actually. Our proposed HiCE however, can capture the true semantic meaning of the OOV words. For example, it finds "cars", "motorhomes" (all are vehicles) for "scooter", and finds "piano", "orchestral" (all are instruments) for "cello", etc. This case study shows that HiCE can truly infer a high-quality embedding for OOV words.

6.5 Summary

We studied the problem of learning accurate embedding for Out-Of-Vocabulary word and augment them to a per-trained embedding by only a few observations. We formulated the problem as a K-shot regression problem and proposed a hierarchical context encoder (HiCE) architecture that learns to predict the oracle OOV embedding by aggregating only K contexts and morphological features. We further adopt MAML for fast and robust adaptation to mitigate semantic gap between corpus. Experiments on both benchmark corpus and downstream tasks demonstrate the superiority of HiCE over existing approaches.

CHAPTER 7

Causal Representation Learning for Improving Multi-Task Generalization

Multi-Task Learning (MTL) is a powerful learning paradigm to improve generalization performance via knowledge sharing. However, existing studies find that MTL could sometimes hurt generalization, especially when two tasks are less correlated. One possible reason that hurts generalization is spurious correlation, i.e., some knowledge is spurious and not causally related to task labels, but the model could mistakenly utilize them and thus fail when such correlation changes. In MTL setup, there exist several unique challenges of spurious correlation. First, the risk of having non-causal knowledge is higher, as the shared MTL model needs to encode all knowledge from different tasks, and causal knowledge for one task could be potentially spurious to the other. Second, the confounder between task labels brings in a different type of spurious correlation to MTL. Given such label-label confounders, we theoretically and empirically show that MTL is prone to taking non-causal knowledge from other tasks. To solve this problem, we propose Multi-Task Causal Representation Learning (MT-CRL) framework. MT-CRL aims to represent multi-task knowledge via disentangled neural modules, and learn robust task-to-module routing graph weights via MTL-specific invariant regularization. Experiments show that MT-CRL could enhance MTL model's performance by 5.5% on average over Multi-MNIST, MovieLens, Taskonomy,

CityScape, and NYUv2, and show it could indeed alleviate spurious correlation problem.

7.1 Introduction

Multi-Task Learning (MTL), a learning paradigm [Car97; ZY18] aiming to train a single model for multiple tasks, is expected to benefit the overall generalization performance than single-task learning [MPR16; TJJ20] given the assumption that there exists some common knowledge to handle different tasks. However, recent studies observed that, when two tasks are less correlated, MTL could lead to even worse overall performance [PBS16; Zha+21b]. A line of works [Yu+20b; Wan+21b; Fif+21] resort performance drop to optimization challenge because conflicting tasks might compete for model capacity. However, both Standley, Zamir, Chen, Guibas, Malik, and Savarese [Sta+20] and our analysis in Section 7.3.2 show that, even with an over-parameterized model that achieves low MTL training loss, the final generalization performance could be worse than single-task learning. This finding motivates us to think about the following question: Are there any intrinsic problems in MTL that hurt generalization?

One widely studied issue that influences generalization is the spurious correlation problem [Gei+19; Gei+20], i.e., correlation that only existed in training datasets due to unobserved confounders [Lop16], but not causally correct. For example, as Beery, Horn, and Perona [BHP18] discussed, when we train an image classification model to identify cows with a biased dataset where cows mostly appear in pastures, the trained cow classification model could exploit the features of background (e.g., pastures) to make prediction. Thus, when we apply the classifier to another dataset where cows also appear in other locations such as farms or rivers, it will fail to generalize [NAN21].

When it comes to MTL setting, there exist several unique challenges to handle spurious correlation problem. **First, the risk of having non-causal features is higher**. Suppose each task has different sets of causal features. To train a single model for all these tasks, the shared representation should encode all required features. Consequently, the causal

features for one task could be potentially spurious to the other tasks, and such risk could be even higher with an increasing number of tasks. **Second, the confounder that leads to spurious correlation is different**. Instead of the standard confounders between feature and label, the nature of MTL brings in a unique type of confounders between task labels, e.g., correlation between tasks' labels could change in different distributions. For example, when we train a MTL model to solve both cow classification and scene recognition tasks, its encoder needs to capture both foreground and background information, and the spurious correlation between the two tasks in training set could mislead per-task model to utilize irrelevant information, e.g., use background to predict cow. Given such label-label confounders that are unique for MTL, we theoretically prove that MTL is prone to taking non-causal knowledge learned from other tasks. We then conduct empirical analysis to validate the hypothesis. In summary, we point out the unique challenges of spurious correlation in MTL setup, and show that it indeed influences multi-task generalization.

In light of the analysis, we try to solve the spurious correlation problem in MTL. Among all the knowledge learned in the shared representation layer through end-to-end training, an ideal MTL framework should learn to leverage only the causal knowledge to solve each task by identifying the correct causal structure. Following the recent advances that enable causal learning in an end-to-end learning model [Sch+21; Mit+21], we propose a Multi-Task Causal Representation Learning (MT-CRL) framework, aiming to represent the multi-task knowledge via a set of disentangled neural modules instead of a single encoder, and learn the task-to-module causal relationship jointly. We adopt de-correlation and sparsity regularization over popular Mixture-of-Expert (MoE) architecture [Sha+17]. The most critical and challenging step is to learn the causal graph in the MTL setup, which requires distinguishing the genuine causal correlation from spurious ones for all tasks. Motivated by the recent studies that invariance could lead to causality [Ahu+20; KY21], we propose to penalize the variance of gradients w.r.t. causal graph weights across different distributions. On a high level, this invariance regularization encourages the causal graph to assign higher weights to the modules that are consistently useful. In contrast, the modules encoding

spurious knowledge that cannot consistently achieve graph optimality are assigned lower weights and be discarded by task predictors.

We evaluate our method on existing MTL benchmarks, including Multi-MNIST, MovieLens, Taskonomy, CityScape, and NYUv2. For each dataset, to mimic distribution shifts, we adopt some attribute information given in the dataset, such as the released time of the movie or district of a building, to split train/valid/test datasets. The results show that MT-CRL could consistently enhance the MTL model's performance by 5.5% on average, and outperform both the MTL optimization and robust machine learning baselines. We also conduct case studies to show that MT-CRL indeed alleviate spurious correlation problem in MTL setup.

The key contributions of this paper are as follows:

1. We are the first to analyze spurious correlation problem in MTL setup, and point out several key challenges unique to MTL with theoretical and empirical analysis.

2. We propose MT-CRL with MTL-specific invariant regularizers to elleviate spurious correlation problem, and enhances the performance on several MTL benchmarks.

7.2 Related Work

Multi-Task Generalization. A deep neural model often requires a large number of training samples to generalize well [Aro+19; CG19]. To alleviate the sample sparsity problem, MTL could leverage more labeled data from multiple tasks [ZY18]. Most works studying multi-task generalization are based on a core assumption that the tasks are correlated. Earlier research directly define the task relatedness with statistical assumption [Bax00; BB08; LG19]. With the increasing focus on deep learning models, recent research decompose ground-truth MTL models into a shared representation and different task-specific layers from a hypothesis family [MPR16]. With such decomposition, Tripuraneni, Jordan, and Jin [TJJ20] and Du, Hu, Kakade, Lee, and Lei [Du+21] prove that a diverse set of tasks

could help learn more generalizable representation. Wu, Zhang, and Ré [WZR20] study how covariate shifts influence MTL generalization. Despite these findings, the core assumption of task relatedness might not be satisfied in many real-world applications [PBS16; Zha+21b], in which tasks could even conflict with each other to compete model capacity, and the generalization performance of MTL could be worse than single-task training.

To solve the task conflict problem, a number of MTL model architectures have utilized modular [Mis+16; Lu+17; RKR18; Ma+18; GLU20] or attention-based [LJD19; MRK19] design to enlarge model capacity while preserving information sharing. Our work is model-agnostic and could be applied to existing architectures to further solve the spurious feature problem. Another line of research alleviate task conflict during optimization. Some propose to balance the task weight via uncertainty estimation [KGC18], gradient norm [Che+18], convergence rate [LJD19], or pareto optimality [SK18]. Others directly modulate task gradients via dropping part of the conflict gradient [Che+20b] or project task's gradient onto other tasks' gradient surface [Yu+20b; Wan+21b]. Though these works successfully facilitate MTL model to converge easier, our analysis show that with spurious correlation, the MTL model with low training loss could still generalize bad. Therefore, our proposed MT-CRL that alleviates spurious correlation is orthogonal to these prior works, and could be combined to further improve overall performance.

Spurious Correlation Problem. Due to the selection bias [TE11; Gur+18] or unobserved confounding factors [Lop16], training datasets always contain spurious correlations between non-causal features and task labels, with which trained models often leverage non-causal knowledge and may fail to generalize Out-Of-Distribution (OOD) when such correlation changes [NAN21]. To solve the spurious correlation problem, some fairness research pre-define a set of non-causal features (e.g., gender and underrepresented identity) and then explicitly remove them from the learned representation [Zem+13; Gan+16; Wan+19a]. Another line of robust machine learning research does not assume to know spurious features, but regularize the model to perform equally well under different distri-

bution. Distributionally Robust Optimization (DRO) optimizes worst-case risk [Sag+20]. Invariant Causal Prediction (ICP) learns causal relations via invariance testing [PBM16]. Invariant Risk Minimization (IRM) forces the final predictor to be optimal across different domains [Arj+19]. Risk Extrapolation (REx) directly penalizes the variance of training risk in different domains [Kru+21]. Another line of work aim at learning causal representation [Sch+21], i.e., high-level variables representing different aspect of knowledge from raw data input. Most of these works try to recover disentangled causal generative mechanisms [Par+18; Ben+20; Liu+20; Mit+21]. Despite the extensive study of spurious correlation in single-task setting, few work discuss it for MTL models. This paper is the first to point out the unique challenges of spurious correlation in MTL setup.

7.3 Analyzing Spurious Correlation in MTL

To systematically analyze the spurious correlation problem in MTL, we first assume that data and task labels are generated by ground-truth causal mechanisms described in Suter, Miladinovic, Schölkopf, and Bauer [Sut+19]. We denote X as the variable of observed data, and each data is associated with K latent generative factors $\mathbb{F} = \{F_i\}_{i=1}^K$ representing different semantics of the data (e.g., color, shape, background of an image). We follow [Sch+21] to assume that the data X is generated by disentangled causal mechanisms $P(X|F_i)$, such that $P\big(X|\mathbb{F}\big) = \prod_{i=1}^K P\big(X|F_i\big)$.

As $\mathbb{F}$ represents high-level knowledge of the data, we could naturally define task label variable Y_t for task t as the cause of a subset of generative factors. We denote $\mathbb{F}_t^C$ as a subset of causal feature variables within $\mathbb{F}$ that are causally related to each task variable Y_t, and we could define $\mathbb{F}_t^S = \mathbb{F} \setminus \mathbb{F}_t^C$ as a subset of non-causal feature variables to task t, such that $P(Y_t|\mathbb{F}) = P(Y_t|\mathbb{F}_t^C)$. In other words, changing the values of any non-causal factors in $\mathbb{F}_t^S$ does not change the conditional distribution.

Note that the discussion so far is based on the assumption that the ground-truth causal generative model is known. In a real-world learning setting, however, we are only given

a supervised dataset (X, Y) without access to generative factors $\mathbb{F}$. To solve the task, a neural encoder $\Phi(\cdot)$ is required to extract representation $\mathbf{Z}$ from the data that encodes the information about the causal factors, on top of which a task predictor $f(\cdot)$ could predict the label.

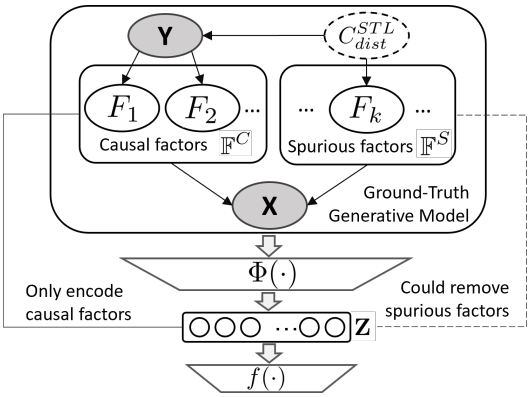


Figure 7.1: Spurious correlation in **Single-Task Learning** is mainly caused by factor-label confounders C_{dist}^{STL}. We could remove spurious factors $\mathbb{F}^S$ from representation Z.

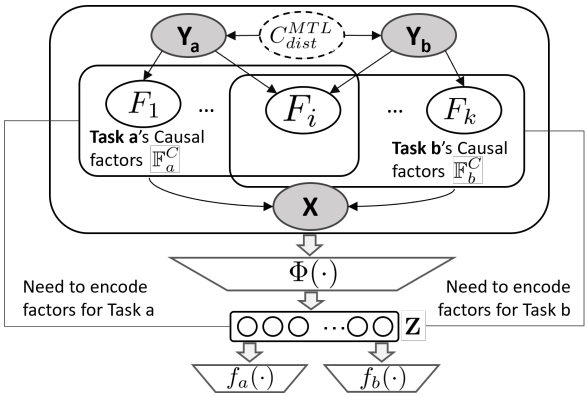


Figure 7.2: Spurious correlation in **Multi-Task Learning** could be caused by label-label confounders C_{dist}^{MTL}. Factors for both tasks $\mathbb{F}_a^C$ and $\mathbb{F}_b^C$ need to be encoded and potentially spurious.

7.3.1 Spurious Correlation Problem

Based on the ground-truth generative model, an ideal predictor for each task should only utilize the causal factors, and keep invariant to any intervention on non-causal factors. However, in real-world problems, it is hard to achieve an invariant predictor due to the spurious correlation issue due to unobserved confounders C_{dist} [Lop16]. Formally, confounders are variables that influence the two connected variables' correlation, and such correlation could change under different distribution (different value of C_{dist}), thus the model exploiting such spurious correlation will fail to generalize. Below we summarize the differences of spurious correlation problems for single-task and multi-task learning settings:

Single-Task Learning (STL). As illustrated in Figure 7.1, the label-factor confounders for single task learning C_{dist}^{STL} connects non-causal factors $F \in \mathbb{F}^S$ and task label Y, bringing in spurious correlation. For example, temperature could confound crime and ice cream

consumption. When the weather is hot, both crime rates and ice cream sales increase, but these two phenomena are not causally related. Based on the proof by Nagarajan, Andreassen, and Neyshabur [NAN21] and Khani and Liang [KL21], such spurious correlation could lead the model to use non-causal factors, and thus hurt generalization performance.

Multi-Task Learning (MTL). In the MTL setting, there exist several unique challenges to handle spurious correlation. First, the risk of having non-causal features is higher. As is illustrated in Figure 7.2, the shared encoder Φ needs to encode all the factors causally related to each task in the representation Z. Therefore, for each task, all non-overlapping factors from other tasks could be potentially spurious. Second, besides the standard label-factor confounders C_{dist}^{STL} for each single task introduced above, we define label-label confounders C_{dist}^{MTL} connecting multiple tasks' label $\{Y\}$. Such confounder is unique to MTL setting.

As an example, consider two binary classification tasks, with Y_a and Y_b as variables from $\{\pm 1\}$ for task label. The two labels' correlation $P(Y_a = Y_b) = m_C$ could change with different confounder $C_{dist}^{MTL} = C$. We assume the two tasks have non-overlapping factors F_a and F_b drawn from Gaussian distribution. We then show MTL model with both two factors as input will utilize non-causal factors:

Proposition 1. *Given $m_C \neq 0.5$, the Bayes Optimal per-task classifier has non-zero weights to non-causal factor. Given $m_C = 0.5$ and limited training dataset, the trained per-task classifier will assign non-zero weights to non-causal factor as noise.*

Therefore, in this linear classification example, when we deploy the trained model to a new distribution with changed label-label confounder C_{dist}^{MTL}, the model trained by MTL that utilizes non-causal factors generalize relatively worse. On the contrary, the model trained by STL don't need to encode all causal factors from two tasks. Assuming there is no task-label confounder C_{dist}^{STL} in each task's dataset, the trained model could remove non-causal factors from representation.

7.3.2 Empirical Experiments

In the following, we conduct experiments to validate the claims. As there is no existing MTL datasets specifically designed to analyze spurious correlation problem, we construct synthetic Multi-SEM [RRR21] and Multi-MNIST [HK16] datasets with known causal structure to study whether the model trained by MTL indeed exploits more non-causal factors, and how the spurious correlation influences multi-task generalization.

Spurious Score. As we know the ground-truth causal structure for the two datasets, we could quantify how much a model utilizes the non-causal factors. Following the gradient saliency map proposed by Simonyan, Vedaldi, and Zisserman [SVZ14], we calculate the average absolute gradients w.r.t each factor as $Grad(F) = \sum_{(x(\mathbb{F}),y)\in D} \left| \frac{\partial \left(f(\Phi(x))[y] \right)}{\partial F} \right|$, which measures how much a model leverage this factor to make prediction. We then define the spurious score ρ_{spur} as the proportion of average gradients over non-causal feature $\rho_{spur} = \frac{\sum_{F\in\mathbb{F}^S} Grad(F)}{\sum_{F\in\mathbb{F}} Grad(F)}$.

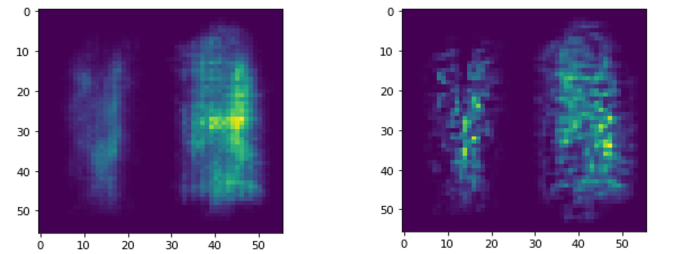


Figure 7.3: The gradient saliency map of right-side digit classifier. The model trained by MTL exploits left pixels (spurious) more.

	Multi-SEM		**Multi-MNIST**	
	STL	MTL	STL	MTL
Acc_{train}	0.931	0.936	0.981	0.987
Acc_{val}	0.906	0.882	0.874	0.846
ρ_{spur}	0.128	0.246	0.261	0.328

Table 7.1: Empirical results of multi-task (MTL) and single-task learning (STL) model on synthetic datasets with changing C_{dist}^{MTL}.

Empirical Results. We train a shared-bottom model via Multi-task learning (MTL) and single-task learning (STL) over the two datasets and report both the training and test accuracy with spurious ratio ρ_{spur} in Table 7.1. As illustrated, the test accuracies of MTL for both Multi-SEM and Multi-MNIST datasets are both worse than STL. The training accuracies of MTL are very similar to STL, meaning that the performance drop is not due

to the optimization difficulty that many previous works try to address. The spurious ratio ρ_{spur} of MTL is much higher than the STL, which means that it exploits more non-causal factors. To give a more straightforward illustration, we plot the gradient saliency map of the right-side digit classifier for Multi-MNIST in Figure 7.3. The model trained by MTL utilizes more left-side pixels, which are non-causal to the final prediction. These findings support our hypothesis that with spurious correlation caused by label-label confounder C_{dist}^{MTL}, models trained by MTL is more prone to leverage non-causal knowledge than STL, and thus influence generalization performance.

7.4 Methodology

Based on the previous analysis of the spurious correlation problem in MTL, we now introduce a Multi-Task Causal Representation Learning (MT-CRL) framework with the goal that the per-task predictor only leverages the causal knowledge instead of spurious correlation. The high-level idea of the framework is to reconstruct the ground-truth causal mechanisms introduced in section 7.3 through end-to-end representation learning. To accomplish this goal, the framework aims to 1) model multi-task knowledge via a set of disentangled neural modules; 2) learn the task-to-module causal graph that is optimal across different distributions. With the correct causal graph as routing layer, per-task predictor only utilizes outputs from causally-related modules, thus alleviating the spurious correlation problem. We introduce the two crucial designs as follows.

7.4.1 Modelling via Disentangled Neural Modules

In order to alleviate spurious correlation, an ideal MTL model should learn the multi-task knowledge in the shared representation while identifying which part of the knowledge is causally related to each task. However, directly conducting causal discovery is impossible if all the knowledge is fused in a single shared encoder. Thus, we seek to adopt a modularized architecture in which each module encodes disentangled knowledge, and thus enable modeling

causal relationship between task and modules. We adopt Multi-gate Mixture-of-Experts (MMoE) [Ma+18], a variant of MoE [Sha+17] architecture tailored for MTL setting, as our underlying model. Specifically, we have K different neural modules as shared encoders $\Phi = \big[\Phi_i(\cdot)\big]_{i=1}^{K}$. Given a batch of input data $\mathbf{X} = \{x_n\}_{n=1}^{B}$ with batch size B, we extract k representations via different neural modules, i.e., $\mathbf{Z}_i = \Phi_i(X) \in \mathbb{R}^{B\times d}$. Based on sparsity assumption of the causal mechanisms [Par+18; Ben+20; Lac+21], only a few modules should be causally related to each task. Therefore, on top of the learned neural modules, we learn a task-to-module routing graph, aiming to estimate which module is causally related to each task. We model the bipartite adjacency (a.k.a. bi-adjacency) matrix $A = \text{sigmoid}(\theta) \in [0,1]^{T\times K}$ by applying sigmoid over a learnable parameter θ to enforce the range constraint. Note that original MMoE adopts softmax to get gate vector, which encourages only a small portion of modules being utilized for each task. Our graph modelling allows multiple modules utilized for each task. With the correct graph weights A as routing layer, we could utilize only the causally related modules and make predictions with per-task predictor $f_t(\cdot)$ as $\hat{Y}_t(\mathbf{X}) = f_t\big(\sum_i A_{t,i} \cdot \Phi_i(\mathbf{X})\big)$.

Disentangling Modules. One of the main properties of the causal mechanisms we introduced in section 7.3 is disentanglement, such that each factor represents a different view of the data, and changing the value of one factor does not influence the others. If without explicit constraints, the learned modules' outputs could still be correlated and hinder the causal structure learning. Therefore, we need to add regularization to disentangle these modules during training.

Most existing disentangled representation learning methods are under the generative modeling framework, e.g. VAE [Hig+17] or GAN [Che+16]. However, Locatello, Bauer, Lucic, Rätsch, Gelly, Schölkopf, and Bachem [Loc+19] argues that without explicit supervision, it is hard for generative models to learn correct disentangled factors. We therefore only borrow the regularization methods utilized in existing generative disentangled representation works [Che+15b; Cog+16] to directly penalize the correlation of learned modules.

Specifically, we regularize the in-batch Pearson correlation $\rho(\mathbf{Z}_i, \mathbf{Z}_j)$ between every pair of output dimensions from different representation matrices $\mathbf{Z}_i$ and $\mathbf{Z}_j$, as:

$$\rho(\mathbf{Z}_i, \mathbf{Z}_j) = \frac{Cov(\mathbf{Z}_i, \mathbf{Z}_j)}{\sqrt{Cov(\mathbf{Z}_i, \mathbf{Z}_i)}\sqrt{Cov(\mathbf{Z}_j, \mathbf{Z}_j)}}, \text{ where } Cov(\mathbf{Z}_i, \mathbf{Z}_j) = \big[\mathbf{Z}_i - \overline{\mathbf{Z}_i}\ \big]^T \big[\mathbf{Z}_j - \overline{\mathbf{Z}_j}\ \big]. \tag{7.1}$$

By minimizing the Frobenius norm of the correlation matrix ρ for every two different representation pairs, we could enforce the encoder Φ to extract disentangled representations.

$$\mathcal{L}_{decor}(\Phi) = \lambda_{decor} \cdot \sum_{i=1}^{k} \sum_{j=i+1}^{k} \Big\| \rho\big(\Phi_i(X), \Phi_j(X)\big) \Big\|_F^2. \tag{7.2}$$

Task-to-Module Graph Regularization. Based on sparsity assumption of the causal mechanisms [Par+18; Ben+20; Lac+21], each task is causally related to only a few modules. To learn the graph structure, existing works [Zhe+18; Ng+19; Lac+20] propose to to fit structural equation model (SEM) with sparsity regularization over the graph weights. We adopt a similar sparse regularization with an entropy balancing term [Hai12] over the bi-adjacency matrix A weights of the task-to-module routing graph:

$$\mathcal{L}_{graph}(A) = \lambda_{sps} \cdot ||A||_1 - \lambda_{bal} \cdot \text{Entropy}\Big(\frac{\sum_t A_{t,*}}{\sum_{t,i} A_{t,i}}\Big). \tag{7.3}$$

Note that the entropy term aims at keeping the causal weights for each module i summing over all the tasks to be balanced. This could help avoid degenerate solutions in which only a few modules are utilized. By combining the two regularizations in Eq.(7.2) and Eq.(7.3) with per-task supervised risk term $R_t\big(\Phi, A_t, f_t\big) = \sum_{(\mathbf{X}, \mathbf{Y}_t) \in \mathcal{D}} L_t\big(\hat{Y}_t(\mathbf{X}), \mathbf{Y}_t\big)$, we get the regularized loss as:

$$\tilde{\mathcal{L}}(\Phi, A, f) = \sum_{t \in \mathcal{T}} R_t\big(\Phi, A_t, f_t\big) + \mathcal{L}_{decor}(\Phi) + \mathcal{L}_{graph}(A). \tag{7.4}$$

7.4.2 Causal Learning via Graph-Invariant Regularization

It is critical and challenging to learn the correct causal graph, which requires distinguishing the true causal correlation from spurious ones. Motivated by the recent studies of robust machine learning that a predictor invariant to multiple distributions could learn causal correlation [Ahu+20; KY21], we assume the true causal relationship to be optimal across different distributions. To do so, we assume to have access to multiple slices of datasets collected from different environments $e \in \mathcal{E}$ in which the confounder C_{dist}^{MTL} that controls task correlation might change. For example, one natural choice is to consider train/valid dataset split (the setting we utilize in experiment), or assume the training set is split into multiple slices with different attributes. We desire the task-to-module graph weights A and per-task predictor f_t to be optimal across all environments $e \in \mathcal{E}$. Formally, we aim to solve the following bi-level optimization problem:

$$\min_{\Phi, A, f} \tilde{\mathcal{L}}(\Phi, A, f) \quad \text{s.t.} \quad A_t, f_t \in \arg\min_{A,f} R_t^e\big(\Phi, A, f\big), \forall\, t \in \mathcal{T}, e \in \mathcal{E}. \tag{7.5}$$

where R_t^e denotes the risk over data slice in environment e. This optimization problem could be regarded as a multi-task version of IRM. Based on Theorem 9 described in Ahuja, Shanmugam, Varshney, and Dhurandhar [Ahu+20], by enforcing invariance over a sufficient number of environments that exhibit distribution shifts (i.e., changes of confounder C_{dist}^{MTL}), per-task predictors should only utilize modules that are consistently helpful to the task, and assign zero weights to modules that encode non-causal factors to the task, and thus alleviate spurious correlation and help out-of-distribution generalization. Even if all data are sampled from the same distribution and there are no distribution shifts, invariance could also help eliminate noisy correlation due to the limited training dataset and help in-distribution generalization.

Invariant Optimality of Task-to-Module Graph for MTL. As discussed in IRM, the bi-leveled optimization problem in Eq.(7.5) is highly intractable, especially with complex

and non-linear Φ. To implement a practical optimization objective, IRM proposes to softly regularize the gradient of the task-predictor at different environments to enforce it to be optimal:

$$\min_{\Phi,A,f} \Big(\tilde{\mathcal{L}}(\Phi, A, f) + \sum_{t\in\mathcal{T}}\sum_{e\in\mathcal{E}} \Big\|\nabla_{A=A_t, f=f_t} R_t^e\big(\Phi, A, f\big)\Big\|^2\Big). \tag{7.6}$$

However, as is discussed in IRM paper, if the complexity of a task-predictor f is much larger than the number of environments, it could learn an over-fitted solution that makes gradient zero but does not achieve invariance. IRM adopts a fixed all-one vector as predictor to reduce complexity. This approach **is not applicable to MTL setup**, as the optimal task-predictors f_t^* for different task t could be very distinctive and complex, and we cannot use a fixed uniform predictor for all tasks.

To strike a balance between invariance and complexity of multi-task predictors, we propose only to regularize the gradient of the task-to-module routing graph while assuming the complex predictor f_t for each task is fixed at each iteration. We call this modification as **Graph-Invariant Risk Minimization (G-IRM)**, which is designed specifically to MTL setup:

$$\min_{\Phi,A,f} \Big(\tilde{\mathcal{L}}(\Phi, A, f) + \lambda_{G\text{-}IRM} \cdot \mathcal{L}_{G\text{-}IRM}(\Phi, A|f)\Big). \tag{7.7}$$

By adopting the similar gradient penalty term as adopted in IRM, we define $\mathcal{L}^{Norm}_{G\text{-}IRM}$ as:

$$\mathcal{L}^{Norm}_{G\text{-}IRM}(\Phi, A|f) = \sum_{t\in\mathcal{T}}\sum_{e\in\mathcal{E}} \Big\|\nabla_{A=A_t} R_t^e\big(\Phi, A, f_t\big)\Big\|^2. \tag{7.8}$$

As we assume f_t is fixed for invariance regularization term $\mathcal{L}^{Norm}_{G\text{-}IRM}$, we only calculate gradient and optimize for Φ and A, but not updating f_t. This could avoid the over-parametrized predictor f_t finding a trivial solution to achieve zero gradients instead of learning the correct causal correlation. Similar trick is utilized in [Ahm+21]. Note that the gradient w.r.t each graph weight means whether a module could help reduce the risk for this task. Therefore,

by penalizing the invariance regularization, the modules containing non-causal factors will be assigned zero weights.

In the experiments, we observe that at the early optimization stage, the model has non-zero gradients for all parameters, including the graph weights, thus directly regularizing the gradient norm might influence the optimization. Therefore, we propose a modified version of gradient regularization $\mathcal{L}^{Var}_{G\text{-}IRM}$ that penalizes the variance of the task-to-module graph's gradient on different environments:

$$\mathcal{L}^{Var}_{G\text{-}IRM}(\Phi, A|f) = \sum_{t\in\mathcal{T}} \sum_{e\in\mathcal{E}} \frac{1}{|\mathcal{E}|} \left\| \nabla_{A=A_t} R^e_t\big(\Phi, A, f_t\big) - \mathrm{Avg}_e\Big(\nabla_{A=A_t} R^e_t\Big) \right\|^2. \tag{7.9}$$

By minimizing $\mathcal{L}^{Var}_{G\text{-}IRM}$, we force all the learned modules to have similar gradients across different environments, and not overfit only to some of the environments. It still allows some modules to have non-zero gradients as long as it's the same across environments, and relies on loss term $\tilde{\mathcal{L}}$ to update these weights, while $\mathcal{L}^{Norm}_{G\text{-}IRM}$ forces all gradient to be zero. Therefore, $\mathcal{L}^{Var}_{G\text{-}IRM}$ is a loose regularization that not influences the overall optimization too much. It shares similar intuition of REx [Kru+21] that penalizes risk variance, while $\mathcal{L}^{Var}_{G\text{-}IRM}$ penalize gradient variance.

7.5 Experiment

In this section, we evaluate whether MT-CRL could benefit the performance of MTL models on existing benchmark datasets, and study whether it could indeed alleviate spurious correlation.

7.5.1 Experimental Setup

One key ingredient of our MT-CRL is to achieve the optimality of causal graph over different distributions. However, we might not access multiple environmental labels in most real-world multi-task learning datasets. Therefore, we adopt a more realistic setup, such

that we only assume to have a single validation set that contains unknown distribution shifts (i.e. change of confounder C_{dist}^{MTL}) compared to the training dataset. We thus could utilize training and valid sets as two environments to calculate invariance regularization, while we only utilize the training set to calculate task loss to avoid the task predictor overfits. Note that in this way, our method could get access to the label information in the validation set. To avoid the possibility that the performance improvement is brought by additional label, for all the other baseline methods, we also add the validation data into the training set to calculate task loss and learn MTL model.

Dataset. We choose five widely-used real-world MTL benchmark datasets, i.e., Multi-MNIST [Sun19], MovieLens [HK16], Tasknomy [Zam+18], NYUv2 [Sil+12] and CityScape [Cor+16], and try to determine train/valid/test split such that there exist distribution shifts between these sets. Note that except NYUv2, our data split is the same as the default split settings of these datasets, which also try to test model's capacity to generalize across domains.

Baselines. As MT-CRL is a regularization framework built upon modular MTL architecture (in this paper we choose MMoE as instantiation, but it can be applied to other modular networks), we mainly compare with two gradient-based multi-task optimization baselines: **PCGrad** [Yu+20b] and **GradVac** [Wan+21b]. We also compare with two domain generalization baselines: **IRM** [Ahu+20] and **DANN** [Gan+16]. For IRM we adopt different per-task predictors instead of all-one vector to adapt MTL setup, and calculate penalty via Eq. (7.6).

Hyper-Parameter Selection. For a fair comparison, all methods are based on the same MMoE architecture. Our methods contain a lot of hyper-parameters, including some model specific ones such as number of modules (K) and regularization specific ones. To avoid the case that performance improvement is caused by extensive hyper-parameter tuning, we mainly search optimal model hyper-parameter on Vanilla MTL setting, and use for all baselines. For regularization specific parameters, we take Multi-MNIST, the simplest

Methods	Multi-MNIST	MovieLens	Taskonomy	CityScape	NYUv2	Avg.
Vanilla MTL	(—baseline to calculate relative improvement—)					
Single-Task Learning	+3.3%	+0.2%	-2.5%	-2.4%	-12.2%	-2.7%
MTL + PCGrad	+4.5%	+0.2%	+3.1%	+2.1%	+7.4%	+3.5%
MTL + GradVac	+4.6%	+0.3%	+3.5%	+2.1%	+7.2%	+3.5%
MTL + DANN	+4.1%	+0.4%	+1.2%	+0.3%	-0.4%	+1.1%
MTL + IRM	+5.0%	+0.4%	+1.1%	+0.6%	-0.1%	+1.4%
MT-CRL w/o $\mathcal{L}_{G\text{-}IRM}$	+5.9%	+0.2%	+3.2%	+1.5%	+4.3%	+3.0%
MT-CRL with $\mathcal{L}^{Norm}_{G\text{-}IRM}$	+7.8%	+1.0%	+6.5%	**+2.9%**	+8.0%	+5.2%
MT-CRL with $\mathcal{L}^{Var}_{G\text{-}IRM}$	**+8.1%**	**+1.1%**	**+7.1%**	+2.8%	**+8.2%**	**+5.5%**

Table 7.2: Relative Performance improvement of different multi-task learning (MTL) strategies compared to vanilla MTL baseline.

dataset among the testbed, to find a optimal combination, and use for all other datasets.

7.5.2 Experiment Results

As each task has a different evaluation metric and cannot be directly compared, we calculate the relative performance improvement of each method compared to vanilla MTL, and then average the relative improvement for all tasks of each dataset. As summarized in Table 7.2, the average improvement of MT-CRL with $\mathcal{L}^{Var}_{G\text{-}IRM}$ is 5.5%, significantly higher than all other baseline methods. The most critical step of MT-CRL is to learn correct causal graph. We therefore report MT-CRL with different invariance regularization. As is shown in the last block, $\mathcal{L}^{Var}_{G\text{-}IRM}$ achieve better results for most datasets than $\mathcal{L}^{Norm}_{G\text{-}IRM}$, while removing the invariance regularization could significantly drop the relative performance. Compared to IRM which calculate gradient and update per-task predictors, MT-CRL uses disentangled modules and G-IRM to avoid overfitting to achieve invariance. Results show that for datasets with large amount of tasks, e.g., Taskonomy and NYUv2, MT-CRL significantly outperform IRM, showing the modification is more suitable for MTL setup.

Ablation Studies. We then study the effectiveness of the other two components in MT-CRL, i.e., disentangled and graph regularization. We mainly report the ablation studies on Multi-MNIST in table 7.3 as it's relatively small so that we could quickly get the results

Disentangled Regularization		Graph Regularization		Multi-MNIST
$\mathcal{L}_{decor}$	$\mathcal{L}_{\beta\text{-VAE}}$	$\mathcal{L}_{sps}$	$\mathcal{L}_{bal}$	Accuracy
✓	✗	✓	✓	0.915 ± 0.018
✗	✓	✓	✓	0.896 ± 0.024
✗	✗	✓	✓	0.882 ± 0.020
✓	✗	✗	✗	0.891 ± 0.016
✓	✗	✗	✓	0.903 ± 0.017
✓	✗	✓	✗	0.908 ± 0.021

Table 7.3: **Ablation Studies** of disentangled and Graph regularization components in MT-CRL, evaluated on Multi-MNIST dataset.

of all combinations.

For disentangled regularization, after removing $\mathcal{L}_{decor}$, the performance drops from 0.915 to 0.882, which fits our discussion that we cannot conduct causal learning over entangled modules. We also explore one classical generative disentangled representation method, i.e., β-VAE. As shown in the table, the results of using β-VAE are 0.896, lower than our utilized decorrelation regularization.. We hypothesize that this is probably because not all generative factors are useful for downstream tasks. Generative objectives might compete for the model capacity and in addition, the unused factors could be potentially spurious.

Another key component is graph regularization. After removing both $\mathcal{L}_{sps}$ and $\mathcal{L}_{bal}$, the performance drops to 0.891. This show that even if invariance regularization could penalize non-causal modules, it would be better to force their weights to be zero via sparsity regularization, and to be non-degenerate via balance regularization. We also conduct ablation studies to remove either $\mathcal{L}_{sps}$ or $\mathcal{L}_{bal}$, and results show both are important, and combining the two could help to achieve the best results.

7.5.3 Case Study: how MT-CRL help alleviate spurious correlation

To show that real-world MTL problem indeed have spurious correlation problem and our MT-CRL could alleciate it, we take MovieLens as an example to conduct case study. Each task is for different movie types, and bag-of-word of movie title is one of the features.

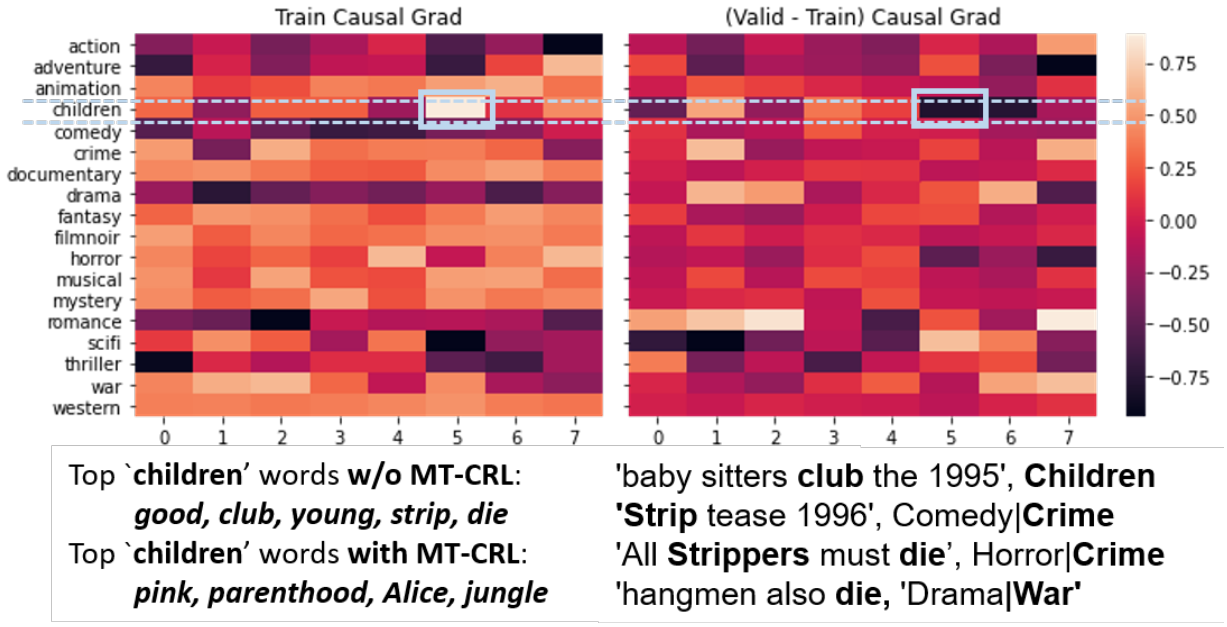


Figure 7.4: Task-to-Module gradients of model without MT-CRL show Module 5 is spurious. MT-CRL could help alleviate spurious correlation.

We calculate the task-to-module gradients $\frac{\partial(f(\Phi(x))[y])}{\partial F}$ of the vanilla MMoE model without MT-CRL. We then visualize 'train' gradients, which shows how much each module is utilized to fit the training set, and 'valid-train' gradients, which shows how generalizable each module is. We find that module 5 is utilized for **children** movie, but harmful in valid set, indicating it is a spurious feature. We then use Grad-CAM to show that top words of module 5 include *strip* and *die*, which is not relevant to **children** movies. One possible reason is that some children movies contain the words *club*, which is often co-occurred with *strip* and *die* in **crime** and **war** movies. After adding our MT-CRL, the module assigned to 'children' movie attends *Pink*, *Parenthood*, *Alice* and *Jungle*.

We then show the (valid-train) Task-to-Module gradients over Multi-MNIST datasets. With MT-CRL, in Figure 7.5, each module's saliency map only focus on one side of pixels. By looking at each task output's saliency map, which help model to focus only on causal part, compared with Figure 7.3(b) that have high weights on both. All these case studies show MT-CRL could indeed alleviate spurious correlation in real MTL problems.

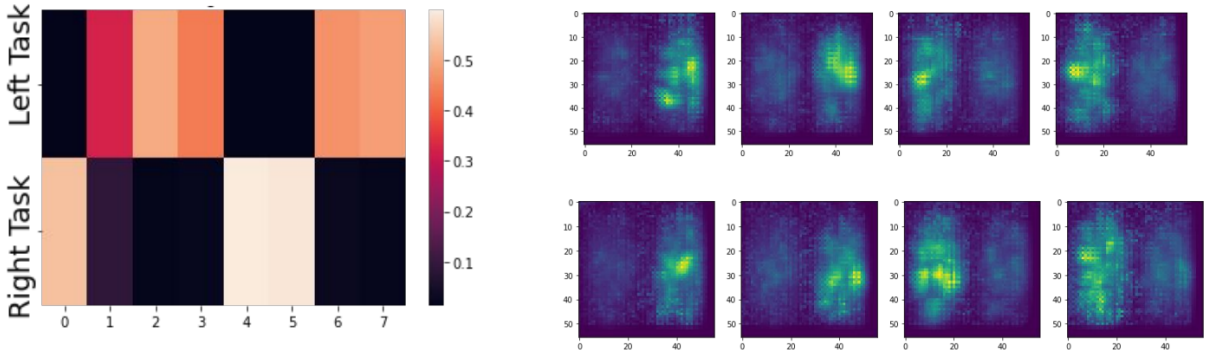


Figure 7.5: (valid-train) Task-to-Module gradients of model **with** MT-CRL on Multi-MNIST.

7.6 Summary

In this paper, we study spurious correlation problem in the Multi-Task Learning (MTL) setting. We theoretically and experimentally shows that task correlation can introduce special type of spurious correlation in MTL, and the model trained by MTL is more prone to leverage non-causal knowledge from other tasks than single-task learning. To solve the problem, we propose Multi-Task Causal Representation Learning (MT-CRL) which consists of: 1) a decorrelation regularizer to learn disentangled modules; 2) a graph regularizer to learn sparse and non-degenerate task-to-module graph; 3) G-IRM invariant regularizer. We show MT-CRL could improve performance of MTL models on benchmark datasets and could alleviate spurious correlation.

CHAPTER 8

Conclusion

This thesis set out to build a bridge between deep learning and symbolic reasoning, aiming to train a Neural-Symbolic model in an end-to-end manner without requiring intermediate labels. Through the development of a novel reasoning module, the employment of self-supervised learning techniques, and strategies for generalization across domains, significant strides have been made towards realizing this my ultimate research goal: building a Neural-Symbolic Reasoning framework that has the ability to solve challenging tasks in many different research domains of computer science.

8.1 Future Research Agenda

In the future, I am excited to further improve my proposed neural-symbolic reasoning framework, as well as using it to solve most fundamental and significant challenges in other areas in computer science, such as program synthesis, hardware design, mathematical auto-proving and scientific discovery:

Towards More Expressive Differentiable Symbolic Reasoning Systems. My past studies have focused on symbolic reasoning over knowledge graphs. Outside this research domain, there exist many other interesting and powerful symbolic reasoning AI, including

numerical reasoning, physics simulation, mathematical theorem prover, as well as many pre-defined APIs provided by industrial services. To enable integration of these symbolic reasoning capacities into neural models, I plan to build a more general interface to bridge the two worlds, supporting free interaction and backpropagation. Many challenges remain to be addressed, including how to properly model this heterogeneous and structural knowledge in a principled manner (ideally in a unified graph view), choose appropriate abstractions for reasoning procedure, and make reasoning differentiable. I am also interested in improving causal representation learning via modular design, and making the AI model capable of conducting causal inference and estimate uncertainty and risks.

Explore Program Synthesis via Neural-Symbolic Reasoning. Many fundamental tasks in computer science and artificial intelligence could be formalized as program synthesis. For example, dialogue chatbots require parsing human language into formal SQL programs; mathematical auto-proving requires transforming math equations; high-level synthesis of FPGA program requires compiling and latent execution of discrete `C/C++` programs. My past research on graph representation learning and symbolic reasoning is a natural solution for conducting program synthesis. Therefore, I am excited to apply my proposed neural-symbolic models to solve these interesting and challenging tasks. Take hardware synthesis as an example, I aim to represent symbolic program as latent variables, which we could execute via neural module to infer results. Based on it, we could search for the best program by optimizing the latent program to maximize output, in a differentiable manner.

Empower Scientific Discovery via Neural-Symbolic Reasoning. My proposed Neural-Symbolic models have already shown improvement in a wide range of artificial intelligence tasks. Outside of AI domains, many general scientific problems could also be abstracted as symbolic reasoning. For example, drug discovery and design require representing the molecule as geometric graphs; physics simulation requires understanding complex physic environments (represented with graph with particles, fluids, plasma as nodes, and their mutual interactions as edges). I am particularly interested in whether

my proposed neural-symbolic AI models could be applied and benefit these fundamental scientific problems, helping building better scientific simulation tools. In addition, I am interested in utilizing the neural-symbolic systems to automatically discover world knowledge, including constructing domain-specific knowledge graph, discovering new Physics or Chemical governing laws from experiments, and identifying causal structures from real-world social data.

Bibliography

[Agr+19] Harsh Agrawal, Peter Anderson, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, and Stefan Lee. "nocaps: novel object captioning at scale". In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 8947–8956. DOI: 10.1109/ICCV.2019.00904. URL: https://doi.org/10.1109/ICCV.2019.00904.

[Ahm+21] Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron C. Courville. "Systematic generalisation with group invariant predictions". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=b9PoimzZFJ.

[Ahu+20] Kartik Ahuja, Karthikeyan Shanmugam, Kush R. Varshney, and Amit Dhurandhar. "Invariant Risk Minimization Games". In: *CoRR* abs/2002.04692 (2020). arXiv: 2002.04692. URL: https://arxiv.org/abs/2002.04692.

[Ala+22] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. "Flamingo: a visual language model for few-shot learning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 23716–23736.

[Ant+15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. "VQA: Visual Question Answering". In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2015, pp. 2425–2433. DOI: 10.1109/ICCV.2015.279. URL: https://doi.org/10.1109/ICCV.2015.279.

[Arj+19] Martın Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. "Invariant Risk Minimization". In: *CoRR* abs/1907.02893 (2019). arXiv: 1907.02893. URL: http://arxiv.org/abs/1907.02893.

[Aro+19] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. "Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 322–332. URL: http://proceedings.mlr.press/v97/arora19a.html.

[Asa+20] Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. "Learning to Retrieve Reasoning Paths over Wikipedia Graph for Question Answering". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=SJgVHkrYDH.

[Bax00] Jonathan Baxter. "A model of inductive bias learning". In: *Journal of artificial intelligence research* 12 (2000), pp. 149–198.

[BB08] Shai Ben-David and Reba Schuller Borbely. "A notion of task relatedness yielding provable multiple-task learning guarantees". In: *Mach. Learn.* 73.3 (2008), pp. 273–287. DOI: 10.1007/s10994-007-5043-5. URL: https://doi.org/10.1007/s10994-007-5043-5.

[Ben+20] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Nan Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher J. Pal. "A Meta-Transfer Objective for Learning to Disentangle Causal Mechanisms". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=ryxWIgBFPS.

[Ben12] Yoshua Bengio. "Deep Learning of Representations for Unsupervised and Transfer Learning". In: *Unsupervised and Transfer Learning - Workshop held at ICML 2011, Bellevue, Washington, USA, July 2, 2011*. 2012, pp. 17–36. URL: http://jmlr.csail.mit.edu/proceedings/papers/v27/bengio12a.html.

[Ber+13] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. "Semantic Parsing on Freebase from Question-Answer Pairs". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, 2013, pp. 1533–1544. URL: https://aclanthology.org/D13-1160.

[BHP18] Sara Beery, Grant Van Horn, and Pietro Perona. "Recognition in Terra Incognita". In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Vol. 11220. Lecture Notes in Computer Science. Springer, 2018, pp. 472–489. DOI: 10.1007/978-3-030-01270-0_28. URL: https://doi.org/10.1007/978-3-030-01270-0%5C_28.

[BKH16] Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. "Layer Normalization". In: *CoRR* abs/1607.06450 (2016). arXiv: 1607.06450. URL: http://arxiv.org/abs/1607.06450.

[Boj+17] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching Word Vectors with Subword Information". In: *TACL* 5 (2017), pp. 135–

146. URL: https://transacl.org/ojs/index.php/tacl/article/view/999.

[Bol+08] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. "Freebase: a collaboratively created graph database for structuring human knowledge". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*. ACM, 2008, pp. 1247–1250. DOI: 10.1145/1376616.1376746. URL: https://doi.org/10.1145/1376616.1376746.

[Bor+13] Antoine Bordes, Nicolas Usunier, Alberto Garcıa-Durán, Jason Weston, and Oksana Yakhnenko. "Translating Embeddings for Modeling Multi-relational Data". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. 2013, pp. 2787–2795. URL: https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html.

[Bro+20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[Bru+13] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. "Spectral networks and locally connected networks on graphs". In: *arXiv:1312.6203* (2013).

[Cao+21] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. "Autoregressive Entity Retrieval". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=5k8F6UU39V.

[Car97] Rich Caruana. "Multitask learning". In: *Machine learning* 28.1 (1997), pp. 41–75.

[CG19] Yuan Cao and Quanquan Gu. "Generalization Bounds of Stochastic Gradient Descent for Wide and Deep Neural Networks". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett. 2019, pp. 10835–10845. URL: https://proceedings.neurips.cc/paper/2019/hash/cf9dc5e4e194fc21f397b4cac9cc3ae9-Abstract.html.

[Cha+21] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. "Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts". In: *CVPR*. 2021.

[Che+] Zhenpeng Chen, Sheng Shen, Ziniu Hu, Xuan Lu, Qiaozhu Mei, and Xuanzhe Liu. "Emoji-Powered Representation Learning for Cross-Lingual Sentiment Classification". In: *Proceedings of the ACM Web Conference (WWW 2019 Best Full Paper Award)*.

[Che+15a] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft COCO Captions: Data Collection and Evaluation Server". In: *ArXiv preprint* abs/1504.00325 (2015). URL: https://arxiv.org/abs/1504.00325.

[Che+15b] Brian Cheung, Jesse A. Livezey, Arjun K. Bansal, and Bruno A. Olshausen. "Discovering Hidden Factors of Variation in Deep Networks". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6583.

[Che+16] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett. 2016, pp. 2172–2180. URL: https://proceedings.neurips.cc/paper/2016/hash/7c9d0b1f96aebd7b5eca8c3edaa19ebb-Abstract.html.

[Che+17] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. "Reading Wikipedia to Answer Open-Domain Questions". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 1870–1879. DOI: 10.18653/v1/P17-1171. URL: https://aclanthology.org/P17-1171.

[Che+18] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. "GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 793–802. URL: http://proceedings.mlr.press/v80/chen18a.html.

[Che+20a] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. "A Simple Framework for Contrastive Learning of Visual Representations". In: *arxiv:2002.05709* (2020).

[Che+20b] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. "Just Pick a Sign: Optimizing Deep Multitask Models with Gradient Sign Dropout". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information*

Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/16002f7a455a94aa4e91cc34ebdb9f2d-Abstract.html.

[Che+22a] Wenhu Chen, Pat Verga, Michiel de Jong, John Wieting, and William Cohen. "Augmenting Pre-trained Language Models with QA-Memory for Open-Domain Question Answering". In: *CoRR* abs/2204.04581 (2022). DOI: 10.48550/arXiv.2204.04581. arXiv: 2204.04581. URL: https://doi.org/10.48550/arXiv.2204.04581.

[Che+22b] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. "Pali: A jointly-scaled multilingual language-image model". In: *arXiv preprint arXiv:2209.06794* (2022).

[Che+22c] Felix Chern, Blake Hechtman, Andy Davis, Ruiqi Guo, David Majnemer, and Sanjiv Kumar. "TPU-KNN: K Nearest Neighbor Search at Peak FLOP/s". In: *CoRR* abs/2206.14286 (2022). DOI: 10.48550/arXiv.2206.14286. arXiv: 2206.14286. URL: https://doi.org/10.48550/arXiv.2206.14286.

[Cho+22] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. "Palm: Scaling language modeling with pathways". In: *arXiv preprint arXiv:2204.02311* (2022).

[CK04] Nigel Collier and Jin-Dong Kim. "Introduction to the Bio-entity Recognition Task at JNLPBA". In: *NLPBA/BioNLP*. 2004.

[CMX18] Jie Chen, Tengfei Ma, and Cao Xiao. "FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 2018.

[Cog+16] Michael Cogswell, Faruk Ahmed, Ross B. Girshick, Larry Zitnick, and Dhruv Batra. "Reducing Overfitting in Deep Networks by Decorrelating Representations". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2016. URL: http://arxiv.org/abs/1511.06068.

[Coh+17] Trevor Cohn, Steven Bird, Graham Neubig, Oliver Adams, and Adam J. Makarucha. "Cross-Lingual Word Embeddings for Low-Resource Language Modeling". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*. Ed. by Mirella Lapata, Phil Blunsom, and Alexander Koller. Association for Computational Linguistics, 2017, pp. 937–947. URL: https://aclanthology.info/papers/E17-1088/e17-1088.

[Cor+16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 3213–3223. DOI: 10.1109/CVPR.2016.350. URL: https://doi.org/10.1109/CVPR.2016.350.

[CZS] Xuelu Chen, **Ziniu Hu**, and Yizhou Sun. "Fuzzy Logic based Logical Query Answering on Knowledge Graphs". In: *AAAI Conference on Artificial Intelligence (AAAI 2022)*.

[CZS18] Jianfei Chen, Jun Zhu, and Le Song. "Stochastic Training of Graph Convolutional Networks with Variance Reduction". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 2018.

[Das+18] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. "Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=Syg-YfWCW.

[Das+21] Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. "Case-based Reasoning for Natural Language Queries over Knowledge Bases". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 9594–9611. DOI: 10.18653/v1/2021.emnlp-main.755. URL: https://aclanthology.org/2021.emnlp-main.755.

[Das+22] Rajarshi Das, Ameya Godbole, Ankita Naik, Elliot Tower, Robin Jia, Manzil Zaheer, Hannaneh Hajishirzi, and Andrew McCallum. "Knowledge Base Question Answering by Case-based Reasoning over Subgraphs". In: *ArXiv preprint* abs/2202.10610 (2022). URL: https://arxiv.org/abs/2202.10610.

[DCS17] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. "metapath2vec: Scalable Representation Learning for Heterogeneous Networks". In: *KDD 2017*. 2017.

[Den+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: (2009).

[Der+17] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. "Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition". In: *NUT@EMNLP*. Association for Computational Linguistics, 2017, pp. 140–147.

[Dev+] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *preprint arXiv:1810.04805* (). URL: https://arxiv.org/pdf/1810.04805.pdf.

[Dev+19a] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[Dev+19b] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *NAACL 2019*. 2019.

[Dhi+20] Bhuwan Dhingra, Manzil Zaheer, Vidhisha Balachandran, Graham Neubig, Ruslan Salakhutdinov, and William W. Cohen. "Differentiable Reasoning over a Virtual Knowledge Base". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=SJxstlHFPH.

[Din+19] Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. "Cognitive Graph for Multi-Hop Reading Comprehension at Scale". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 2694–2703. DOI: 10.18653/v1/P19-1259. URL: https://aclanthology.org/P19-1259.

[Don+] Yuxiao Dong, Ziniu Hu, Kuansan Wang, Yizhou Sun, and Jie Tang. "Heterogeneous Network Representation Learning". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2020)*.

[Don+14a] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition". In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. 2014, pp. 647–655. URL: http://jmlr.org/proceedings/papers/v32/donahue14.html.

[Don+14b] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. "Decaf: A deep convolutional activation feature for generic visual recognition". In: *ICML 2014*. 2014.

[Don+20] Yuxiao Dong, Ziniu Hu, Kuansan Wang, Yizhou Sun, and Jie Tang. "Heterogeneous Network Representation Learning". In: *IJCAI*. 2020.

[Dos+21] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[Du+21] Simon Shaolei Du, Wei Hu, Sham M. Kakade, Jason D. Lee, and Qi Lei. "Few-Shot Learning via Learning the Representation, Provably". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=pW2Q2xLwIMD.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2017, pp. 1126–1135. URL: http://proceedings.mlr.press/v70/finn17a.html.

[Fen+20] Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. "Scalable Multi-Hop Relational Reasoning for Knowledge-Aware

Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 1295–1309. DOI: 10.18653/v1/2020.emnlp-main.99. URL: https://aclanthology.org/2020.emnlp-main.99.

[Fév+20] Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. "Entities as Experts: Sparse Memory Access with Entity Supervision". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 4937–4951. DOI: 10.18653/v1/2020.emnlp-main.400. URL: https://aclanthology.org/2020.emnlp-main.400.

[Fif+21] Christopher Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. "Efficiently Identifying Task Groupings for Multi-Task Learning". In: *CoRR* abs/2109.04617 (2021). arXiv: 2109.04617. URL: https://arxiv.org/abs/2109.04617.

[Fir57] John R. Firth. "A Synopsis of Linguistic Theory, 1930-1955". In: *Studies in Linguistic Analysis* (1957).

[FL19] Matthias Fey and Jan Eric Lenssen. "Fast Graph Representation Learning with PyTorch Geometric". In: *ICLR Workshop* (2019).

[Gan+16] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. "Domain-Adversarial Training of Neural Networks". In: *J. Mach. Learn. Res.* 17 (2016), 59:1–59:35. URL: http://jmlr.org/papers/v17/15-239.html.

[Gao+19] Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. "Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-19), New York, USA, April 15-18, 2019*. 2019.

[Gar+20] François Gardères, Maryam Ziaeefard, Baptiste Abeloos, and Freddy Lecue. "ConceptBert: Concept-Aware Representation for Visual Question Answering". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, 2020, pp. 489–498. DOI: 10.18653/v1/2020.findings-emnlp.44. URL: https://aclanthology.org/2020.findings-emnlp.44.

[Gei+19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=Bygh9j09KX.

[Gei+20] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. "Shortcut learning in deep neural networks". In: *Nat. Mach. Intell.* 2.11 (2020), pp. 665–673. DOI: 10.1038/s42256-020-00257-z. URL: https://doi.org/10.1038/s42256-020-00257-z.

[Gil+17a] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. "Neural Message Passing for Quantum Chemistry". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2017, pp. 1263–1272. URL: http://proceedings.mlr.press/v70/gilmer17a.html.

[Gil+17b] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. "Neural Message Passing for Quantum Chemistry". In: *ICML 2017*. 2017.

[Gir+14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *CVPR 2014*. 2014.

[GL16] Aditya Grover and Jure Leskovec. “node2vec: Scalable feature learning for networks”. In: *KDD 2016*. 2016.

[GLU20] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. “Learning to Branch for Multi-Task Learning”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 3854–3863. URL: http://proceedings.mlr.press/v119/guo20e.html.

[Goy+17] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. “Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 6325–6334. DOI: 10.1109/CVPR.2017.670. URL: https://doi.org/10.1109/CVPR.2017.670.

[Gu+18] Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. “Meta-Learning for Low-Resource Neural Machine Translation”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Ed. by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun’ichi Tsujii. Association for Computational Linguistics, 2018, pp. 3622–3631. URL: https://aclanthology.info/papers/D18-1398/d18-1398.

[Gui+21] Liangke Gui, Borui Wang, Qiuyuan Huang, Alex Hauptmann, Yonatan Bisk, and Jianfeng Gao. “Kat: A knowledge augmented transformer for vision-and-language”. In: *arXiv preprint arXiv:2112.08614* (2021).

[Gur+18] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. “Annotation Artifacts in Natural Language Inference Data”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018,*

Volume 2 (Short Papers). Ed. by Marilyn A. Walker, Heng Ji, and Amanda Stent. Association for Computational Linguistics, 2018, pp. 107–112. DOI: 10.18653/v1/n18-2017. URL: https://doi.org/10.18653/v1/n18-2017.

[Guu+20] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. "REALM: Retrieval-Augmented Language Model Pre-Training". In: *ArXiv preprint* abs/2002.08909 (2020). URL: https://arxiv.org/abs/2002.08909.

[Hai12] Jens Hainmueller. "Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies". In: *Political analysis* 20.1 (2012), pp. 25–46.

[HB17] Aurélie Herbelot and Marco Baroni. "High-risk learning: acquiring new word vectors from tiny data". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Association for Computational Linguistics, 2017, pp. 304–309. URL: https://aclanthology.info/papers/D17-1030/d17-1030.

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90. URL: https://doi.org/10.1109/CVPR.2016.90.

[He+19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. "Momentum contrast for unsupervised visual representation learning". In: *arXiv:1911.05722* (2019).

[Hig+17] Irina Higgins, Loıc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *5th International Conference on Learning Representations, ICLR 2017,*

Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017. URL: https://openreview.net/forum?id=Sy2fzU9gl.

[HK16] F. Maxwell Harper and Joseph A. Konstan. "The MovieLens Datasets: History and Context". In: *ACM Trans. Interact. Intell. Syst.* 5.4 (2016), 19:1–19:19. DOI: 10.1145/2827872. URL: https://doi.org/10.1145/2827872.

[HLJ16] William L. Hamilton, Jure Leskovec, and Dan Jurafsky. "Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. 2016. URL: http://aclweb.org/anthology/P/P16/P16-1141.pdf.

[HSC21] Ziniu Hu, Yizhou Sun, and Kai-Wei Chang. "Relation-Guided Pre-Training for Open-Domain Question Answering". In: *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Association for Computational Linguistics, 2021, pp. 3431–3448. DOI: 10.18653/v1/2021.findings-emnlp.292. URL: https://doi.org/10.18653/v1/2021.findings-emnlp.292.

[Hu+20] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. "Strategies for Pre-training Graph Neural Networks". In: *ICLR 2020*. 2020.

[Hu+22a] Ziniu Hu, Yichong Xu, Wenhao Yu, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Kai-Wei Chang, and Yizhou Sun. "Empowering Language Models with Knowledge Graph Reasoning for Open-Domain Question Answering". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Association for Computational Linguistics, 2022, pp. 9562–9581. URL: https://aclanthology.org/2022.emnlp-main.650.

[Hu+22b] Ziniu Hu, Zhe Zhao, Xinyang Yi, Tiansheng Yao, Lichan Hong, Yizhou Sun, and Ed H. Chi. "Improving Multi-Task Generalization via Regularizing Spurious Correlation". In: *CoRR* abs/2205.09797 (2022). DOI: 10.48550/arXiv.2205.09797. arXiv: 2205.09797. URL: https://doi.org/10.48550/arXiv.2205.09797.

[HYL17] William L. Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive Representation Learning on Large Graphs". In: *NeurIPS 2017*. 2017.

[IG21] Gautier Izacard and Edouard Grave. "Distilling Knowledge from Reader to Retriever for Question Answering". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=NTEz-6wysdb.

[Iza+22] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. "Few-shot Learning with Retrieval Augmented Language Models". In: *CoRR* abs/2208.03299 (2022). DOI: 10.48550/arXiv.2208.03299. arXiv: 2208.03299. URL: https://doi.org/10.48550/arXiv.2208.03299.

[Jae+21] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and João Carreira. "Perceiver: General Perception with Iterative Attention". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 4651–4664. URL: http://proceedings.mlr.press/v139/jaegle21a.html.

[Jia+21] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. "Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Vol. 139. Proceedings of Machine

Learning Research. PMLR, 2021, pp. 4904–4916. URL: http://proceedings.mlr.press/v139/jia21b.html.

[Jos+17] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 1601–1611. DOI: 10.18653/v1/P17-1147. URL: https://aclanthology.org/P17-1147.

[Kam+22] Amita Kamath, Christopher Clark, Tanmay Gupta, Eric Kolve, Derek Hoiem, and Aniruddha Kembhavi. "Webly Supervised Concept Expansion for General Purpose Vision Models". In: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXVI*. Vol. 13696. Lecture Notes in Computer Science. Springer, 2022, pp. 662–681. DOI: 10.1007/978-3-031-20059-5_38. URL: https://doi.org/10.1007/978-3-031-20059-5%5C_38.

[Kar+20] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. "Dense Passage Retrieval for Open-Domain Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. URL: https://aclanthology.org/2020.emnlp-main.550.

[Ke+21] Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. "JointGT: Graph-Text Joint Representation Learning for Text Generation from Knowledge Graphs". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, 2021, pp. 2526–2538. DOI: 10.18653/v1/2021.

findings-acl.223. URL: https://aclanthology.org/2021.findings-acl.223.

[KGC18] Alex Kendall, Yarin Gal, and Roberto Cipolla. "Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 7482–7491. DOI: 10.1109/CVPR.2018.00781. URL: http://openaccess.thecvf.com/content%5C_cvpr%5C_2018/html/Kendall%5C_Multi-Task%5C_Learning%5C_Using%5C_CVPR%5C_2018%5C_paper.html.

[Kho+18] Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. "A La Carte Embedding: Cheap but Effective Induction of Semantic Feature Vectors". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. Ed. by Iryna Gurevych and Yusuke Miyao. Association for Computational Linguistics, 2018, pp. 12–22. URL: https://aclanthology.info/papers/P18-1002/p18-1002.

[Kim+16] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. "Character-Aware Neural Language Models". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*. 2016, pp. 2741–2749. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12489.

[KL21] Fereshte Khani and Percy Liang. "Removing Spurious Features can Hurt Accuracy and Affect Groups Disproportionately". In: *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*. Ed. by Madeleine Clare Elish, William Isaac, and Richard S. Zemel. ACM, 2021, pp. 196–205. DOI: 10.1145/3442188.3445883. URL: https://doi.org/10.1145/3442188.3445883.

[Kru+21] David Krueger, Ethan Caballero, Jörn-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Rémi Le Priol, and Aaron C. Courville. "Out-of-Distribution Generalization via Risk Extrapolation (REx)". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 5815–5826. URL: http://proceedings.mlr.press/v139/krueger21a.html.

[KW16] Thomas N. Kipf and Max Welling. "Variational Graph Auto-Encoders". In: *arXiv:1611.07308* (2016).

[KW17] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *ICLR 2017*. 2017.

[Kwi+19] Tom Kwiatkowski et al. "Natural Questions: A Benchmark for Question Answering Research". In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 452–466. DOI: 10.1162/tacl_a_00276. URL: https://aclanthology.org/Q19-1026.

[KY21] Masanori Koyama and Shoichiro Yamaguchi. "When is invariance useful in an Out-of-Distribution Generalization problem?" In: *CoRR* abs/2008.01883 (2021). arXiv: 2008.01883. URL: https://arxiv.org/abs/2008.01883.

[Lac+20] Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. "Gradient-Based Neural DAG Learning". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=rklbKA4YDS.

[Lac+21] Sébastien Lachapelle, Pau Rodrıguez López, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. "Disentanglement via Mechanism Sparsity Regularization: A New Principle for Nonlinear ICA". In: *arXiv preprint arXiv:2107.10098* (2021).

[Lam+16] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. "Neural Architectures for Named Entity Recognition". In: *HLT-NAACL*. The Association for Computational Linguistics, 2016, pp. 260–270.

[Lew+20] Patrick S. H. Lewis et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[Lew+21] Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. "PAQ: 65 Million Probably-Asked Questions and What You Can Do With Them". In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 1098–1115. DOI: 10.1162/tacl_a_00415. URL: https://aclanthology.org/2021.tacl-1.65.

[LG19] Andrew K. Lampinen and Surya Ganguli. "An analytic theory of generalization dynamics and transfer learning in deep linear networks". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=ryfMLoCqtQ.

[LH17] Ilya Loshchilov and Frank Hutter. "SGDR: Stochastic Gradient Descent with Warm Restarts". In: *ICLR 2017*. 2017.

[LH19] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *ICLR 2019*. 2019.

[Li14] Hang Li. *Learning to Rank for Information Retrieval and Natural Language Processing, Second Edition*. Synthesis Lectures on Human Language Technolo-

gies. Morgan & Claypool Publishers, 2014. DOI: 10.2200/S00607ED2V01Y201410HLT026. URL: https://doi.org/10.2200/S00607ED2V01Y201410HLT026.

[Lia+19] Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, William L. Hamilton, David Duvenaud, Raquel Urtasun, and Richard S. Zemel. "Efficient Graph Generation with Graph Recurrent Attention Networks". In: *NeurIPS 2019*. 2019.

[Lin+19] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. "KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning". 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)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 2829–2839. DOI: 10.18653/v1/D19-1282. URL: https://aclanthology.org/D19-1282.

[Lin+21] Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William Cohen. "Differentiable Open-Ended Commonsense Reasoning". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, 2021, pp. 4611–4625. DOI: 10.18653/v1/2021.naacl-main.366. URL: https://aclanthology.org/2021.naacl-main.366.

[Lin+22] Yuanze Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. "Revive: Regional visual representation matters in knowledge-based visual question answering". In: *arXiv preprint arXiv:2206.01201* (2022).

[Liu+20] Chang Liu, Xinwei Sun, Jindong Wang, Tao Li, Tao Qin, Wei Chen, and Tie-Yan Liu. "Learning Causal Semantic Representation for Out-of-Distribution Prediction". In: *CoRR* abs/2011.01681 (2020). arXiv: 2011.01681. URL: https://arxiv.org/abs/2011.01681.

[Liu+21] Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. "KG-BART: Knowledge Graph-Augmented BART for Generative Commonsense Reasoning". In: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 6418–6425. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16796.

[Liu11] Tie-Yan Liu. *Learning to Rank for Information Retrieval*. Springer, 2011. ISBN: 978-3-642-14266-6.

[LJD19] Shikun Liu, Edward Johns, and Andrew J. Davison. "End-To-End Multi-Task Learning With Attention". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 1871–1880. DOI: 10.1109/CVPR.2019.00197. URL: http://openaccess.thecvf.com/content%5CCVPR%5C2019/html/Liu%5CEnd-To-End%5CMulti-Task%5CLearning%5CWith%5CAttention%5CCVPR%5C2019%5Cpaper.html.

[LMB17] Angeliki Lazaridou, Marco Marelli, and Marco Baroni. "Multimodal Word Meaning Induction From Minimal Exposure to Natural Text". In: *Cognitive Science*. (2017).

[LMC11] Ni Lao, Tom Mitchell, and William W. Cohen. "Random Walk Inference and Learning in A Large Scale Knowledge Base". In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, 2011, pp. 529–539. URL: https://aclanthology.org/D11-1049.

[Loc+19] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations". In: *Proceed-*

ings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 4114–4124. URL: http://proceedings.mlr.press/v97/locatello19a.html.

[Lon+22] Alexander Long, Wei Yin, Thalaiyasingam Ajanthan, Vu Nguyen, Pulak Purkait, Ravi Garg, Alan Blair, Chunhua Shen, and Anton van den Hengel. "Retrieval Augmented Classification for Long-Tail Visual Recognition". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 6949–6959. DOI: 10.1109/CVPR52688.2022.00683. URL: https://doi.org/10.1109/CVPR52688.2022.00683.

[Lop16] David Lopez-Paz. "From Dependence to Causation". In: *arXiv: Machine Learning* (2016).

[LSM13] Thang Luong, Richard Socher, and Christopher D. Manning. "Better Word Representations with Recursive Neural Networks for Morphology". In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*. Ed. by Julia Hockenmaier and Sebastian Riedel. ACL, 2013, pp. 104–113. URL: http://aclweb.org/anthology/W/W13/W13-3512.pdf.

[LSR21] Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. "Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, 2021, pp. 1000–1008. DOI: 10.18653/v1/2021.eacl-main.86. URL: https://aclanthology.org/2021.eacl-main.86.

[Lu+17] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogério Schmidt Feris. "Fully-Adaptive Feature Sharing in Multi-Task Net-

works with Applications in Person Attribute Classification". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 1131–1140. DOI: 10.1109/CVPR.2017.126. URL: https://doi.org/10.1109/CVPR.2017.126.

[Lu+19] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 2019, pp. 13–23. URL: https://proceedings.neurips.cc/paper/2019/hash/c74d97b01eae257e44aa9d5bade97baf-Abstract.html.

[Luo+21] Man Luo, Yankai Zeng, Pratyay Banerjee, and Chitta Baral. "Weakly-supervised visual-retriever-reader for knowledge-based question answering". In: *arXiv preprint arXiv:2109.04014* (2021).

[Ma+18] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. "Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. Ed. by Yike Guo and Faisal Farooq. ACM, 2018, pp. 1930–1939. DOI: 10.1145/3219819.3220007. URL: https://doi.org/10.1145/3219819.3220007.

[Mar+19] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. "OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 3195–3204. DOI: 10.1109/CVPR.2019.00331. URL: http://openaccess.thecvf.com/content%5C_CVPR%5C_2019/html/

Marino%5C_OK-VQA%5C_A%5C_Visual%5C_Question%5C_Answering%5C_Benchmark%5C_Requiring%5C_External%5C_Knowledge%5C_CVPR%5C_2019%5C_paper.html.

[Mar+21] Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. "Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14111–14121.

[Mer+17] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. "Pointer Sentinel Mixture Models". In: *ICLR'17*. 2017.

[MHB21] Ron Mokady, Amir Hertz, and Amit H. Bermano. "ClipCap: CLIP Prefix for Image Captioning". In: *ArXiv preprint* abs/2111.09734 (2021). URL: https://arxiv.org/abs/2111.09734.

[Mik+13a] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[Mik+13b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. "Distributed Representations of Words and Phrases and their Compositionality". In: *NIPS*. 2013, pp. 3111–3119.

[Min+19] Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. "Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering". In: *ArXiv preprint* abs/1911.03868 (2019). URL: https://arxiv.org/abs/1911.03868.

[Mis+16] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. "Cross-Stitch Networks for Multi-task Learning". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 3994–4003. DOI:

10.1109/CVPR.2016.433. URL: https://doi.org/10.1109/CVPR.2016.433.

[Mit+21] Jovana Mitrovic, Brian McWilliams, Jacob C. Walker, Lars Holger Buesing, and Charles Blundell. "Representation Learning via Invariant Causal Mechanisms". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=9p2ekP904Rs.

[MPR16] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. "The Benefit of Multitask Representation Learning". In: *J. Mach. Learn. Res.* 17 (2016), 81:1–81:32. URL: http://jmlr.org/papers/v17/15-242.html.

[MRK19] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. "Attentive Single-Tasking of Multiple Tasks". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 1851–1860. DOI: 10.1109/CVPR.2019.00195. URL: http://openaccess.thecvf.com/content%5C_CVPR%5C_2019/html/Maninis%5C_Attentive%5C_Single-Tasking%5C_of%5C_Multiple%5C_Tasks%5C_CVPR%5C_2019%5C_paper.html.

[NAN21] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. "Understanding the failure modes of out-of-distribution generalization". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=fSTD6NFIW%5C_b.

[Ng+19] Ignavier Ng, Zhuangyan Fang, Shengyu Zhu, and Zhitang Chen. "Masked Gradient-Based Causal Structure Learning". In: *CoRR* abs/1910.08527 (2019). arXiv: 1910.08527. URL: http://arxiv.org/abs/1910.08527.

[NLM19] Jianmo Ni, Jiacheng Li, and Julian J. McAuley. "Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects". In: *EMNLP 2019*. 2019.

[NLS18] Medhini Narasimhan, Svetlana Lazebnik, and Alexander G. Schwing. "Out of the Box: Reasoning with Graph Convolution Nets for Factual Visual Question Answering". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. 2018, pp. 2659–2670. URL: https://proceedings.neurips.cc/paper/2018/hash/c26820b8a4c1b3c2aa868d6d57e14Abstract.html.

[Ogu+20] Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Sejr Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. "Unified Open-Domain Question Answering with Structured and Unstructured Knowledge". In: *ArXiv preprint* abs/2012.14610 (2020). URL: https://arxiv.org/abs/2012.14610.

[OLV18] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation Learning with Contrastive Predictive Coding". In: *arXiv:1807.03748* (2018).

[Par+18] Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. "Learning Independent Causal Mechanisms". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 4033–4041. URL: http://proceedings.mlr.press/v80/parascandolo18a.html.

[Pat+16] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. "Context Encoders: Feature Learning by Inpainting". In: *CVPR 2016*. 2016.

[PBM16] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. "Causal inference by using invariant prediction: identification and confidence intervals". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* (2016), pp. 947–1012.

[PBS16] Emilio Parisotto, Lei Jimmy Ba, and Ruslan Salakhutdinov. "Actor-Mimic: Deep Multitask and Transfer Reinforcement Learning". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2016. URL: http://arxiv.org/abs/1511.06342.

[Pet+18] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. "Deep Contextualized Word Representations". In: *NAACL-HLT*. Association for Computational Linguistics, 2018, pp. 2227–2237.

[PGE17] Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. "Mimicking Word Embeddings using Subword RNNs". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Association for Computational Linguistics, 2017, pp. 102–112. URL: https://aclanthology.info/papers/D17-1010/d17-1010.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation". In: *EMNLP 2014*. 2014.

[PWS20] Nina Poerner, Ulli Waltinger, and Hinrich Schütze. "E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, 2020, pp. 803–818. DOI: 10.18653/v1/2020.findings-emnlp.71. URL: https://aclanthology.org/2020.findings-emnlp.71.

[Qiu+18] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec". In: *WSDM '18*. 2018, pp. 459–467.

[Rad+19] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language Models are Unsupervised Multitask Learners". In: (2019).

[Raf+20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67. URL: http://jmlr.org/papers/v21/20-074.html.

[RHL20] Hongyu Ren, Weihua Hu, and Jure Leskovec. "Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=BJgr4kSFDS.

[Rit+11] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. "Named Entity Recognition in Tweets: An Experimental Study". In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, 2011, pp. 1524–1534. URL: http://aclweb.org/anthology/D11-1141.

[RKR18] Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. "Routing Networks: Adaptive Selection of Non-Linear Functions for Multi-Task Learning". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=ry8dvM-R-.

[RL17] Sachin Ravi and Hugo Larochelle. "Optimization as a Model for Few-Shot Learning". In: *ICLR*. 2017.

[RRR21] Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. "The Risks of Invariant Risk Minimization". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=BbNIbVPJ-42.

[RRS20] Adam Roberts, Colin Raffel, and Noam Shazeer. "How Much Knowledge Can You Pack Into the Parameters of a Language Model?" In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 5418–5426. DOI: 10.18653/v1/2020.emnlp-main.437. URL: https://aclanthology.org/2020.emnlp-main.437.

[Sac+21a] Devendra Singh Sachan, Siva Reddy, William L. Hamilton, Chris Dyer, and Dani Yogatama. "End-to-End Training of Multi-Document Reader and Retriever for Open-Domain Question Answering". In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan. 2021, pp. 25968–25981. URL: https://proceedings.neurips.cc/paper/2021/hash/da3fde159d754a2555eaa198d2d105b2-Abstract.html.

[Sac+21b] Devendra Singh Sachan, Siva Reddy, William L. Hamilton, Chris Dyer, and Dani Yogatama. "End-to-End Training of Multi-Document Reader and Retriever for Open-Domain Question Answering". In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. 2021, pp. 25968–25981. URL: https://proceedings.neurips.cc/paper/2021/hash/da3fde159d754a2555eaa198d2d105b2-Abstract.html.

[Sag+20] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. "Distributionally Robust Neural Networks". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=ryxGuJrFvS.

[SBC19] Haitian Sun, Tania Bedrax-Weiss, and William Cohen. "PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and 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)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 2380–2390. DOI: 10.18653/v1/D19-1242. URL: https://aclanthology.org/D19-1242.

[Sch+18] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. "Modeling Relational Data with Graph Convolutional Networks". In: *ESWC 2018*. 2018.

[Sch+21] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. "Toward Causal Representation Learning". In: *Proc. IEEE* 109.5 (2021), pp. 612–634. DOI: 10.1109/JPROC.2021.3058954. URL: https://doi.org/10.1109/JPROC.2021.3058954.

[Sch+22] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. "A-OKVQA: A Benchmark for Visual Question Answering using World Knowledge". In: *CoRR* abs/2206.01718 (2022). DOI: 10.48550/arXiv.2206.01718. arXiv: 2206.01718. URL: https://doi.org/10.48550/arXiv.2206.01718.

[Sci+21] Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. "Simple Entity-Centric Questions Challenge Dense Retrievers". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 6138–6148. DOI: 10.18653/v1/2021.emnlp-main.496. URL: https://aclanthology.org/2021.emnlp-main.496.

[SH12] Yizhou Sun and Jiawei Han. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan & Claypool Publishers, 2012.

[Sha+17] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=B1ckMDqlg.

[Sil+12] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. "Indoor Segmentation and Support Inference from RGBD Images". In: *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V*. Ed. by Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid. Vol. 7576. Lecture Notes in Computer Science. Springer, 2012, pp. 746–760. DOI: 10.1007/978-3-642-33715-4_54. URL: https://doi.org/10.1007/978-3-642-33715-4%5C_54.

[Sin+15] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Paul Hsu, and Kuansan Wang. "An Overview of Microsoft Academic Service (MAS) and Applications". In: *WWW 2015*. 2015.

[SK18] Ozan Sener and Vladlen Koltun. "Multi-Task Learning as Multi-Objective Optimization". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman

Garnett. 2018, pp. 525–536. URL: https://proceedings.neurips.cc/paper/2018/hash/432aca3a1e345e339f35a30c8f65edce-Abstract.html.

[SL14] Anshumali Shrivastava and Ping Li. "Asymmetric LSH (ALSH) for Sublinear Time Maximum Inner Product Search (MIPS)". In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. 2014, pp. 2321–2329. URL: https://proceedings.neurips.cc/paper/2014/hash/310ce61c90f3a46e340ee8257bc70e93-Abstract.html.

[SLS18] Prathusha K. Sarma, Yingyu Liang, and Bill Sethares. "Domain Adapted Word Embeddings for Improved Sentiment Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*. 2018, pp. 37–42. URL: https://aclanthology.info/papers/P18-2007/p18-2007.

[Sri+21] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. "WIT: Wikipedia-based Image Text Dataset for Multimodal Multilingual Machine Learning". In: *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 2021, pp. 2443–2449. DOI: 10.1145/3404835.3463257. URL: https://doi.org/10.1145/3404835.3463257.

[SSZ17] Jake Snell, Kevin Swersky, and Richard S. Zemel. "Prototypical Networks for Few-shot Learning". In: *NIPS*. 2017, pp. 4080–4090.

[Sta+20] Trevor Standley, Amir Roshan Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. "Which Tasks Should Be Learned Together in Multi-task Learning?" In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119.

Proceedings of Machine Learning Research. PMLR, 2020, pp. 9120–9132. URL: http://proceedings.mlr.press/v119/standley20a.html.

[Sun+11] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks". In: *VLDB 2011*. 2011.

[Sun+12] Yizhou Sun, Brandon Norick, Jiawei Han, Xifeng Yan, Philip S. Yu, and Xiao Yu. "Integrating meta-path selection with user-guided object clustering in heterogeneous information networks". In: *KDD 2012*. 2012.

[Sun+20] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. "InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization". In: *ICLR 2020*. 2020.

[Sun19] Shao-Hua Sun. *Multi-digit MNIST for Few-shot Learning*. 2019. URL: https://github.com/shaohua0116/MultiDigitMNIST.

[Sut+19] Raphael Suter, DHe Miladinovic, Bernhard Schölkopf, and Stefan Bauer. "Robustly Disentangled Causal Mechanisms: Validating Deep Representations for Interventional Robustness". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 6056–6065. URL: http://proceedings.mlr.press/v97/suter19a.html.

[SVZ14] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: http://arxiv.org/abs/1312.6034.

[Tan+08] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. "Arnetminer: extraction and mining of academic social networks". In: *KDD 2008*. 2008.

[Tan+15] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. "Line: Large-scale information network embedding". In: *WWW 2015*. 2015.

[TB18] Alon Talmor and Jonathan Berant. "The Web as a Knowledge-Base for Answering Complex Questions". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 641–651. DOI: 10.18653/v1/N18-1059. URL: https://aclanthology.org/N18-1059.

[TB19] Hao Tan and Mohit Bansal. "LXMERT: Learning Cross-Modality Encoder Representations from Transformers". 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)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 5100–5111. DOI: 10.18653/v1/D19-1514. URL: https://aclanthology.org/D19-1514.

[TE11] Antonio Torralba and Alexei A. Efros. "Unbiased look at dataset bias". In: *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. IEEE Computer Society, 2011, pp. 1521–1528. DOI: 10.1109/CVPR.2011.5995347. URL: https://doi.org/10.1109/CVPR.2011.5995347.

[TJJ20] Nilesh Tripuraneni, Michael I. Jordan, and Chi Jin. "On the Theory of Transfer Learning: The Importance of Task Diversity". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan,

and Hsuan-Tien Lin. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/59587bffec1c7846f3e34230141556ae-Abstract.html.

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *NIPS*. 2017, pp. 6000–6010.

[Vel+18] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. "Graph Attention Networks". In: *ICLR 2018*. 2018.

[Vel+19] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. "Deep Graph Infomax". In: *ICLR 2019*. 2019.

[Ver+21] Pat Verga, Haitian Sun, Livio Baldini Soares, and William Cohen. "Adaptable and Interpretable Neural MemoryOver Symbolic Knowledge". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, 2021, pp. 3678–3691. DOI: 10.18653/v1/2021.naacl-main.288. URL: https://aclanthology.org/2021.naacl-main.288.

[Vin+16] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. "Matching Networks for One Shot Learning". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 2016, pp. 3630–3638. URL: http://papers.nips.cc/paper/6385-matching-networks-for-one-shot-learning.

[VK14] Denny Vrandecic and Markus Krötzsch. "Wikidata: a free collaborative knowledgebase". In: *Commun. ACM* 57.10 (2014), pp. 78–85. DOI: 10.1145/2629489. URL: https://doi.org/10.1145/2629489.

[VZP15] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. "CIDEr: Consensus-based image description evaluation". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pp. 4566–4575. DOI: 10.1109/CVPR.2015.7299087. URL: https://doi.org/10.1109/CVPR.2015.7299087.

[Wan+] Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. "Improving Neural Language Generation with Spectrum Control". In: *International Conference on Learning Representations (ICLR 2020)*.

[Wan+17] Peng Wang, Qi Wu, Chunhua Shen, Anthony R. Dick, and Anton van den Hengel. "Explicit Knowledge-based Reasoning for Visual Question Answering". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. ijcai.org, 2017, pp. 1290–1296. DOI: 10.24963/ijcai.2017/179. URL: https://doi.org/10.24963/ijcai.2017/179.

[Wan+18] Peng Wang, Qi Wu, Chunhua Shen, Anthony R. Dick, and Anton van den Hengel. "FVQA: Fact-Based Visual Question Answering". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.10 (2018), pp. 2413–2427. DOI: 10.1109/TPAMI.2017.2754246. URL: https://doi.org/10.1109/TPAMI.2017.2754246.

[Wan+19a] Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. "Balanced Datasets Are Not Enough: Estimating and Mitigating Gender Bias in Deep Image Representations". In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 5309–5318. DOI: 10.1109/ICCV.2019.00541. URL: https://doi.org/10.1109/ICCV.2019.00541.

[Wan+19b] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. "Heterogeneous Graph Attention Network". In: *WWW 2019*. 2019.

[Wan+20] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. "Microsoft Academic Graph: When experts are not enough". In: *Quantitative Science Studies* 1.1 (2020), pp. 396–413.

[Wan+21a] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. "KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation". In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 176–194. DOI: 10.1162/tacl_a_00360. URL: https://aclanthology.org/2021.tacl-1.11.

[Wan+21b] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. "Gradient Vaccine: Investigating and Improving Multi-task Optimization in Massively Multilingual Models". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=F1vEjWK-lH%5C_.

[Wan+22a] Wenhui Wang et al. "Image as a Foreign Language: BEiT Pretraining for All Vision and Vision-Language Tasks". In: *CoRR* abs/2208.10442 (2022). DOI: 10.48550/arXiv.2208.10442. arXiv: 2208.10442. URL: https://doi.org/10.48550/arXiv.2208.10442.

[Wan+22b] Yanan Wang, Michihiro Yasunaga, Hongyu Ren, Shinya Wada, and Jure Leskovec. "VQA-GNN: Reasoning with Multimodal Semantic Graph for Visual Question Answering". In: *CoRR* abs/2205.11501 (2022). DOI: 10.48550/arXiv.2205.11501. arXiv: 2205.11501. URL: https://doi.org/10.48550/arXiv.2205.11501.

[Wan+22c] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. "SimVLM: Simple Visual Language Model Pretraining with Weak Supervision". In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL: https://openreview.net/forum?id=GUrhfTuf%5C_3.

[Wei+] Tianxin Wei, Ziwei Wu, Ruirui Li, Ziniu Hu, Fuli Feng, Xiangnan He, Yizhou Sun, and Wei Wang. "Fast Adaptation for Cold-start Collaborative Filtering with Meta-learning". In: *IEEE International Conference on Data Mining (ICDM 2020)*.

[Wol+19] Thomas Wolf et al. *Transformers: State-of-the-art Natural Language Processing*. 2019. arXiv: 1910.03771 [cs.CL].

[Wu+22] Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. "Multi-modal answer validation for knowledge-based vqa". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 3. 2022, pp. 2712–2721.

[WZR20] Sen Wu, Hongyang R. Zhang, and Christopher Ré. "Understanding and Improving Information Transfer in Multi-Task Learning". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=SylzhkBtDB.

[XHW17] Wenhan Xiong, Thien Hoang, and William Yang Wang. "DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 564–573. DOI: 10.18653/v1/D17-1060. URL: https://aclanthology.org/D17-1060.

[Xio+18] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. "One-Shot Relational Learning for Knowledge Graphs". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. 2018, pp. 1980–1990. URL: https://aclanthology.info/papers/D18-1223/d18-1223.

[Yan+18] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. "HotpotQA: A Dataset for

Diverse, Explainable Multi-hop Question Answering". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 2369–2380. DOI: 10.18653/v1/D18-1259. URL: https://aclanthology.org/D18-1259.

[Yan+19a] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. "End-to-End Open-Domain Question Answering with BERTserini". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 72–77. DOI: 10.18653/v1/N19-4013. URL: https://aclanthology.org/N19-4013.

[Yan+19b] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *NeurIPS 2019*. 2019.

[Yan+19c] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 2019, pp. 5754–5764. URL: https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

[Yan+21] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. "An Empirical Study of GPT-3 for Few-Shot Knowledge-Based VQA". In: *ArXiv preprint* abs/2109.05014 (2021). URL: https://arxiv.org/abs/2109.05014.

[Yas+21] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. "QA-GNN: Reasoning with Language Models and Knowledge

Graphs for Question Answering". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, 2021, pp. 535–546. DOI: 10.18653/v1/2021.naacl-main.45. URL: https://aclanthology.org/2021.naacl-main.45.

[Yih+15] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. "Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, 2015, pp. 1321–1331. DOI: 10.3115/v1/P15-1128. URL: https://aclanthology.org/P15-1128.

[Yin+] Da Yin, Liunian Harold Li, Ziniu Hu, Nanyun Peng, and Kai-Wei Chang. "Broaden the Vision: Geo-Diverse Visual Commonsense Reasoning". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*.

[Yoo+] Minji Yoon, John Palowitch, Dustin Zelle, Ziniu Hu, Russ Salakhutdinov, and Bryan Perozzi. "Zero-shot Transfer Learning within a Heterogeneous Graph via Knowledge Transfer Networks". In: *Advances in Neural Information Processing Systems (NeurIPS 2022)*.

[You+18] Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. "GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models". In: *ICML 2018*. 2018.

[Yu+18] Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. "Diverse Few-Shot Text Classification with Multiple Metrics". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana,*

USA, June 1-6, 2018, Volume 1 (Long Papers). 2018, pp. 1206–1215. URL: https://aclanthology.info/papers/N18-1109/n18-1109.

[Yu+20a] Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. "JAKET: Joint Pre-training of Knowledge Graph and Language Understanding". In: *ArXiv preprint* abs/2010.00796 (2020). URL: https://arxiv.org/abs/2010.00796.

[Yu+20b] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. "Gradient Surgery for Multi-Task Learning". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

[Yu+20c] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. "A Survey of Knowledge-Enhanced Text Generation". In: *ArXiv preprint* abs/2010.04389 (2020). URL: https://arxiv.org/abs/2010.04389.

[Yu+22a] Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. "KG-FiD: Infusing Knowledge Graph in Fusion-in-Decoder for Open-Domain Question Answering". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 4961–4974. DOI: 10.18653/v1/2022.acl-long.340. URL: https://aclanthology.org/2022.acl-long.340.

[Yu+22b] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. "CoCa: Contrastive Captioners are Image-Text Foundation Models". In: *CoRR* abs/2205.01917 (2022). DOI: 10.48550/arXiv.2205.

01917. arXiv: 2205.01917. URL: https://doi.org/10.48550/arXiv.2205.01917.

[Yu+22c] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. "Generate rather than Retrieve: Large Language Models are Strong Context Generators". In: *CoRR* abs/2209.10063 (2022). DOI: 10.48550/arXiv.2209.10063. arXiv: 2209.10063. URL: https://doi.org/10.48550/arXiv.2209.10063.

[Yun+19] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. "Graph Transformer Networks". In: *NeurIPS'19*. 2019.

[Zam+18] Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. "Taskonomy: Disentangling Task Transfer Learning". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 3712–3722. DOI: 10.1109/CVPR.2018.00391. URL: http://openaccess.thecvf.com/content%5C_cvpr%5C_2018/html/Zamir%5C_Taskonomy%5C_Disentangling%5C_Task%5C_CVPR%5C_2018%5C_paper.html.

[ZCS] **Ziniu Hu**, Kai-Wei Chang, and Yizhou Sun. "Relation-Guided Pre-Training for Open-Domain Question Answering". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-Finding 2021)*.

[Zem+13] Richard S. Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. "Learning Fair Representations". In: *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*. Vol. 28. JMLR Workshop and Conference Proceedings. JMLR.org, 2013, pp. 325–333. URL: http://proceedings.mlr.press/v28/zemel13.html.

[Zha+19a] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. "Heterogeneous Graph Neural Network". In: *Proceedings of the 25th*

ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019. 2019.

[Zha+19b] Fanjin Zhang et al. "OAG: Toward Linking Large-scale Heterogeneous Entity Graphs". In: *KDD 2019*. 2019.

[Zha+19c] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. "ERNIE: Enhanced Language Representation with Informative Entities". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 1441–1451. DOI: 10.18653/v1/P19-1139. URL: https://aclanthology.org/P19-1139.

[Zha+21a] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. "VinVL: Revisiting Visual Representations in Vision-Language Models". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 5579–5588. DOI: 10.1109/CVPR46437.2021.00553. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Zhang%5C_VinVL%5C_Revisiting%5C_Visual%5C_Representations%5C_in%5C_Vision-Language%5C_Models%5C_CVPR%5C_2021%5C_paper.html.

[Zha+21b] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. "A Survey on Negative Transfer". In: *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS* (2021).

[Zha+22a] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. "Scaling Vision Transformers". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 1204–1213. DOI: 10.1109/CVPR52688.2022.01179. URL: https://doi.org/10.1109/CVPR52688.2022.01179.

[Zha+22b] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. "LiT: Zero-Shot Transfer with Locked-image text Tuning". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 18102–18112. DOI: 10.1109/CVPR52688.2022.01759. URL: https://doi.org/10.1109/CVPR52688.2022.01759.

[Zha+22c] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. "GreaseLM: Graph REASoning Enhanced Language Models for Question Answering". In: *ArXiv preprint* abs/2201.08860 (2022). URL: https://arxiv.org/abs/2201.08860.

[Zhe+18] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. "DAGs with NO TEARS: Continuous Optimization for Structure Learning". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett. 2018, pp. 9492–9503. URL: https://proceedings.neurips.cc/paper/2018/hash/e347c51419ffb23ca3fd5050202f9c3d-Abstract.html.

[Zin+a] **Ziniu Hu**, Ting Chen, Kai-Wei Chang, and Yizhou Sun. "Few-Shot Representation Learning for Out-Of-Vocabulary Words". In: *Proceedings of the Association for Computational Linguistics (ACL 2019)*.

[Zin+b] **Ziniu Hu**, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. "GPT-GNN: Generative Pre-Training of Graph Neural Networks". In: *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2020, Oral)*.

[Zin+c] **Ziniu Hu**, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. "Heterogeneous Graph Transformer". In: *Proceedings of the ACM Web Conference (WWW 2020, mostly cited paper)*.

[Zin+d] **Ziniu Hu**, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A. Ross, and Alireza Fathi. "REVEAL: Retrieval-Augmented Visual-Language Pre-Training with Multi-Source Multimodal Knowledge Memory". In: *Conference on Computer Vision and Pattern Recognition (CVPR 2023, Highlight)*.

[Zin+e] **Ziniu Hu**, Yang Wang, Qu Peng, and Hang Li. "Unbiased LambdaMART: An Unbiased Pairwise Learning-to-Rank Algorithm". In: *Proceedings of the ACM Web Conference (WWW 2019 with US Patent)*.

[Zin+f] **Ziniu Hu**, Yichong Xu, Shuohang Wang, Ziyi Yang, Chengguang Zhu, Kai-Wei Chang, and Yizhou Sun. "Empowering Language Models with Knowledge Graph Reasoning for Question Answering". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2022, also best paper award in SoCal-NLP 2022)*.

[Zin+g] **Ziniu Hu**, Zhe Zhao, Xinyang Yi, Tiansheng Yao, Lichan Hong, Yizhou Sun, and Ed H Chi. "Improving Multi-Task Generalization via Regularizing Spurious Correlation". In: *Advances in Neural Information Processing Systems (NeurIPS 2022, Spotlight)*.

[Zop+16] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. "Transfer Learning for Low-Resource Neural Machine Translation". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. 2016, pp. 1568–1575. URL: http://aclweb.org/anthology/D/D16/D16-1163.pdf.

[Zou+] Difan Zou*, **Ziniu Hu*** (equal contribution), Yewen Wang, Song Jiang, Yizhou Sun, and Quanquan Gu. "Layer-Dependent Importance Sampling for Training Deep and Large Graph Convolutional Networks". In: *Advances in Neural Information Processing Systems (NeurIPS 2019)*.

[ZY18] Yu Zhang and Qiang Yang. “An overview of multi-task learning”. In: *National Science Review* 5.1 (2018), pp. 30–43.