Relational Learning in Pre-Trained Models: A Theory from Hypergraph Recovery Perspective

Yang Chen [1] Cong Fang [1 2] Zhouchen Lin [1 2 3] Bing Liu [4]

Abstract

Foundation Models (FMs) have demonstrated remarkable insights into the relational dynamics of the world, leading to the crucial question: *how do these models acquire an understanding of world hybrid relations?* Traditional statistical learning, particularly for prediction problems, may overlook the rich and inherently structured information from the data, especially regarding the relationships between objects. We introduce a mathematical model that formalizes relational learning as hypergraph recovery to study pre-training of FMs. In our framework, the world is represented as a hypergraph, with data abstracted as random samples from hyperedges. We theoretically examine the feasibility of a Pre-Trained Model (PTM) to recover this hypergraph and analyze the data efficiency in a minimax near-optimal style. By integrating rich graph theories into the realm of PTMs, our mathematical framework offers powerful tools for an in-depth understanding of pre-training from a unique perspective and can be used under various scenarios. As an example, we extend the framework to entity alignment in multimodal learning.

1. Introduction

Foundation Models (FMs) (Bommasani et al., 2021; OpenAI, 2023) have emerged as transformative forces in the realm of artificial intelligence, demonstrating impressive performance in various real-world tasks such as knowledge retrieval (Liu et al., 2023), mathematics problem solving (Frieder et al., 2023), coding (Zhang et al., 2022), commonsense reasoning (Rajani et al., 2019; Zhao et al., 2023b), and text-to-image generation (Ramesh et al., 2021; Li et al., 2023b). During interactions with humans, FMs seem to exhibit an understanding of real-world entities to a certain degree, engaging in reasoning based on these entities (Bubeck et al., 2023). For example, FMs can deduce the entity "table" from descriptions of objects placed on it, such as a cup, book, or computer, which raises a fundamental question: *how do FMs learn real-world entities from pre-training?*

To investigate the learning of entities via pre-training, a formidable challenge is to formalize how the relationships between the entities are learned from data. Traditional statistical learning, such as PAC (Valiant, 1984; Mohri et al., 2018), particularly in classification problems, typically treats data as pairs of objects and their corresponding labels, focusing primarily on predicting these absolute labels. However, this approach may overlook the richer, more nuanced information that data inherently carry, especially regarding the relationships between objects. For instance, an image of a camel does not just represent the animal; it may also encapsulate its context, like a desert background, offering deeper relational insights on the camel and the context objects. Similarly, in natural language processing, the meaning of a sentence transcends the mere sum of its words, revealing complex interdependencies between the entities represented by the words. At the same time, PTMs, such as LLMs, often respond to complex relationships between objects. Recognizing this, a new mathematical model is essential to capture these critical, yet often overlooked, facets of relational learning in pre-training, crucial for understanding the capabilities and generalization of the PTMs.

In this work, we propose a novel mathematical framework based on hypergraph recovery to more fully capture the essence of relational learning. Specifically, we abstract the world as a hypergraph: entities are nodes, and relationships between entities are hyperedges. Each hyperedge is assigned a weight, signifying the strength of the corresponding relation. We formulate relational learning from pre-training as hypergraph recovery of the world hypergraph using the information of data. We model data generation as random sampling from the hyperedges. This data gen-

[1] National Key Lab of General Artificial Intelligence, School of Intelligence Science and Technology, Peking University [2] Institute for Artificial Intelligence, Peking University [3] Pazhou Laboratory (Huangpu), Guangzhou, China [4] Department of Computer Science, University of Illinois Chicago. Correspondence to: Cong Fang <fangcong@pku.edu.cn>, Zhouchen Lin <zlin@pku.edu.cn>.

Proceedings of the 41^st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

eration process mirrors real-world data collection, where a sample represents a perception of a relation between entities, with stronger relations having a higher likelihood of being observed and recorded. Our framework presents two-fold advantages: 1) In contrast to traditional statistical learning, our framework adopts a more nuanced approach. It goes beyond merely capturing individual labels within each data sample, delving into the interrelations between entities. This method yields a richer and more holistic understanding of relational learning in pre-training scenarios. 2) Additionally, the framework integrates rich graph theories into the field of PTMs. This integration invokes powerful analytical tools, providing a novel perspective for relational learning.

Based on the framework, we can answer two important questions about relational learning in PTMs: 1) Identification: Does the data provide sufficient information for relational learning? 2) Data efficiency: If so, what is the essential amount of data required? For the first question, we approach it as an estimation problem within a hypergraph framework and give an affirmative answer by demonstrating that the hypergraph can be identified from sufficient hyperedge samples. To address the second question, we first establish a lower bound $\Omega\left(\frac{m}{\epsilon^2}\right)$ for ϵ-approximate relational learning of the hypergraph with m hyperedges. We further investigate how a model learns relations via Masked Modeling (MM), a common practical pre-training algorithm (Kenton & Toutanova, 2019; He et al., 2022). In the hypergraph recovery framework, an MM PTM learns a set of relative weight ratios between certain entity relations. We show that MM achieves the near-optimal (in terms of approximation error) sample complexity $\tilde{O}\left(\frac{m}{\epsilon^2}\right)$, matching the information theoretical lower bound if logarithmic factors are neglected.

Our hypergraph framework is adaptable to scenarios necessitating the capture of entity relations, including multimodal entity alignment (Chen et al., 2020; Zhao et al., 2023a), social network privacy (Korolova et al., 2008), and relational reinforcement learning (Zambaldi et al., 2018a), etc., allowing for an analysis of key relational learning from pre-training data. We focus on multimodal entity alignment, demonstrating feasible alignment across modalities using sufficient unlabeled data, achieved through hypergraph matching. Although aligning without labeled pairs is theoretically possible, practical computational constraints necessitate labeled pairs to reduce complexity.

We conduct experiments to back up the validity of our hypergraph formulation for relational learning in PTMs. In the first experiment of synthetic relational learning, we create synthetic entities whose relations compose weighted graphs, showing the power of MM for learning the synthetic relations. In the second experiment, we examine real-world relational learning of LLMs by evaluating their relational subgraphs and measuring how well the evaluated subgraphs align with the real world. Our results show that the evaluated relations do align with the real world to some degree and more powerful models exhibit better alignment.

We list the contributions of the paper as follows:

- We propose a new mathematical model to formalize relational learning in PTMs, which is grounded in the principles of hypergraph recovery.
- We demonstrate the feasibility of a learning model achieving relational learning and establish a minimax lower bound for the sample complexity involved. Additionally, we show that pre-training using Masked Modeling (MM) approaches near-optimal data efficiency in terms of approximation error within our framework.
- We extend our framework to entity alignment in multimodal learning. We show the feasibility of entity alignment without labeled pairs and demonstrate the role of labeled pairs in reducing the computational complexity.

2. Related Work

Graph Models. Graphs have long been used to characterize structures of data. For instances, parsing graphs use graphs to represent the grammatical dependencies of text, (Chomsky, 2014; Chen & Manning, 2014; Hewitt & Manning, 2019). Semantic networks model the semantic relationships between words and entities by graphical representations (Miller, 1995; Speer et al., 2017). Knowledge graphs represent knowledge as entities and complex relationships within graphs (Suchanek et al., 2007; Lin et al., 2015; Dettmers et al., 2018). Following a similar philosophy, we model the concepts and the relations in the world as a weighted hypergraph and pre-training data as samples of hyperedges from the hypergraph. Our formulation is, instead, a simplified mathematical model to explain how pre-training can learn the complex relations in the world.

Combinatorial Statistics. Combinatorial statistics studies the statistical properties of data with discrete structures. The most related topic in combinatorial statistics to this work is random graph isomorphism. These works model real-world problems, namely, DNA shotgun assembly (Idury & Waterman, 1995), protein matching (Zaslavskiy et al., 2009), social network privacy (Korolova et al., 2008), etc., by random graph problems such as shotgun assembly (Mossel & Ross, 2017; Ding et al., 2023) and random graph matching (Cullina & Kiyavash, 2016; Barak et al., 2019; Ding et al., 2021), exploiting both the combinatorial and statistical properties of the data. Our work takes a step to build the connections between combinatorial statistics and PTM capabilities, harnessing mathematical tools from the former to enhance our understanding of PTMs.

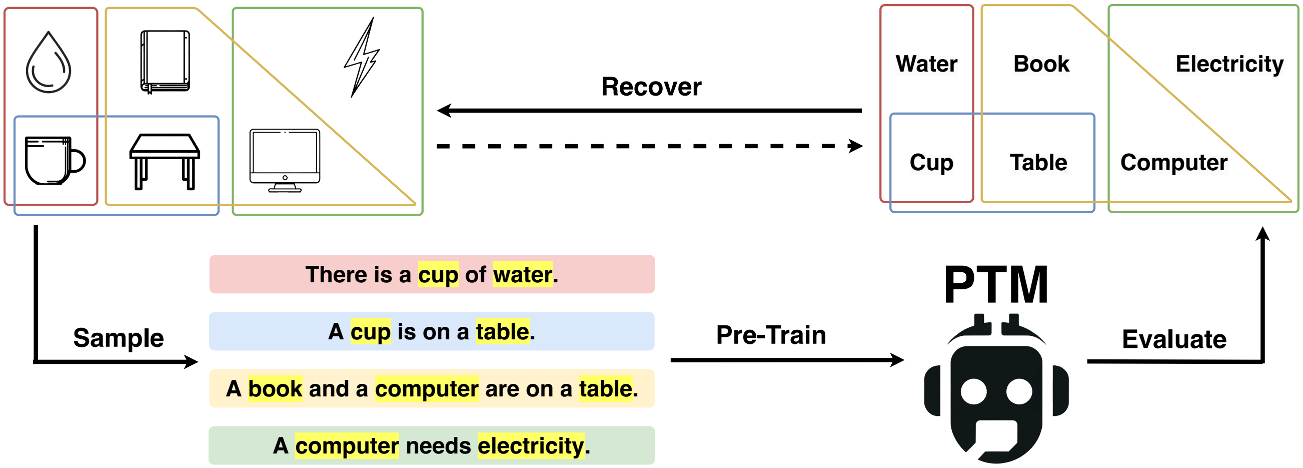


Figure 1. Our hypergraph recovery framework for relational learning in PTMs. The relational model of the world is viewed as a hypergraph. Data are generated by sampling hyperedges from the world relational model and mapping them to perception domains. PTMs learn the entity relations from the data. Recovered relational hypergraphs can be evaluated from the PTMs.

Relational Learning. Relational learning focuses on identifying the relationships among entities (Struyf & Blockeel, 2010). To understand and exploit the relational structure of data, various relational learning techniques and methods are employed, including inductive logic programming (De Raedt, 2008), probabilistic logic learning (De Raedt & Kersting, 2008), relational reinforcement learning (Džeroski et al., 2001; Zambaldi et al., 2018b), graph neural networks (Chen et al., 2021; Fey et al., 2023), etc. While these works aim to capture entity relations more precisely, our research is dedicated to exploring the emergence of relational learning from pre-training in theory.

Theories of PTMs. Various theoretical frameworks have been proposed to elucidate the mechanisms by which PTMs leverage pre-training data and tasks to achieve generalization. Multi-task learning suggests that PTMs acquire generalizable representations through simultaneous training on diverse tasks (Ando et al., 2005; Xie et al., 2020; Hu et al., 2021; Chen et al., 2022; Yang et al., 2022), under the assumption that these representations are the invariant components across the various tasks. Meta-learning posits that PTMs develop the ability to learn efficiently, postulating that certain meta parameters exist that enable fast adaptation to new tasks, with optimization processes geared towards these meta parameters (Finn et al., 2017; 2018; Tripuraneni et al., 2021). In certain in-context learning scenarios, some in-context learning theories propose that PTMs internalize optimization or learning algorithms, facilitating task and distribution generalization (Akyürek et al., 2022; Li et al., 2023a; Von Oswald et al., 2023). This work diverges by explicitly modeling generalizable knowledge as a relational hypergraph of the world, framing pre-training as a process of hypergraph recovery.

3. Preliminary

Hypergraph. A hypergraph $\mathcal{H}$ is a tuple $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a finite set called *nodes* and $\mathcal{E}$ is a family of subsets of $\mathcal{V}$ called *hyperedges* (Bretto, 2013). A weighted hypergraph $\mathcal{H}$, denoted by a tuple $(\mathcal{V}, \mathcal{E}, w)$, is a hypergraph equipped with an additional weight function $w : \mathcal{E} \mapsto \mathbb{R}_{\geq 0}$. The line graph of the hypergraph $\mathcal{H}$, denoted by $L(\mathcal{H})$, is the graph whose node set is the set of the hyperedges of $\mathcal{H}$ and edge set is the set of pairs of the hyperedges that intersect. Consider transformations between hypergraphs. Suppose that $\phi : \mathcal{V} \mapsto \mathcal{V}'$ is a bijection from $\mathcal{V}$ to a set of nodes $\mathcal{V}'$. For a hyperedge $e = \{v_1, \ldots, v_k\}$, we use $\phi(e)$ to denote the hyperedge $\{\phi(v_1), \ldots, \phi(v_k)\}$. We use $\phi(\mathcal{H})$ to denote the hypergraph $\mathcal{H}' = (\mathcal{V}', \mathcal{E}', w')$ where $\mathcal{E}' = \{\phi(e) \mid e \in \mathcal{E}\}$ and $w'(e') = w(\phi^{-1}(e))$. We write $\mathcal{H}_1 \cong \mathcal{H}_2$ if $\mathcal{H}_1$ equals to $\mathcal{H}_2$ up to some bijection, i.e., there exists a bijection ϕ such that $\phi(\mathcal{H}_1) = \mathcal{H}_2$. To measure the differences between two hypergraphs $\mathcal{H}_1 = (\mathcal{V}_1, \mathcal{E}_1, w_1)$ and $\mathcal{H}_2 = (\mathcal{V}_2, \mathcal{E}_2, w_2)$, we consider the following dissimilarity measure

$$d(\mathcal{H}_1, \mathcal{H}_2) = \sum_{e \in \mathcal{E}_1 \cup \mathcal{E}_2} |\bar{w}_1(e) - \bar{w}_2(e)|, \tag{1}$$

where the weight function $\bar{w}_i(e) = w_i(e)$ if $e \in \mathcal{E}_i$ and $\bar{w}_i(e) = 0$ otherwise, $i = 1, 2$. This measure corresponds to the dissimilarity between two graphs constructed from the hypergraphs by the star expansion algorithm (Surana et al., 2021) and captures the hyperedge weight differences between the hypergraphs.

Notation. We use A^* to denote the Kleene closure of set A, i.e., $A^* = \bigcup_{i=0}^{\infty} A^i$ where $A^0 = \{\varepsilon\}$ (the set consisting of only the empty sequence) and $A^i = \{(a_1, \dots, a_i) \mid a_j \in A,\ j = 1, \dots, i\}$. We use $\mathrm{Bij}(A, B)$ to denote the set of all bijections from set A to set B. The notation $O(k)$ (resp., $\Omega(k)$) represents the upper bound (resp., the lower bound) of $C \cdot k$ for some constant C.

4. Hypergraph Recovery Framework

This section introduces a mathematical framework of hypergraph recovery for relational learning in PTMs and how it could emerge from pre-training. We first model the entities and their relations in the world as a weighted hypergraph.

Abstraction 4.1 (Relational Model of the World)**.** The relational model of the world is a hypergraph $\mathcal{H}_0 = (\mathcal{V}_0, \mathcal{E}_0, w_0)$, where each node $v \in \mathcal{V}_0$ represents an entity, each hyperedge $e \in \mathcal{E}_0$ represents a relation between entities, and the weight function $w_0 : \mathcal{E} \mapsto \mathbb{R}$ represents the strength of the relations. Without loss of generality, we assume the weight function is normalized, i.e., $\sum_{e \in \mathcal{E}_0} w_0(e) = 1$. We further assume that $|\mathcal{V}_0| = n$ and $|\mathcal{E}_0| = m$.

Since data is the perception of the world, we formalize the data generation as sampling from the relational hypergraph of the world, as described in Abstraction 4.2.

Abstraction 4.2 (Data Generation)**.** In the data generation process, the entities are mapped to a perception domain (e.g., language and vision). We denote the perception mapping by ϕ_0. In this work, we consider the perception mapping ϕ_0 as a bijection, which keeps the structure of the relational hypergraph $\mathcal{H}_0$. Each data point e is a perception of the relations in the domain, corresponding to a hyperedge sampled i.i.d. from the hypergraph $\phi_0(\mathcal{H}_0)$ according to the weights, i.e., $e \sim P_w(e) = w(e) = w_0(\phi_0^{-1}(e))$.

Under this model, we define relational learning as follows.

Definition 4.3 (Relational Learning)**.** A hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E}, w)$ achieves relational learning for the relational model of the world if $\mathcal{H} \cong \mathcal{H}_0$, i.e., there exists a bijection $\phi : \mathcal{V} \mapsto \mathcal{V}_0$ such that $\phi(\mathcal{H}) = \mathcal{H}_0$.

In practice, we have only finite samples and it is unrealistic to expect that the estimated relational hypergraph is completely the same as the relational model of the world. We further define ϵ-approximate relational learning to consider the approximation error of estimation with finite samples.

Definition 4.4 (ϵ-Approximate Relational Learning)**.** A hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E}, w)$ achieves ϵ-approximate relational learning for the relational model of the world if there exists a bijection $\phi : \mathcal{V} \mapsto \mathcal{V}_0$ such that $d(\phi(\mathcal{H}), \mathcal{H}_0) \leq \epsilon$.

We also say that a model $\mathcal{M}$ achieves (ϵ-approximate) relational learning if we can reconstruct a hypergraph that (ϵ-approximate) relational learning from the model.

Definition 4.5 ((ϵ-Approximate) Relational Learning of Models)**.** A model $\mathcal{M}$ achieves (ϵ-approximate) relational learning if there exists a testing algorithm $\mathcal{A}_{\text{test}} : \mathscr{M} \mapsto \mathscr{H}$ can estimate hypergraphs from models such that $\mathcal{A}_{\text{test}}(\mathcal{M}) = \mathcal{H}_{\mathcal{M}}$ achieves (ϵ-approximate) relational learning. Here, $\mathscr{M}$ and $\mathscr{H}$ denote the sets of all models and all hypergraphs of interest, respectively.

For PTMs, a typical process of relational learning is as follows: a pre-training algorithm $\mathcal{A}_{\text{pre}}$ learns a model $\mathcal{M}$ from a dataset D and a testing algorithm $\mathcal{A}_{\text{test}}$ examines whether the model achieves relational learning, i.e.,

$$\mathcal{H}_0 \xrightarrow{\text{Sample}} D \xrightarrow{\mathcal{A}_{\text{pre}}} M \xrightarrow{\mathcal{A}_{\text{test}}} \mathcal{H}. \tag{2}$$

From the information perspective, whether (ϵ-approximate) relational learning is achievable from a dataset D is equivalent to whether there exists a pre-training algorithm and a testing algorithm that can reconstruct a relational hypergraph equal to the relational hypergraph of the world (up to some bijection). The pre-training algorithm and the testing algorithm are expected to work well for a class of target relational hypergraphs. This goal can be captured by the following minimax formula:

$$\inf_{\mathcal{A}_{\text{pre}}, \mathcal{A}_{\text{test}}} \sup_{\mathcal{H}_0 \in \mathscr{H}_0} d\left(\mathcal{A}_{\text{test}}\left(\mathcal{A}_{\text{pre}}(D)\right), \phi_0\left(\mathcal{H}_0\right)\right) \leq \epsilon, \tag{3}$$

where the $\mathscr{H}_0$ is the set of target relational hypergraphs.

When we consider whether a model pre-trained by a certain algorithm can achieve relational learning, we need to consider how the pre-training algorithm can utilize the data. In this work, we consider Masked Modeling (MM), a common pre-training method that is widely used in various fields. In MM, a model is pre-trained to predict a sample e based on an input e^- that is generated by masked several tokens in e according to a masking strategy $\pi = \pi(e^- \mid e)$.

Abstraction 4.6 (Masked Modeling)**.** Given a masked input e^-, a model $\mathcal{M}$ pretrained by MM complements it and outputs e, reflecting the model's belief $\mathcal{M}(e \mid e^-)$ on

$$P(e \mid e^-) = \frac{w_0(\phi_0^{-1}(e))\pi(e \mid e^-)}{\sum_{e'} w_0(\phi_0^{-1}(e'))\pi(e^- \mid e')}.$$

The model predicts a hyperedge $e \sim \mathcal{M}(e \mid e^-)$. With a slight abuse of notation, we denote the prediction of $\mathcal{M}$ given e^- by $\mathcal{M}(e^-)$.

For two hyperedges e_1, e_2 such that $\pi(e^- \mid e_1) > 0$ and $\pi(e^- \mid e_2) > 0$, we can further infer their relative weights from the MM model $\mathcal{M}$ as $\frac{\hat{w}(e_1)}{\hat{w}(e_2)} = \frac{M(e_1|e^-)\pi(e^-|e_2)}{M(e_2|e^-)\pi(e^-|e_1)}$. To capture such relations between two hyperedges, we define $e_1 \overset{\pi}{\leftrightarrow} e_2$ if there exists a masked hyperedge e^- such that $\pi(e^- \mid e_1) > 0$ and $\pi(e^- \mid e_2) > 0$. For the sake of notational simplicity and in cases where it does not lead to ambiguity, we use $e_1 \leftrightarrow e_2$ without the superscript π. Therefore, under our framework, we can view MM as learning the relative weights between $\leftrightarrow$ related hyperedges.

We also abstract the data generation process of MM.

Abstraction 4.7 (Masked Modeling Data Generation)**.** In the data generation of MM, each hyperedge e_t is first sampled i.i.d. from $P_w(e)$ where $P_w(e) = w_0(\phi_0^{-1}(e))$, for all $t = 1, \dots, N$. For each hyperedge e_t, K masked hyperedges $\{e_{tk}^-\}_{k=1}^K$ are generated i.i.d. by a masking strategy π, i.e., $e_{tk}^- \sim \pi(e_{tk}^- \mid e_{tk})$ where $e_{tk} = e_t$, for all $1 \leq k \leq K$. The dataset for MM is $D = \{(e_{tk}, e_{tk}^-)\}_{1 \leq t \leq N, 1 \leq k \leq K}$.

Under Abstractions 4.6 and 4.7, an MM model $\mathcal{M}$ pre-trained on D with a loss ℓ is

$$\mathcal{M} = \arg\min_{\mathcal{M}' \in \mathscr{M}} \sum_{t=1}^{N} \sum_{k=1}^{K} \ell(\mathcal{M}'(e_{tk}^-), e_{tk}). \tag{4}$$

For an MM pre-trained model to achieve relational learning, it needs to learn relative weights from an MM dataset such that these relative weights amount to the recovery of the relational hypergraph $\mathcal{H}_0$. Denote the MM pre-training algorithm in (4) by $\mathcal{A}_{\text{MM}}$ under Abstractions 4.6 and 4.7. Following (2) and (3), this is to consider

$$\inf_{\mathcal{A}_{\text{test}}} \sup_{\mathcal{H}_0} d\left(\mathcal{A}_{\text{test}}\left(\mathcal{A}_{\text{MM}}(D)\right), \phi_0\left(\mathcal{H}_0\right)\right) \leq \epsilon. \tag{5}$$

5. Main Results for Entity Relational Learning

5.1. Identification

We first consider whether identifying the relational hypergraph $\mathcal{H}_0$ from a pre-training dataset is possible at the population level. The following theorem affirms the feasibility of relational learning if sufficient data are available.

Theorem 5.1 (Identifiability)**.** *Under Abstractions 4.1 and 4.2, suppose that e_t is a generated data sequence. Let D_N be the dataset consisting of the first N elements of the sequences, i.e., $D_N = (e_1, \dots, e_N)$. Then there exist an pre-training algorithm $\mathcal{A}_{pre}$ and a testing algorithm $\mathcal{A}_{test}$, $\mathcal{A} = \mathcal{A}_{test}\left(\mathcal{A}_{pre}(\cdot)\right) : \mathcal{E}^* \mapsto \mathscr{H}$ such that $\mathcal{A}(D_N)$ converges to a hypergraph $\mathcal{H}$ that achieves relational learning as $N \to \infty$ almost surely, i.e., $\mathcal{A}(D_N) \overset{a.s.}{\to} \mathcal{H} \cong \mathcal{H}_0$.*

Theorem 5.1 asserts the asymptotic identifiability of the target hypergraph as the dataset size approaches infinity. The proof of Theorem 5.1 leverages the law of large numbers to show that the distance between the estimated hypergraph and the actual relational hypergraph converges to 0. For detailed proof, refer to Appendix A.

5.2. Data Efficiency

Since relational learning is feasible at the population level, we then consider the data efficiency to achieve ϵ-approximate relational learning at the sample level. We first consider an information theoretical lower bound of the sample complexity to achieve ϵ-approximate relational learning.

Theorem 5.2 (Information Theoretical Lower Bound)**.** *Under Abstractions 4.1 and 4.2 and assuming that the generated dataset D is of size $|D| = N \geq m$ with m sufficiently large, the minimax risk of reconstruction error satisfies*

$$\inf_{\mathcal{A}_{pre}, \mathcal{A}_{test}} \sup_{\mathcal{H}_0} \mathbb{E}_D \left[d(\mathcal{A}_{test}\left(\mathcal{A}_{pre}(D)\right), \phi_0(\mathcal{H}_0))\right] \geq \frac{1}{16}\sqrt{\frac{m}{N}}.$$

Theorem 5.2 presents an information theoretical lower bound $\Omega\left(\frac{m}{\epsilon^2}\right)$ of the sample complexity for ϵ-approximate relational learning. This lower bound is derived from the sample complexity lower of the discrete distribution estimation problem under ℓ_1 distance, by a reduction from the estimation problem to an approximate relational learning problem. The lower bound highlights that the number of the hyperedges m is an important factor in the difficulty of relational learning.

Now we consider the data efficiency of MM to achieve ϵ-approximate relational learning. We assume that the model $\mathcal{M}$ is expressive enough to fit the pre-training data, i.e., for a MM dataset D, the model $\mathcal{M}$ pre-trained on $\mathcal{D}$ satisfies

$$\mathcal{M} = \arg\min \sum_{t=1}^{N} \sum_{k=1}^{K} \ell(\mathcal{M}'(e_{tk}^-), e_{tk}). \tag{6}$$

To characterize the sample complexity, we introduce the following additional assumptions.

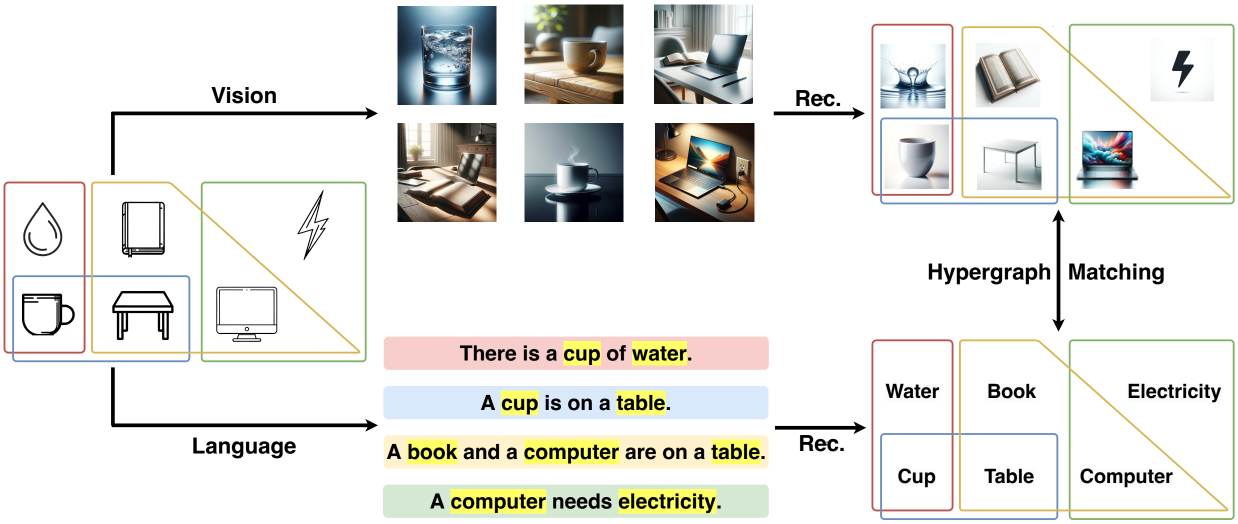
Assumption 5.3 (Range ratio of the weight function)**.** The range ratio of the weight function is $\kappa = \frac{\max_{e \in \mathcal{E}} w(e)}{\min_{e \in \mathcal{E}} w(e)}$.

Assumption 5.4 (Bound on the masking strategy)**.** For each hyperedge $e \in \mathcal{E}$, the support set of masked hyperedges is upper bounded, i.e., $|\operatorname{supp} \pi(\cdot \mid e)| < C_\pi$ for some constant C_π. For each $e \in \mathcal{E}$ and $e^- \in \operatorname{supp} \pi(\cdot \mid e)$, the probability $\pi(e^- \mid e)$ is lower bounded by some constant c_π.

Assumption 5.5 (Bound on the MM path length)**.** For any hyperedges $e, e' \in \mathcal{E}$, there exists a path bounded by L such that $e = e_1 \leftrightarrow e_2 \leftrightarrow \cdots \leftrightarrow e_\ell = e'$.

Assumption 5.3 bounds the weights of each hyperedge within a certain range. Assumption 5.4 bounds the complexity of the masking strategy by limiting the support set of masked hyperedges and setting a minimum probability threshold for potentially masked hyperedges. Assumption 5.5 bounds the connectivity complexity among the hyperedges under the masking strategy.

We analyze the sample complexity for the PTM pre-trained by MM $\mathcal{M}$ to achieve ϵ-approximate relational learning with cross-entropy loss in Theorem 5.6.



There is a cup of water.

A cup is on a table.

A book and a computer are on a table.

A computer needs electricity.

Figure 2. Extension of our hypergraph framework to entity alignment in multimodal learning (taking vision and language for illustration). The relational hypergraphs in different modalities can be reconstructed from data. The entities from different modalities can be aligned by matching the relational hypergraphs. "Rec." represents "Reconstruct".

Theorem 5.6 (Upper Bound by MM)**.** *Suppose that $\mathcal{M}$ is an FM pre-trained by MM on a dataset D with cross-entropy loss. Then $\mathcal{M}$ achieves ϵ-approximate relational learning with probability at least $1-\delta$ if*

$$
\begin{aligned}
K &\geq \frac{2^{14} m^2 \kappa^2 L^2}{c_\pi^2 \epsilon^2} \log \frac{6 m C_\pi}{\delta}, \\
N &\geq \max\left\{ \frac{2m\kappa}{c_\pi} \log \frac{3 m C_\pi}{\delta}, \frac{8m}{\epsilon^2} \log \frac{6m}{\delta} \right\}.
\end{aligned} \tag{7}
$$

In scenarios defined by specific problems and masking strategies, the term $\tilde{O}\left(\frac{m}{\epsilon^2}\right)$ predominates at low approximation errors, especially when $\epsilon = o\left(\sqrt{\frac{c_\pi}{\kappa}}\right)$. This aligns with the information theoretical lower bound $\Omega\left(\frac{m}{\epsilon^2}\right)$ in Theorem 5.2, disregarding the logarithmic factor. This suggests that MM is near-optimal in data efficiency.

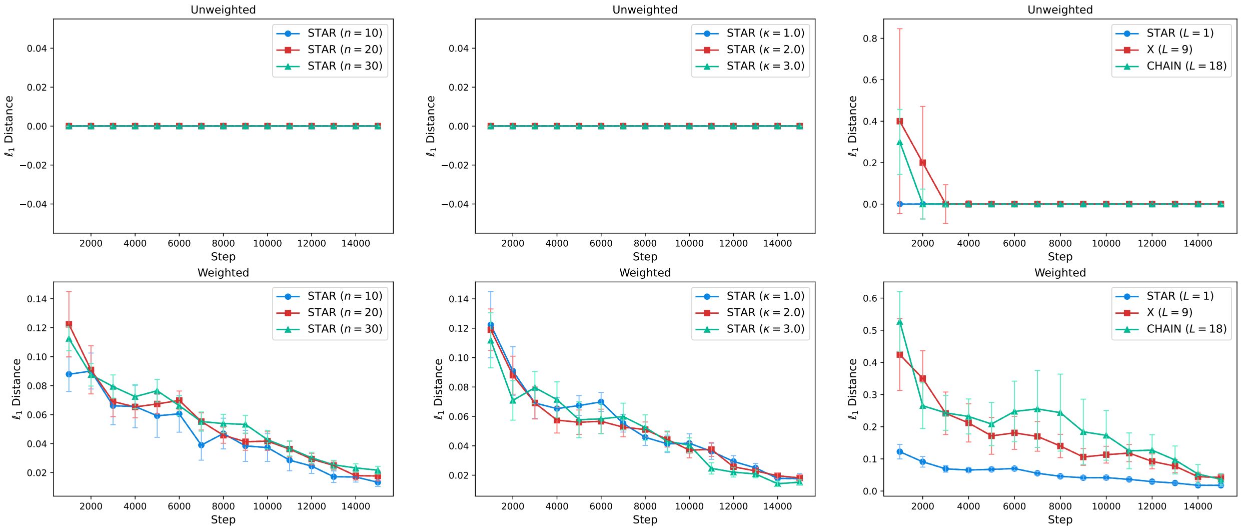
To prove Theorem 5.6, we design an algorithm that computes the relative weights between the pairs of the hyperedges along $e_1 \leftrightarrow \cdots \leftrightarrow e_\ell$ paths. By normalization, we obtain an estimation of the hyperedge weights and further a recovered hypergraph from the relative weights. We show that when the dataset D is sufficiently large, the model $\mathcal{M}$ can learn all the relative weights well enough and therefore the reconstructed hypergraph is a good approximation for the relational hypergraph $\mathcal{H}_0$ (up to some bijection).

Theorem 5.6 reveals that the data efficiency to achieve relational learning is predominantly influenced by three factors: the number of hyperedges m, the range ratio of the weight function κ, and the upper bound of the MM path lengths L. The number of hyperedges m and the range ratio of the weight function κ characterize the complexity of the world relational hypergraph, i.e., the hypergraph with more hyperedges and a larger range ratio requires more samples to be recovered by MM. The MM path length bound L reflects the connectivity under the masking strategy π, influencing how MM learns the relative weights between hyperedges. Efficient recovery of the relational hypergraph is contingent on a small L, indicating well-connected hyperedges; a large L suggests inefficiency in recovery. This aligns with empirical observations that effective MM performance requires masking a sufficient proportion of each sample (He et al., 2022; Wettig et al., 2023).

6. Main Results for Entity Alignment

We further extend our framework to encompass entity alignment within the realm of multimodal learning. In this context, the relational models associated with different modalities are interpreted as distinct representations or "images" of the relational model of the world, each shaped by its unique perception mapping. Although our focus here is on two modalities for illustrative purposes, the principles and methodologies we discuss are readily generalizable to scenarios involving a greater number of modalities.

Concretely, the relational hypergraph in modality i is mapped from $\mathcal{H}_0$ by the perception ϕ_i, i.e., $\mathcal{H}_i = \phi_i(\mathcal{H}_0)$ for $i = 1, 2$. Entity alignment is to find a bijection $\phi \in \mathrm{Bij}(\mathcal{V}_1, \mathcal{V}_2)$ such that $\phi(\mathcal{H}_1) = \mathcal{H}_2$. The data supporting entity alignment consists of three parts: D_1, D_2, and D_2. Here, D_1 and D_2 represent data from the two individual modalities, while D_{12} comprises labeled pairs that denote corresponding relationships across the modalities. For example, in aligning entities between visual and linguistic modalities, the data includes images, text, and labeled



(a) Different numbers of edges.

(b) Different range ratios.

(c) Different MM path lengths.

Figure 3. Evaluation results of synthetic relational learning. (a) STAR graphs with different numbers of edges ($m = n - 1$). (b) STAR graphs with different range ratios. (c) Graphs with different MM path lengths. For each, the experiments are repeated for 5 times and the evaluation results are averaged over the 5 trials.

pairs that link images with their textual descriptions.

Assuming the data from each modality are sufficient, we can recover the relational hypergraphs $\mathcal{H}_1$ and $\mathcal{H}_2$. Entity alignment is achieved by solving the optimization problem:

$$\phi^* = \underset{\phi \in \mathrm{Bij}(\mathcal{V}_1, \mathcal{V}_2)}{\arg\min} \; d(\phi(\mathcal{H}_1), \mathcal{H}_2) \tag{8}$$

Practically, labeled pairs are typically necessary to address the computational difficulty of the graph isomorphism problem in (8), as no polynomial-time solution has been found to date (Babai, 2016; Neuen & Schweitzer, 2018). Labeled pairs are external information that pinpoints partial correspondences between the entities of different modalities, potentially reducing the computational complexity. For example, the labeled pairs can reduce dimensions of Weisfeiler-Lehman methods required (Cai et al., 1992) or prune search trees in individualization-refinement algorithms (McKay & Piperno, 2014) (See Appendix B for further illustration).

When the underlying hypergraph structure has no automorphism, it is possible to align the entities without estimating the weighted relational hypergraph in each domain. For instance, we can first estimate the underlying unweighted hypergraphs and then align the entities by solving the graph isomorphism problem for these unweighted hypergraphs. This approach can enhance relational learning in multimodal models, as the fusion of data from different modalities can complement and augment the information within each modality. Proposition 6.1 describes the information gain brought by the fusion of two modalities.

Proposition 6.1. *Suppose that D_i is the dataset in modality i for $i = 1, 2$. Assume that the entity alignment $\phi^* \in \mathrm{Bij}(\mathcal{V}_1, \mathcal{V}_2)$ has been estimated in prior. Suppose that $\mathcal{M}$ is a multimodal pre-trained model by MM on the datasets D_1 and D_2. Then $\mathcal{M}$ achieves ϵ-approximate relational learning with probability at least $1 - \delta$ if*

$$K_1 + K_2 \geq \frac{2^{14} m^2 \kappa^2 L^2}{c_\pi^2 \epsilon^2} \log \frac{6 m C_\pi}{\delta},$$

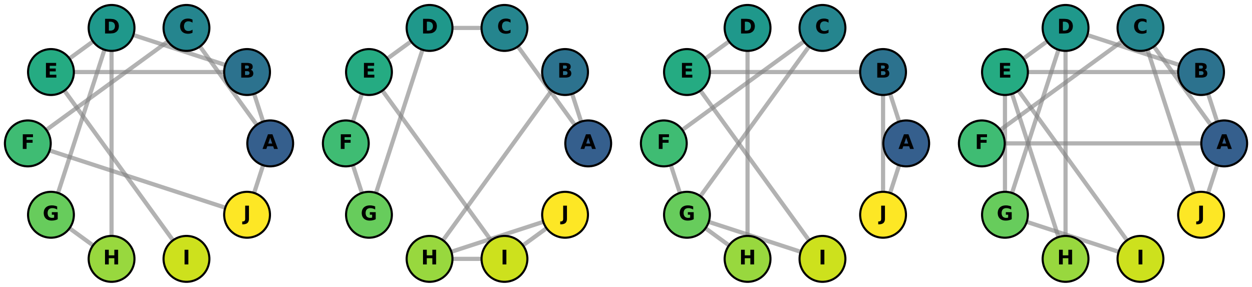
$$N_1 + N_2 \geq \max\left\{ \frac{2 m \kappa}{c_\pi} \log \frac{3 m C_\pi}{\delta}, \frac{8m}{\epsilon^2} \log \frac{6m}{\delta} \right\}.$$

7. Experiments

We conduct two experiments to show empirically that relational learning in PTMs could be seen as relational hypergraph recovery. We consider two settings: synthetic relational learning and real-world relation evaluation.

7.1. Synthetic Relational Learning

In synthetic relational learning, we train PTMs with text consisting of synthetic entities, whose underlying data distribution corresponds to a graph. We show that PTMs can learn the relations between these synthetic entities. To generate data for synthetic relational learning, we first construct a graph, whose nodes are entities (represented by tokens) and edges are relations. We attach edges with random weights and normalize the weights. To generate a training dataset, we sample edges i.i.d. according to the distribution corresponding to the normalized edge weights. We consider masked language modeling (Kenton & Toutanova, 2019). For evaluation, we query the PTM with each synthetic entity to retrieve information about its related entities and the weights of the relations. We reconstruct a graph with the query results and compare the reconstructed graph with



(a) Ground Truth (b) LLAMA-2-70B (c) GPT-3.5 (d) GPT-4

Figure 4. Evaluation results of different LLMs for the real-world relational subgraph generated from the source word "table". We use different letters to represent different entities (see Appendix C.3 for their correspondences). The graphs (from left to right) are the ground truth (extracted from ConceptNet), evaluation results of LLAMA-2-70B, GPT-3.5, and GPT-4, respectively.

Table 1. Summary of the comparison results. The subgraphs are generated from different source entities with $k = 2$ and $d = 3$. The corresponding evaluated graphs are generated from the outputs of different LLMs. The dissimilarity between each pair of the extracted subgraph $\mathcal{H}$ and the estimated graph $\mathcal{H}'$ are measured by their normalized L_1 distance, i.e., $\frac{\|\mathcal{H}-\mathcal{H}'\|_1}{\|\mathcal{H}\|_1}$ where we slightly abuse the notations $\mathcal{H}$ and $\mathcal{H}'$ to denote their adjacent matrices.

	CAKE	DOG	FLY	HUMAN	JACKET	ORANGE	PAPER	SEA	TABLE	ZOO
LLAMA-2-70B	1.00	**0.67**	1.25	**1.00**	1.33	1.33	**0.75**	**0.83**	1.25	1.67
GPT-3.5	**0.67**	1.00	**1.00**	1.25	**1.00**	1.33	**0.75**	**0.83**	1.00	**1.00**
GPT-4	**0.67**	**0.67**	**1.00**	1.50	1.33	**1.00**	**0.75**	**0.83**	**0.75**	1.33

the true underlying graph. We conduct experiments for different graphs, with different numbers of edges, range ratios, and MM path lengths, corresponding to the factors that influence the sample complexity of entity relational learning. More details of the synthetic relational learning experiments can be found in Appendix C.1. The evaluation results are shown in Section 5.2. Our results show that the reconstruction errors of both the unweighted sketch graph and the weighted graph decrease as the training goes on. This the PTMs learn the synthetic relations gradually via MM pre-training. Additionally, the results suggest that larger numbers of edges and larger MM path lengths lead to more steps to converge, which coincides with our theoretical analysis in Theorem 5.6. The effect of the range ratios on the convergence of relational learning is not obvious in our experiments. This may suggest a gap between the theoretical upper bound and the actual convergence rate in the experiments in terms of the range ratio.

7.2. Real-World Relation Evaluation

In real-world relation evaluation, we test whether LLMs such as ChatGPT and GPT-4 learn entities and their relations that align with the real world. We use subgraphs extracted from ConceptNet (Speer et al., 2017) as baselines of the real-world relations graphs. For evaluation, we input the chosen entities to LLMs and ask them to choose top-related ones for each entity. We then construct a graph whose nodes are the entities and edges are those top-related pairs. We compare the subgraph extracted from ConceptNet and the graph evaluated from LLMs. If an LLM learns real-world relations, we expect it to produce a similar graph as the one extracted from ConceptNet. Table 1 summarizes some comparison results of the extracted subgraphs generated by different source entities and the corresponding evaluated graphs. In Figure 4, we visualize the result of the source entity "table". More results are presented in Appendix C.3. We find that GPT-4 achieves the best overall performance among the evaluated LLMs and GPT-3.5 performs slightly better than LLAMA-2-70B. The results suggest different LLMs have different degrees of relational learning and more powerful models seem to understand entity relations better in the sense of relational subgraph reconstruction. Note that we only consider unweighted graphs here because it is difficult to evaluate the relation weights from LLMs accurately. Our results illustrate that the LLMs do organize entities similarly to real-world entities.

8. Conclusion and Outlook

Abstracting the entity relations in the world as a hypergraph, we formalize relational learning in pre-trained models as recovery of the world relational hypergraph. Under the formulation, we show the relational hypergraph is identifiable provided sufficient data at the population level. We also study the sample efficiency and extend the framework to entity alignment in multimodal learning.

While only extending in multimodal learning in this paper, our framework is a general analysis tool. Understanding the capabilities and generalization potential of the PTM is crucial in our field. We would say that PTMs, such as LLMs, often responding to complex relationships between objects, urgently require new mathematical foundations to have a deeper study. This paper paves a new way to study PTM from a unique perspective by capturing the overlooked data information using a hypergraph. Our framework can be potentially used under various scenarios and impacts on application fields. For example, for data and computational efficiency, it is interesting to design more efficient learning algorithms or architectures, such as for multimodal learning. More broadly, for safety, traditional works about adversarial attack and defense theories often focus on several classes that need to be protected. Our framework is not restricted to classification problems and may impose a potential on the entity concept and even human value level. Further, based on the hypergraph, it is promising to understand the reasoning and causality capabilities of PTMs.

Acknowledgements

C. Fang and Z. Lin were supported by National Key R&D Program of China (2022ZD0160300). Z. Lin was additionally supported by the NSF China (No. 62276004) and Qualcomm.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Ando, R. K., Zhang, T., and Bartlett, P. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(11), 2005.

Babai, L. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 684–697, 2016.

Barak, B., Chou, C.-N., Lei, Z., Schramm, T., and Sheng, Y. (nearly) efficient algorithms for the graph matching problem on correlated random graphs. *Advances in Neural Information Processing Systems*, 32, 2019.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Bretto, A. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, 1, 2013.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023.

Cai, J.-Y., Fürer, M., and Immerman, N. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

Chen, D. and Manning, C. D. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 740–750, 2014.

Chen, L., Li, Z., Wang, Y., Xu, T., Wang, Z., and Chen, E. MMEA: entity alignment for multi-modal knowledge graph. In *Knowledge Science, Engineering and Management: 13th International Conference, KSEM 2020, Hangzhou, China, August 28–30, 2020, Proceedings, Part I 13*, pp. 134–147. Springer, 2020.

Chen, Y., Coskunuzer, B., and Gel, Y. Topological relational learning on graphs. *Advances in neural information processing systems*, 34:27029–27042, 2021.

Chen, Y., Jamieson, K., and Du, S. Active multi-task representation learning. In *International Conference on Machine Learning*, pp. 3271–3298. PMLR, 2022.

Chomsky, N. *Aspects of the Theory of Syntax*. Number 11. MIT press, 2014.

Cullina, D. and Kiyavash, N. Improved achievability and converse bounds for erdos-rényi graph matching. *ACM SIGMETRICS performance evaluation review*, 44(1):63–72, 2016.

De Raedt, L. *Logical and relational learning*. Springer Science & Business Media, 2008.

De Raedt, L. and Kersting, K. Probabilistic inductive logic programming. In *Probabilistic inductive logic programming: theory and applications*, pp. 1–27. Springer, 2008.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Ding, J., Ma, Z., Wu, Y., and Xu, J. Efficient random graph matching via degree profiles. *Probability Theory and Related Fields*, 179:29–115, 2021.

Ding, J., Jiang, Y., and Ma, H. Shotgun threshold for sparse erdős–rényi graphs. *IEEE Transactions on Information Theory*, 2023.

Durrett, R. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Džeroski, S., De Raedt, L., and Driessens, K. Relational reinforcement learning. *Machine learning*, 43:7–52, 2001.

Fey, M., Hu, W., Huang, K., Lenssen, J. E., Ranjan, R., Robinson, J., Ying, R., You, J., and Leskovec, J. Relational deep learning: Graph representation learning on relational databases. *arXiv preprint arXiv:2312.04615*, 2023.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. *Advances in neural information processing systems*, 31, 2018.

Frieder, S., Pinchetti, L., Griffiths, R.-R., Salvatori, T., Lukasiewicz, T., Petersen, P. C., Chevalier, A., and Berner, J. Mathematical capabilities of ChatGPT. *arXiv preprint arXiv:2301.13867*, 2023.

Frucht, R. Herstellung von graphen mit vorgegebener abstrakter gruppe. *Compositio Mathematica*, 6:239–250, 1939.

Han, Y., Jiao, J., and Weissman, T. Minimax estimation of discrete distributions. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 2291–2295. IEEE, 2015.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Hewitt, J. and Manning, C. D. A structural probe for finding syntax in word representations. 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)*, pp. 4129–4138, 2019.

Hu, J., Chen, X., Jin, C., Li, L., and Wang, L. Near-optimal representation learning for linear bandits and linear rl. In *International Conference on Machine Learning*, pp. 4349–4358. PMLR, 2021.

Idury, R. M. and Waterman, M. S. A new algorithm for DNA sequence assembly. *Journal of computational biology*, 2 (2):291–306, 1995.

Kenton, J. D. M.-W. C. and Toutanova, L. K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

Korolova, A., Motwani, R., Nabar, S. U., and Xu, Y. Link privacy in social networks. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 289–298, 2008.

Li, Y., Ildiz, M. E., Papailiopoulos, D., and Oymak, S. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, pp. 19565–19594. PMLR, 2023a.

Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., and Lee, Y. J. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22511–22521, 2023b.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Liu, J., Jin, J., Wang, Z., Cheng, J., Dou, Z., and Wen, J.-R. RETA-LLM: A retrieval-augmented large language model toolkit. *arXiv preprint arXiv:2306.05212*, 2023.

McKay, B. D. and Piperno, A. Practical graph isomorphism, ii. *Journal of symbolic computation*, 60:94–112, 2014.

Miller, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

Mossel, E. and Ross, N. Shotgun assembly of labeled graphs. *IEEE Transactions on Network Science and Engineering*, 6(2):145–157, 2017.

Neuen, D. and Schweitzer, P. An exponential lower bound for individualization-refinement algorithms for graph isomorphism. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 138–150, 2018.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Rajani, N. F., McCann, B., Xiong, C., and Socher, R. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4932–4942, 2019.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021. URL https://arxiv.org/abs/2102.12092.

Speer, R., Chin, J., and Havasi, C. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Struyf, J. and Blockeel, H. Relational learning., 2010.

Suchanek, F. M., Kasneci, G., and Weikum, G. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pp. 697–706, 2007.

Surana, A., Chen, C., and Rajapakse, I. Hypergraph dissimilarity measures. *arXiv preprint arXiv:2106.08206*, 2021.

Tripuraneni, N., Jin, C., and Jordan, M. Provable meta-learning of linear representations. In *International Conference on Machine Learning*, pp. 10434–10443. PMLR, 2021.

Valiant, L. G. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Wettig, A., Gao, T., Zhong, Z., and Chen, D. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2977–2992, 2023.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Transformers: State-of-the-art natural language processing, 2020.

Xie, S. M., Kumar, A., Jones, R., Khani, F., Ma, T., and Liang, P. In-n-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. In *International Conference on Learning Representations*, 2020.

Yang, J., Lei, Q., Lee, J. D., and Du, S. S. Nearly minimax algorithms for linear bandits with shared representation. *arXiv preprint arXiv:2203.15664*, 2022.

Zambaldi, V., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D., Lillicrap, T., Lockhart, E., et al. Deep reinforcement learning with relational inductive biases. In *International conference on learning representations*, 2018a.

Zambaldi, V., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D., Lillicrap, T., Lockhart, E., et al. Relational deep reinforcement learning. *arXiv preprint arXiv:1806.01830*, 2018b.

Zaslavskiy, M., Bach, F., and Vert, J.-P. Global alignment of protein–protein interaction networks by graph matching methods. *Bioinformatics*, 25(12):i259–1267, 2009.

Zhang, S., Chen, Z., Shen, Y., Ding, M., Tenenbaum, J. B., and Gan, C. Planning with large language models for code generation. In *The Eleventh International Conference on Learning Representations*, 2022.

Zhao, X., Zeng, W., and Tang, J. Multimodal entity alignment. In *Entity Alignment: Concepts, Recent Advances and Novel Approaches*, pp. 229–247. Springer, 2023a.

Zhao, Z., Lee, W. S., and Hsu, D. Large language models as commonsense knowledge for large-scale task planning. In *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023b.

A. Proof

A.1. Proof of Theorem 5.1

We can consider the combined algorithm $\mathcal{A} = \mathcal{A}_{\text{test}} \circ \mathcal{A}_{\text{pre}}$ directly. We design an algorithm (Algorithm 1) that recovers hypergraphs from dataset and show the reconstructed hypergraph converges to $\mathcal{H}_0$ up to some bijection almost surely by the law of large numbers. Denote the hypergraph recovered from $\mathcal{D}_N$ by $\mathcal{H}_N$. Define random variables $X_N = d(\phi_0^{-1}(\mathcal{H}_N), \mathcal{H}_0)$ for $N = 1, 2, \ldots$. It remains to show $X_N \overset{a.s.}{\to} 0$.

For any $\epsilon > 0$, define

$$E_N := \{\omega \in \Omega : X_N(\omega) > \epsilon\}, \tag{9}$$

where Ω is the sample space.

Let

$$Y_{e,t} = \begin{cases} 1 & x_t = \phi_0(e), \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Then we have

$$\begin{aligned} P(E_N) &= P\left(\sum_{e \in \mathcal{E}_0} \left| \frac{1}{N} \sum_{t=1}^{N} Y_{e,t} - w_0(e) \right| > \epsilon \right) \\ &\leq P\left(\bigcup_{e \in \mathcal{E}_0} \left| \frac{1}{N} \sum_{t=1}^{N} Y_{e,t} - w_0(e) \right| > \frac{\epsilon}{m} \right) \\ &\overset{(a)}{\leq} \sum_{e \in \mathcal{E}_0} P\left(\left| \frac{1}{N} \sum_{t=1}^{N} Y_{e,t} - w_0(e) \right| > \frac{\epsilon}{m} \right) \\ &\overset{(b)}{\leq} 2m \exp\left(-\frac{2N\epsilon^2}{m^2} \right), \end{aligned} \tag{11}$$

where the inequality (a) is due to union bound and the inequality (b) is due to Hoeffding's Inequality.

Notice that

$$\sum_{N=1}^{\infty} P(E_N) \leq \frac{2m \exp\left(-2\epsilon^2/m^2\right)}{1 - \exp\left(-2\epsilon^2/m^2\right)} < \infty. \tag{12}$$

By the first Borel-Cantelli lemma (Durrett, 2019, Chapter 2), we have

$$P\left(\limsup_{N \to \infty} E_N \right) = 0. \tag{13}$$

Equivalently, we have

$$P\left(\lim_{N \to \infty} X_N > \epsilon \right) = 0. \tag{14}$$

Since (14) holds for any $\epsilon > 0$, we have $P(\lim_{n \to \infty} X_N = 0) = 1$, i.e., $X_N \overset{a.s.}{\to} 0$.

A.2. Proof of Theorem 5.2

We prove the information theoretical lower bound by constructing a reduction from finite distribution estimation under ℓ_1 distance to concept understanding.

For any unknown finite distribution $P = (p_1, \ldots, p_m)$ on $\{1, \ldots, m\}$, we construct a world model $\mathcal{H}_0 = (\mathcal{V}_0, \mathcal{E}_0, w_0)$ as follows:

1. $\mathcal{V}_0 = \{v_1, \ldots, v_{m+1}\}$;
2. $\mathcal{E}_0 = \{\{v_1, v_2\}, \ldots, \{v_m, v_{m+1}\}\}$;
3. $w_0(\{v_i, v_{i+1}\}) = p_i$.

Algorithm 1 Hypergraph Estimation from Datasets

```
**Input:** a dataset $D$, a candidate hyperedge set $\mathcal{E}_0$, and a masking strategy $\pi$.

Initialize $\mathcal{E} = \{\}$, $\mathcal{V} =$ , and $\tilde{w} = 0$.
**for** $x \in D$ **do**
    $\mathcal{E} = \mathcal{E} \cup \{x\}$
    $\mathcal{V} = \mathcal{V} \cup x$
    $\tilde{w}(x) = \tilde{w}(x) + 1$
**end for**
Compute $W = \sum_{e \in \mathcal{E}} \tilde{w}(e)$.
$w = \tilde{w}/W$.

Return $\mathcal{H} = (\mathcal{V}, \mathcal{E}, w)$.
```

For a dataset $D' = x_k{}_{k=1}^{N}$ sampled from P. convert it to a dataset $D = \{\{v_{x_k}, v_{x_k+1}\}\}_{k=1}^{N}$ for hypergraph recovery. For an algorithm $\mathcal{A}$, apply it to the dataset D and we obtain an estimation $\mathcal{H} = \mathcal{A}(D) = (\mathcal{V}, \mathcal{E}, w)$ for for the world model $\mathcal{H}_0$. We then compute an estimation P' for the finite distribution P, where $P' = (p'_1, \ldots, p'_m)$ and

$$p'_i = w(\{v_i, v_{i+1}\}). \tag{15}$$

Denote the minimax risk of estimating a finite distribution on $\{1, \ldots, m\}$ with a dataset of size N as $R(m, N)$. Denote the minimax risk of estimating a hypergraph $\mathcal{H}_0$ of m hyperedges with a dataset of size N as $R_{\mathcal{H}}(m, N)$. Then we have

$$\begin{aligned} R(m, N) \leq & \inf_{\mathcal{A}} \sup_{P \in \mathcal{P}_m} \sum_{i=1}^{m} \|p'_i - p_i\| \\ = & \inf_{\mathcal{A}} \sup_{\mathcal{H}_0 \in \mathcal{H}_m} \sum_{e \in \mathcal{E}_0} \|w(e) - w_0(e)\| \\ = & \inf_{\mathcal{A}} \sup_{\mathcal{H}_0 \in \mathcal{H}_m} d(\mathcal{H}, \mathcal{H}_0) \\ = & R_{\mathcal{H}}(m, N), \end{aligned} \tag{16}$$

where the first inequality is due to the definition of the minimax risk $R(m, N)$.

According to Theorem 2 in Han et al. (2015), we have

$$R(m, N) \geq \max_{0 < \zeta \leq 1} F(\zeta), \tag{17}$$

where

$$\begin{aligned} F(\zeta) = & \frac{1}{8} \sqrt{\frac{em}{((1+\zeta)N}} \mathbb{1} \left(\frac{(1+\zeta)N}{m} > \frac{e}{16} \right) \\ & + \exp\left(-\frac{2(1+\zeta)N}{m} \right) \mathbb{1} \left(\frac{(1+\zeta)N}{m} \leq \frac{e}{16} \right) \\ & - \exp\left(-\frac{\zeta^2 N}{24} \right) - 12 \exp\left(-\frac{\zeta^2 m}{32 \ln^2 m} \right). \end{aligned} \tag{18}$$

Combining (16) and (17) and letting $\zeta = 1$, we have

$$R_{\mathcal{H}}(m, N) \geq F(1) \geq \frac{1}{8} \sqrt{\frac{em}{2N}} - \exp\left(-\frac{N}{24} \right) - 12 \exp\left(-\frac{m}{32 \ln^2 m} \right) \geq \frac{1}{16} \sqrt{\frac{m}{N}}. \tag{19}$$

A.3. Proof of Theorem 5.6

Lemma A.1. *Suppose that P_0 is a finite distribution on $[m_0] = \{1, \dots, m_0\}$ whose range ratio is κ_0. Then*

$$\begin{aligned} \min_{i \in [m_0]} P_0(i) &\geq \frac{1}{m_0 \kappa_0} \\ \max_{i \in [m_0]} P_0(i) &\leq \frac{\kappa_0}{m_0 + \kappa_0 - 1} \end{aligned} \tag{20}$$

Proof of Lemma A.1. Let $B_1 := \min_{i \in [m_0]} P_0(i)$ and $B_2 := \max_{i \in [m_0]} P_0(i)$. By the definitions, we have

$$\begin{aligned} B_1 + (m-1)B_2 &\geq 1 \\ B_2 + (m-1)B_1 &\leq 1. \end{aligned}$$

By the definition of range ratio, i.e. $\kappa_0 \frac{B_2}{B_1}$, we further have

$$\begin{aligned} B_1 + (m_0 - 1)\kappa_0 B_1 &\geq 1 \\ B_2 + \frac{m_0 - 1}{\kappa_0} B_2 &\leq 1. \end{aligned}$$

This implies

$$\begin{aligned} B_1 &\geq \frac{1}{m_0 \kappa_0 + 1 - \kappa_0} \geq \frac{1}{m_0 \kappa_0} \\ B_2 &\leq \frac{\kappa_0}{m_0 + \kappa_0 - 1}. \end{aligned}$$

□

Lemma A.2. *Suppose that $\{X_t\}$ is a sequence of random variables sampled i.i.d. from a categorical distribution $\mathrm{Cat}(K, \boldsymbol{p})$ where $\boldsymbol{p} = (p_1, \dots, p_K)$. Then we have*

$$P\left(\sum_{k=1}^{K} \left| \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}\left(X_t = k\right) - p_k \right| \leq \epsilon \right) \geq 1 - \delta \tag{21}$$

if

$$T \geq \frac{2K}{\epsilon^2} \log \frac{2K}{\delta}. \tag{22}$$

Proof of Lemma A.2. Let $S := \sum_{k=1}^{K} \sqrt{p_k(1-p_k)}$ and $\epsilon_k := \frac{\sqrt{p_k(1-p_k)}}{S}\epsilon$ for $k = 1, \dots, K$. Then we have

$$\begin{aligned} & P\left(\sum_{k=1}^{K} \left| \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}\left(X_t = k\right) - p_k \right| \geq \epsilon \right) \\ \overset{(a)}{\leq} & \sum_{k=1}^{K} P\left(\left| \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}\left(X_t = k\right) - p_k \right| \geq \epsilon_k \right) \\ \overset{(b)}{\leq} & \sum_{k=1}^{K} 2 \exp\left(-\frac{T\epsilon_k^2}{2p_k(1-p_k)} \right) \\ \leq & 2K \exp\left(-\frac{T\epsilon^2}{2S^2} \right), \end{aligned} \tag{23}$$

where the inequality (a) is due to union bound and the inequality (b) is due to Chernoff bound.

According to the concavity of the function $f(x) = \sqrt{x(1-x)}$, we have

$$S = K \cdot \frac{1}{K} \sum_{k=1}^{K} f(p_k) \leq K f\left(\frac{1}{K} \sum_{k=1}^{K} p_k \right) = K f\left(\frac{1}{K} \right) = \sqrt{K - 1} < \sqrt{K}. \tag{24}$$

Combining (23) and (24), we obtain the desired result. □

We provide a constructive proof of Theorem 5.6 by designing an algorithm that recover hypergraphs from MM pre-trained models. The algorithm includes two Phases: underlying hypergraph estimation and weight estimation. In Phase 1, we estimate the underlying hypergraph by evaluating the probability of the MM pre-trained model output and selecting all hyperedges of positive probabilities. In Phase 2, we evaluate a sequence of relative weights between the hypergraphs. We estimate the weight function by those relative weights and a normalization. The algorithm is presented in Algorithm 2. Specially, we implement the weight estimation algorithm in a breadth-first style (Algorithm 3). We utilize the data structure queue to implement the algorithm. A queue Q supports two operations: $Q.\operatorname{push\_back}(x)$ that pushes the element x to the back of the queue Q and $Q.\operatorname{pop\_front}(x)$ that removes and returns the front of the queue Q.

Algorithm 2 Hypergraph Estimation from MM Pre-Trained Models

```
Input: a MM pre-trained model $\mathcal{M}$, a candidate hyperedge set $\mathcal{E}_0$, and a masking strategy $\pi$.

// Phase 1: underlying hypergraph estimation
Initialize $\mathcal{E} = \{\}$.
for $e \in \mathcal{E}_0$ do
    Apply $\pi$ to $e$ and get a masked hyperedge $e^-$.
    if $M(e \mid e^-) > 0$ then
        $\mathcal{E} = \mathcal{E} \cup \{e\}$.
    end if
end for
$\mathcal{V} = \cup_{e \in \mathcal{E}} e$.

// Phase 2: weight estimation
Initialize $\tilde{w}(e) = 0$ for all $e \in \mathcal{E}$.
Select $e_0$ from $\mathcal{E}$ and let $\tilde{w}(e_0) = 1$.
$\tilde{w} = \textsc{BFWeightEstimation}(e_0, \mathcal{E}, \mathcal{M}, \pi, \tilde{w})$ (Algorithm 3).
Compute $W = \sum_{e \in \mathcal{E}} \tilde{w}(e)$.
$w = \tilde{w}/W$.

Return $\mathcal{H} = (\mathcal{V}, \mathcal{E}, w)$.
```

Algorithm 3 $\textsc{BFWeightEstimation}(e_{\text{init}}, \mathcal{E}, \mathcal{M}, \pi, \tilde{w})$

```
Input: a selected hyperedge $e_{\text{init}}$, a hyperedge set $\mathcal{E}$, a MM pre-trained model $\mathcal{M}$, a masking strategy $\pi$, and a weight function $\tilde{w}$.

Initialize an empty queue $Q$.
$Q.\operatorname{push\_back}(e_{\text{init}})$.
while $Q$ is not empty do
    $e = Q.\operatorname{pop\_front}()$.
    for $e' \in \mathcal{E}$ such that $e \overset{\pi}{\leftrightarrow} e'$ do
        if $\tilde{w}(e') > 0$ then
            Continue.
        end if
        $\tilde{w}(e') = \frac{\pi(e^-|e)\mathcal{M}(e'|e^-)}{\pi(e^-|e')\mathcal{M}(e|e^-)}\tilde{w}(e)$.
        $Q.\operatorname{push\_back}(e')$.
    end for
end while

Return $\tilde{w}$.
```

We first show that the underlying hypergraph can be recovered with high probability in Phase 1. We denote $\min_{e \in \mathcal{E}_0} w_0(e)$ and $\max_{e \in \mathcal{E}_0} w_0(e)$ by c_w and C_w, respectively. By the definition of the model $\mathcal{M}$, it suffices to show that each hyperedge e and possible masked hypergraphs e^- (i.e., $\pi(e^- \mid e) > 0$) are covered by the training dataset $\mathcal{D}$. According to the data generation process, each sample in the dataset $\mathcal{D}$ corresponds to a pair of (e, e^-) sampled from the distribution $P((e, e^-)) = P_w(e)\pi(e^- \mid e)$. With slight abuse of notation, we write $(e, e^-) \in \mathcal{D}$ if $\mathcal{D}$ contains the corresponding sample of the pair (e, e^-). Denote the support set of $P((e, e^-))$ by S_π. By Assumptions 5.3 and 5.4, we have $|S_\pi| \leq mC_\pi$ and $P(e, e^-) \geq c_w c_\pi$ for all $(e, e^-) \in S_\pi$. Denote the event that the underlying hypergraph $\mathcal{H}_1$ recovered in Phase 1 satisfies $\mathcal{H} \sim \mathcal{H}_0$ by E_1. Then we can obtain

$$\begin{aligned} P(E_1^c) = P\left(\exists (e, e^-) \in S_\pi, (e, e^-) \notin \mathcal{D}\right) \leq & \sum_{(e,e^-)\in S_\pi} P\left((e, e^-) \notin \mathcal{D}\right) \\ \leq & |S_\pi| \min_{(e,e^-)\in S_\pi} P\left((e, e^-) \notin \mathcal{D}\right) \\ \leq & mC_\pi(1 - c_w c_\pi)^N. \end{aligned} \tag{25}$$

We then consider the weight estimation process in Phase 2, supposing that the underlying hypergraph $\mathcal{H}_1$ recovered in Phase 1 satisfies $\mathcal{H} \sim \mathcal{H}_0$ and the isomorphism mapping from $\mathcal{H}$ to $\mathcal{H}_0$ as ϕ. Notice that if we replace $\mathcal{M}$ with $\mathcal{M}_0$ in Algorithm 3, the estimated weight function w satisfies $w(e) = w_0(\phi(e))$ for all $e \in \mathcal{E}$. Since we train by MM with cross-entropy loss, we have

$$\mathcal{M}(e \mid e^-) = \frac{\sum_{t=1}^N \sum_{k=1}^K \mathbb{1}(e_{tk} = e, e^-_{tk} = e^-)}{\sum_{e\in\mathcal{E}} \sum_{t=1}^N \sum_{k=1}^K \mathbb{1}(e_{tk} = e, e^-_{tk} = e^-)}. \tag{26}$$

We first consider only randomness over sampling masked hyperedges for given hyperedges. Denote the number of e in $\{e_t\}_{t=1}^N$ by $f_N(e)$. For any $e \in \mathcal{E}$, $e^- \sim \pi(\cdot \mid e)$ and $\epsilon_1 > 0$, we have

$$\begin{aligned} & P\left(\left|\frac{1}{NK}\sum_{t=1}^N\sum_{k=1}^K \mathbb{1}(e_{tk} = e, e^-_{tk} = e^-) - \frac{f_N(e)}{N}\pi(e^- \mid e)\right| \geq \frac{f_N(e)}{N}\pi(e^- \mid e)\epsilon_1\right) \\ =& P\left(\left|\frac{1}{K}\sum_{k=1}^K\left[\frac{1}{N}\sum_{t=1}^N \mathbb{1}(e_{tk} = e, e^-_{tk} = e^-)\right] - \frac{f_N(e)}{N}\pi(e^- \mid e)\right| \geq \frac{f_N(e)}{N}\pi(e^- \mid e)\epsilon_1\right) \\ \overset{(a)}{\leq}& 2\exp\left[-2K\left(\frac{f_N(e)}{N}\pi(e^- \mid e)\epsilon_1\right)^2\right], \end{aligned} \tag{27}$$

where the inequality (a) is due to Hoeffding's inequality. By union bound, we have

$$\begin{aligned} & P\left(\exists (e, e^-), \left|\frac{1}{NK}\sum_{t=1}^N\sum_{k=1}^K \mathbb{1}(e_{tk} = e, e^-_{tk} = e^-) - \frac{f_N(e)}{N}\pi(e^- \mid e)\right| \geq \frac{f_N(e)}{N}\pi(e^- \mid e)\epsilon_1\right) \\ \leq & \sum_{(e,e^-)} 2\exp\left[-2K\left(\frac{f_N(e)}{N}\pi(e^- \mid e)\epsilon_1\right)^2\right]. \end{aligned} \tag{28}$$

When $\left|\frac{1}{NK}\sum_{t=1}^N\sum_{k=1}^K \mathbb{1}(e_{tk} = e, e^-_{tk} = e^-) - \frac{f_N(e)}{N}\pi(e^- \mid e)\right| \geq \frac{f_N(e)}{N}\pi(e^- \mid e)\epsilon_1$ holds for all pairs of (e, e^-), for any e, e' such that $e \leftrightarrow e'$ with e^- being the common masked hyperedge, we have

$$\begin{aligned} \left|\frac{\tilde{w}(e)}{\tilde{w}(e')} - \frac{f_N(e)}{f_N(e')}\right| =& \left|\frac{\mathcal{M}(e \mid e^-)\pi(e^- \mid e')}{\mathcal{M}(e' \mid e^-)\pi(e^- \mid e)} - \frac{f_N(e)}{f_N(e')}\right| \\ \leq& \left(\frac{1+\epsilon_1}{1-\epsilon_1} - 1\right)\frac{f_N(e)}{f_N(e')} \\ =& \epsilon_2 \frac{f_N(e)}{f_N(e')}, \end{aligned} \tag{29}$$

where $\epsilon_2 := \frac{1+\epsilon_1}{1-\epsilon_1} - 1 = \frac{2\epsilon_1}{1-\epsilon_1}$. This implies

$$(1 - \epsilon_2)\frac{f_N(e)}{f_N(e')} \leq \frac{\tilde{w}(e)}{\tilde{w}(e')} \leq (1 + \epsilon_2)\frac{f_N(e)}{f_N(e')}. \tag{30}$$

By Assumption 5.5, for any $e \in \mathcal{E}$, there exists a path $e_{\text{init}} = e^{(1)} \leftrightarrow \cdots \leftrightarrow e^{(\ell)} = e$, $\ell \le L$ and we have

$$(1-\epsilon_2)^L \frac{f_N(e)}{f_N(e_{\text{init}})} \le \frac{\tilde{w}(e)}{\tilde{w}(e_{\text{init}})} = \tilde{w}(e) \le (1+\epsilon_2)^L \frac{f_N(e)}{f_N(e_{\text{init}})}. \tag{31}$$

Notice that

$$\begin{aligned} w(e) =& \frac{\tilde{w}(e)}{\sum_{e' \in \mathcal{E}} \tilde{w}(e')} \\ =& \frac{\tilde{w}(e)/\tilde{w}(e_{\text{init}})}{\sum_{e' \in \mathcal{E}} \tilde{w}(e')/\tilde{w}(e_{\text{init}})} \\ \in& \left[\frac{(1-\epsilon_2)^L}{(1+\epsilon_2)^L} \cdot \frac{f_N(e)}{N}, \frac{(1+\epsilon_2)^L}{(1-\epsilon_2)^L} \cdot \frac{f_N(e)}{N} \right] \end{aligned} \tag{32}$$

We then obtain

$$\begin{aligned} \|w - w_0 \circ \phi\|_1 =& \sum_{e \in \mathcal{E}} |w(e) - w_0(\phi(e))| \\ =& \sum_{e \in \mathcal{E}} \left| w(e) - \frac{f_N(e)}{N} + \frac{f_N(e)}{N} - w_0(\phi(e)) \right| \\ \le& \sum_{e \in \mathcal{E}} \left| w(e) - \frac{f_N(e)}{N} \right| + \sum_{e \in \mathcal{E}} \left| \frac{f_N(e)}{N} - w_0(\phi(e)) \right| \\ \overset{(a)}{\le}& \left[\frac{(1+\epsilon_2)^L}{(1-\epsilon_2)^L} - 1 \right] \sum_{e \in \mathcal{E}} \frac{f_N(e)}{N} + \sum_{e \in \mathcal{E}} \left| \frac{f_N(e)}{N} - w_0(\phi(e)) \right| \\ \overset{(b)}{=}& \left[\frac{(1+\epsilon_2)^L}{(1-\epsilon_2)^L} - 1 \right] + \sum_{e \in \mathcal{E}} \left| \frac{f_N(e)}{N} - w_0(\phi(e)) \right|, \end{aligned} \tag{33}$$

where the inequality (a) is due to (32) and the equality (b) is due to $\sum_{e \in \mathcal{E}} f_N(e) = N$. Note that $\frac{(1+\epsilon_2)^L}{(1-\epsilon_2)^L} - 1 \le \frac{\epsilon}{2}$ if $\epsilon_1 \le \frac{\epsilon}{64L}$ for ϵ sufficiently small. By (33) and Lemma A.2, with $\epsilon_1 = \frac{\epsilon}{64L}$, we have

$$\begin{aligned} & P\left(E_1 \wedge \|w - w_0 \circ \phi\|_1 \ge \epsilon\right) \\ \le& P\left(\sum_{e \in \mathcal{E}} \left| \frac{f_N(e)}{N} - w_0(\phi(e)) \right| \ge \frac{\epsilon}{2} \right) + P\left(\sum_{e \in \mathcal{E}} \left| \frac{f_N(e)}{N} - w_0(\phi(e)) \right| \le \frac{\epsilon}{2} \right. \\ & \left. \wedge \exists (e, e^-), \left| \frac{1}{NK} \sum_{t=1}^{N} \sum_{k=1}^{K} \mathbb{1}(e_{tk} = e, e_{tk}^- = e^-) - \frac{f_N(e)}{N} \pi(e^- \mid e) \right| \ge \frac{f_N(e)}{N} \pi(e^- \mid e) \epsilon_1 \right) \\ \le& \sum_{(e,e^-)} 2 \exp\left[-2K \left(\frac{f_N(e)}{N} \pi(e^- \mid e) \epsilon_1 \right)^2 \right] + 2m \exp\left(-\frac{N\epsilon^2}{8m} \right) \\ \overset{(a)}{\le}& \sum_{(e,e^-)} 2 \exp\left[-2K \left(\frac{c_w}{2} \pi(e^- \mid e) \epsilon_1 \right)^2 \right] + 2m \exp\left(-\frac{N\epsilon^2}{8m} \right) \\ \le& 2mC_\pi \exp\left[-2K \left(\frac{c_w c_\pi}{128L} \epsilon \right)^2 \right] + 2m \exp\left(-\frac{N\epsilon^2}{8m} \right), \end{aligned} \tag{34}$$

where the inequality (a) is due to $\frac{f_N(e)}{N} \ge c_w - \frac{\epsilon}{2} \ge \frac{c_w}{2}$ when $\sum_{e \in \mathcal{E}} \left| \frac{f_N(e)}{N} - w_0(\phi(e)) \right| \le \frac{\epsilon}{2}$ holds and ϵ is sufficiently small.

Combining (25) and (34), we have

$$\begin{aligned} & P\left(\|w - w_0 \circ \phi\|_1 \leq \epsilon\right) \\ \geq & 1 - P(E_1^c) - P(E_1 \wedge \|w - w_0 \circ \phi\|_1 \geq \epsilon) \\ \geq & 1 - mC_\pi(1 - c_w c_\pi)^N - 2mC_\pi \exp\left[-2K\left(\frac{c_w c_\pi}{128L}\epsilon\right)^2\right] - 2m \exp\left(-\frac{N\epsilon^2}{8m}\right) \\ \geq & 1 - \delta, \end{aligned} \tag{35}$$

if

$$\begin{aligned} mC_\pi(1 - c_w c_\pi)^N &\leq \frac{\delta}{3}, \\ 2mC_\pi \exp\left[-2K\left(\frac{c_w c_\pi}{128L}\epsilon\right)^2\right] &\leq \frac{\delta}{3}, \\ 2m \exp\left(-\frac{N\epsilon^2}{8m}\right) &\leq \frac{\delta}{3}. \end{aligned} \tag{36}$$

After simplification, we have

$$\begin{aligned} K &\geq \frac{2^{14} m^2 \kappa^2 L^2}{c_\pi^2 \epsilon^2} \log \frac{6mC_\pi}{\delta}, \\ N &\geq \max\left\{\frac{2m\kappa}{c_\pi} \log \frac{3mC_\pi}{\delta}, \frac{8m}{\epsilon^2} \log \frac{6m}{\delta}\right\}. \end{aligned} \tag{37}$$

A.4. Proof of Proposition 6.1

Proposition 6.1 is directly implication of Theorem 5.6 in the multimodal model with the prior entity alignment ϕ^*. More concretely, we can generate a dataset $D' = \phi^*(D_1) \cup D_2$ with $N' = N_1 + N_2, K' = K_1 + K_2$ by the entity alignment ϕ^*. Applying Theorem 5.6 to the dataset D', we obtain Proposition 6.1.

B. Entity Alignment

While we show that entity alignment is feasible without labeled pairs in theory, labeled pairs are important in practice. A possible reason is that solving the entity alignment problem is computational challenging, no known polynomial algorithms addressing the problem. The role of the labeled pairs might be reducing the inherent complexity required to solve the computational problem. Here are two examples of how the labeled pairs can help to solve the alignment problem more efficiently.

Example B.1. When all m labeled pairs for the hyperedges are available, we can efficiently determine the alignment mapping between entities by leveraging hyperedges as identifiers. More concretely, we assign a unique number as the identifier to each hyperedge. Subsequently, each node is labeled with a tuple containing the identifiers of the hyperedges it belongs to, arranged in descending order. The nodes within each hypergraph are then organized into sequences based on their lexicographic order. Correspondence between entities is established through the alignment of nodes at identical positions within these sequences. The entire alignment process is of computational complexity $\tilde{O}(mn)$.

Example B.1 shows that we can align entities efficiently given all m labeled pairs for the hyperedges. This also means that as long as we can find the graph matching between the line graphs of the hypergraphs, we can also align the hypergraphs with only polynomial extra computational overhead. Therefore, we can focus on the graph matching problem of the line graphs of the hypergraphs.

WL test serves as a potent heuristic for graph matching, demonstrating efficacy across a wide range of graphs. Nonetheless, certain graphs challenge the capabilities of low-dimensional WL tests, leading to their failure (Cai et al., 1992). Although higher-dimensional WL tests may achieve accurate graph matching, they impose significantly greater computational demands. Labeled pairs could help to overcome this dilemma.

Example B.2. Frucht graph (Figure 5) is a regular graph without non-trivial automorphism (Frucht, 1939). 1-WL does not work for Frucht graph because of its regularity. While higher-dimensional WL tests are applicable, they are significantly less efficient. However, if a labeled pair is identified, one can exclude the nodes in the label pair from both graphs and apply the 1-WL test to the resulting subgraphs, leading to efficient graph matching.

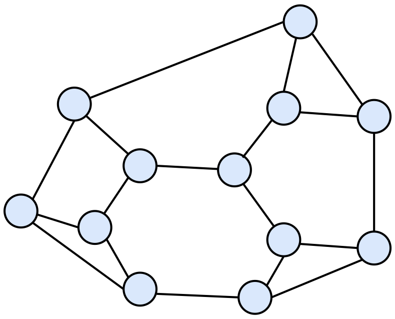


Figure 5. Frucht graph.

C. Experiments

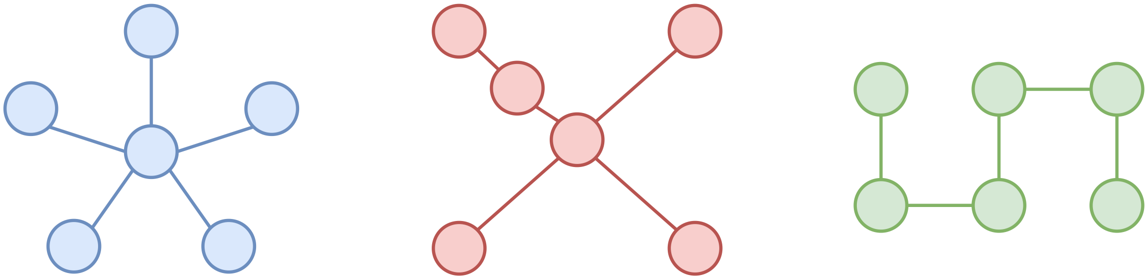
C.1. Synthetic Relational Learning

C.2. Data

C.2.1. Graph Structures

When the number of nodes is n, the different graph structures (Figure 6) are

- STAR:
  - $\mathcal{V} = \{0, 1, \ldots, n-1\}$;
  - $\mathcal{E} = \{\{0, i\} \mid i = 1, \ldots, n-1\}$;
- X:
  - $\mathcal{V} = \{0, 1, \ldots, n-1\}$;
  - $\mathcal{E} = \{\{0, k\} \mid k = 1, 2, 3, 4\} \cup \{\{4i + k, 4i + k + 4\} \mid 4i + k + 4 \leq n - 1\}$;
- CHAIN:
  - $\mathcal{V} = \{0, 1, \ldots, n-1\}$;
  - $\mathcal{E} = \{\{i, i+1\} \mid i = 0, \ldots, n-2\}$.



(a) STAR.

(b) X.

(c) CHAIN

Figure 6. Different graph structures ($n = 6$).

C.2.2. Data Generation

Each node of the graph is attached with a token, starting from “a” and following the order of tokens of BERT’s tokenizer. Each edge is assigned a weight, sampled from $\{w_{\min}, w_{\max}\}$. Specifically, we use $w_{\min} = 1.0, w_{\max} = 1.0$ for $\kappa = 1.0$, $w_{\min} = 1.0, w_{\max} = 10.0$ for $\kappa = 10.0$, and $w_{\min} = 1.0, w_{\max} = 100.0$ for $\kappa = 100.0$ in our experiments. Then the weights of the graph are normalized. When generating data, we first sample an edge from the graph, with probability proportional to the the weights. We then concatenate the tokens of the edges with a random order. Tokens are separated by spaces to avoid that they are combined by the tokenizer. For each graph, we generate 100000 samples for each graph, with 80000 samples for training, 10000 samples for validation, and 10000 samples for testing.

C.2.3. Model

We choose BERT as our underlying PTM. We use the implementation of HuggingFace (Wolf et al., 2020) with the default tokenizer and the default configuration of BERT.

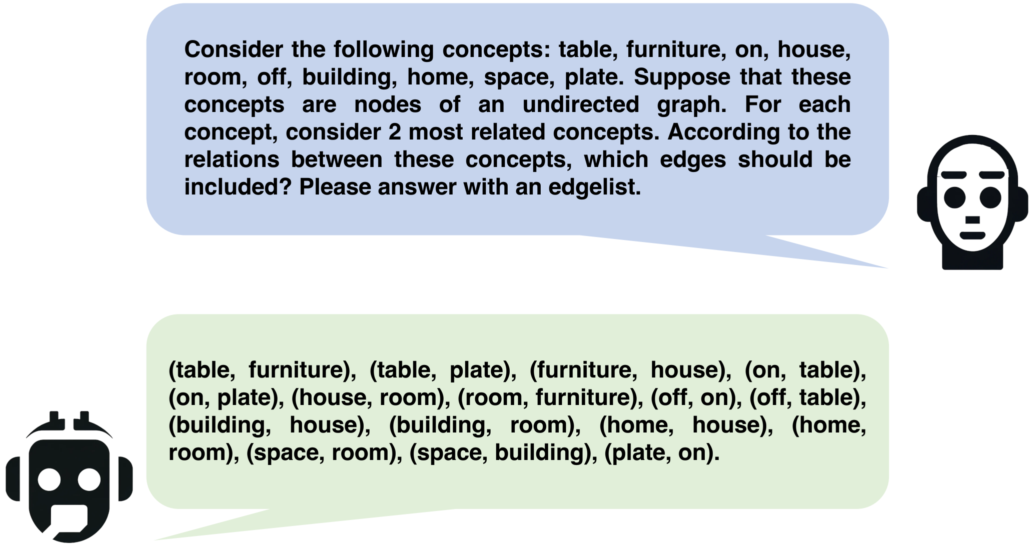
C.2.4. Pre-Training

We pre-train our model by MLM from scratch. For the masking strategy, we mask one of the tokens in a sample uniformly at random. We train the model by AdamW, with the initial learning rate 2×10^{-5}, weight decay 0.01, the cosine scheduler. The other hyperparameters of AdamW are the same as the default of HuggingFace TrainerArguments. We pre-train the model for 100 epochs. Per-device training batch size is 256. The experiments are run on a server with Ubuntu. All the models are trained on two NVIDIA GeForce RTX 3090 GPUs.

C.3. Real-World Relation Evaluation

To extract a subgraph from ConceptNet, we first choose a source entity, query for the k most related entities, and then repeat such a process for the returned entities. We adopt a breadth-first-search-like generation process to choose a subset of entities and construct a subgraph by considering k most related entities within these chosen entities and the generation process is limited to some depth d (the source entity are of depth 0).

In the real-world relation evaluation, we assess the LLMs' understanding of entity relationships by querying the k most related entities within a specified set. We employ the prompt: "Consider the following concepts: [ENTITIES]. Suppose that these concepts are nodes of an undirected graph. For each concept, consider [k] most related concepts. According to the relations between these concepts, which edges should be included? Please answer with an edgelist.", where "[ENTITIES]" and "[k]" are placeholders for the actual entity set and the number of top-related entities, respectively. The LLMs will generate responses comprising edgelists, potentially accompanied by additional text, which are then utilized to construct relational graphs. See Figure 7 for an illustration. These graphs are compared with the corresponding subgraphs extracted from ConceptNet.



Consider the following concepts: table, furniture, on, house, room, off, building, home, space, plate. Suppose that these concepts are nodes of an undirected graph. For each concept, consider 2 most related concepts. According to the relations between these concepts, which edges should be included? Please answer with an edgelist.

(table, furniture), (table, plate), (furniture, house), (on, table), (on, plate), (house, room), (room, furniture), (off, on), (off, table), (building, house), (building, room), (home, house), (home, room), (space, room), (space, building), (plate, on).

Figure 7. An example of real-world relation evaluation using GPT-4. Here, "table" is the source entity, with $k = 2$ indicating the two most closely related entities considered for generating the relational graph.

Figures 8-17 are the evaluation results for all the source entities listed in Table 1.

The correspondences between the entities and the letters used in the above figures are summarized in Tables 2 and 3.

D. Additional Experiments

D.1. Additional Synthetic Relational Learning

To show the applicability of our method to general graphs and its scalability to graphs of different magnitudes we conduct additional synthetic relational learning in random graphs and subgraphs extracted from ConceptNet.

Table 2. The correspondences between the entities and the letters for ConceptNet (Part 1).

	A	B	C	D	E	F
CAKE	CAKE	BIRTHDAY	DESSERT	CELEBRATION	LIZARD	PARTY
DOG	DOG	BARK	HOUSE	TREE	BUILDING	HOME
FLY	FLY	INSECT	BUG	FLEA	MEADOW	WIRETAP
HUMAN	HUMAN	SCHOOL	HOME	LEARN	PLACE	HOUSE
JACKET	JACKET	COAT	SHELL	CLOSET	MATERIAL	HUSK
ORANGE	ORANGE	FRUIT	PEEL	EAT	YOU	SKIN
PAPER	PAPER	WRITE	SHEET	PEN	BED	CLOSET
SEA	SEA	OCEAN	WATER	SAIL	LAKE	DRINK
TABLE	TABLE	FURNITURE	ON	HOUSE	ROOM	OFF
ZOO	ZOO	ANIMAL	ELEPHANT	SQUIRREL	CIRCUS	TRUNK

Table 3. The correspondences between the entities and the letters for ConceptNet (Part 2).

	G	H	I	J	K	L
CAKE	GARDEN	ROCK	-	-	-	-
DOG	PLANT	GROW	TOWN	BANK	PLACE	-
FLY	DOG	WOOD	HAYFIELD	INVESTIGATION	TAP	-
HUMAN	STUDY	KNOWLEDGE	LOCATION	BED	BUILDING	-
JACKET	BEDROOM	CLOTHES	WOOD	WOOL	CHAFF	-
ORANGE	FOOD	HUNGER	ME	BODY	MOLE	-
PAPER	OFFICE	POCKET	SLEEP	FURNITURE	BEDROOM	CLOTHES
SEA	BOAT	WIND	POND	LIQUID	BEVERAGE	-
TABLE	BUILDING	HOME	SPACE	PLATE	-	-
ZOO	RODENT	BALLOON	ATTIC	CAR	-	-

D.1.1. Random Graphs

We address synthetic relational learning tasks in random graphs of varying node counts. Specifically, we generate weighted connected random graphs (WCGNM) with n nodes and $m(n) = \frac{pn(n-1)}{2}$ edges, selected uniformly at random. We vary n across five different magnitudes: $10, 20, 50, 100, and 200$, maintaining parameters $p = 0.2$ and $\kappa = 3.0$ for each. Each experimental setting is repeated 5 times. The results are presented in Figure 18.

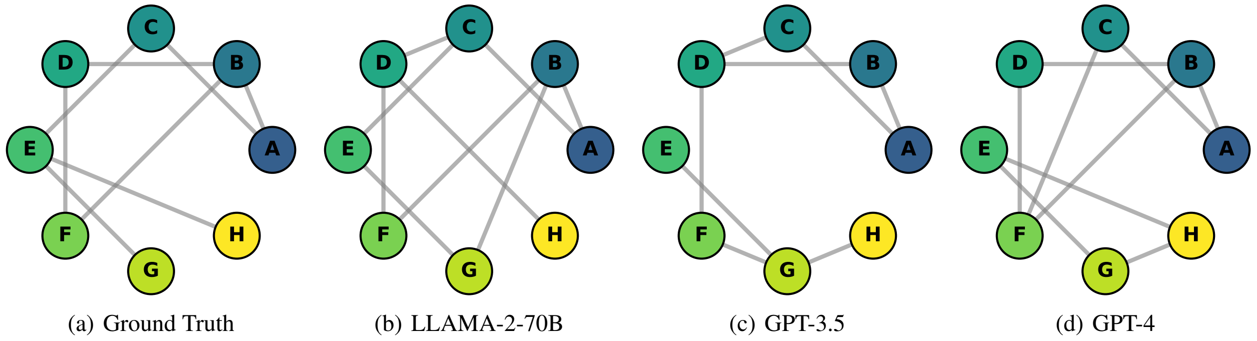
D.1.2. Subgraphs Extracted from ConceptNet

We conduct synthetic relational learning tasks using relational graphs derived from ConceptNet, which represent more intricate real-world relational structures. These subgraphs are generated similarly to the real-world relation evaluation experiments in Section 7.2 but include additional top-related pairs for each entity to increase complexity. Specifically, we focus on the three most related pairs of each entity. Each resulting subgraph comprises approximately 50 nodes, making them more complex than the specific structured graphs used in the experiments of Section 7.1. The results are shown in Figure 19.

D.2. Additional Real-World Relation Evaluation

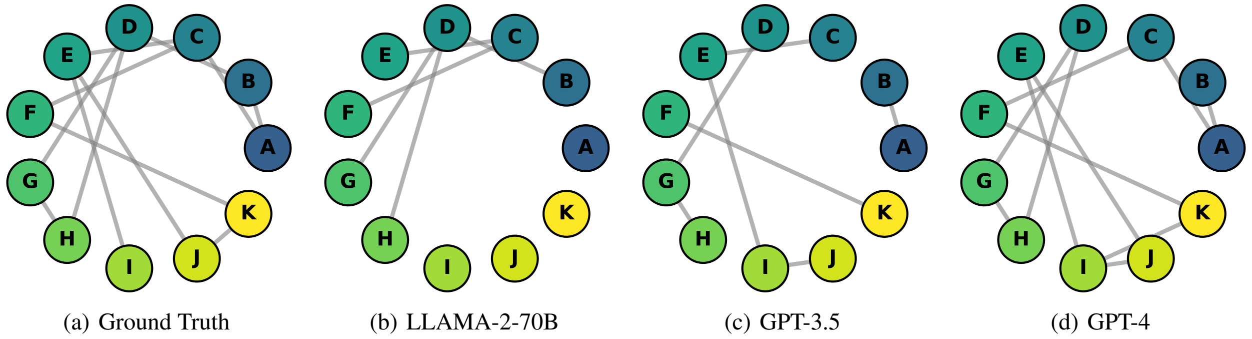
D.2.1. Relation Evaluation in WordNet

We perform similar experiments to that in Appendix C.3 in WordNet (Miller, 1995) to show that our method can be applied to relational learning scenarios beyond ConceptNet. Figures 20 - 24 are the evaluation results, which are summarized in Table 6. The correspondences between the entities and the letters used in the figures are summarized in Tables 4 and 5.



(a) Ground Truth (b) LLAMA-2-70B (c) GPT-3.5 (d) GPT-4

Figure 8. Cake.

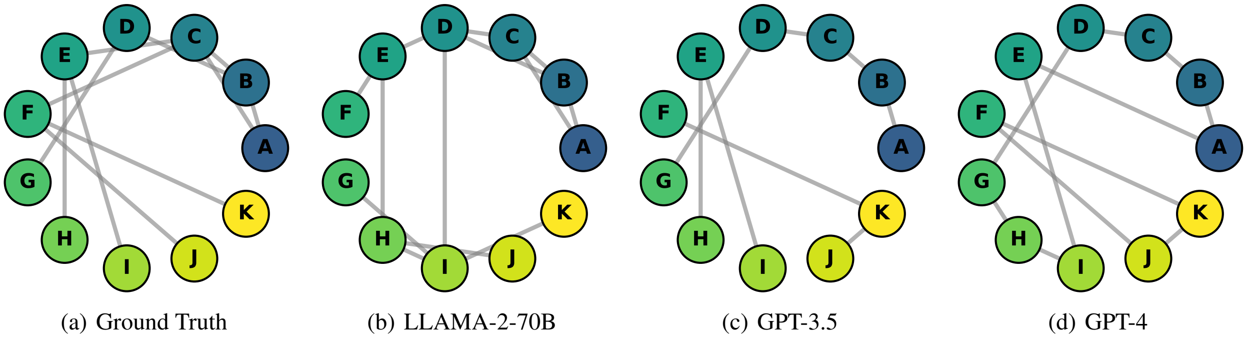


(a) Ground Truth (b) LLAMA-2-70B (c) GPT-3.5 (d) GPT-4

Figure 9. Dog.

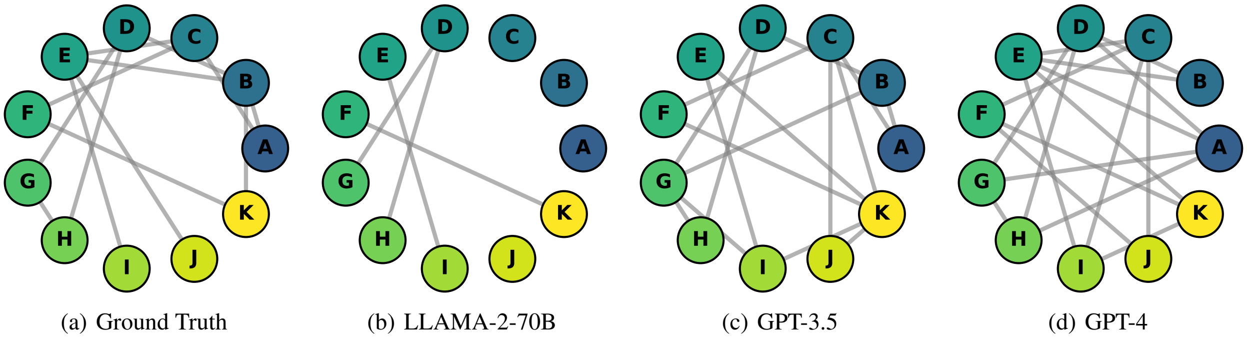
D.2.2. Relation Evaluation with Hypergraphs

We focus on identifying pairwise relationships rather than the full hypergraph structures in Section 7.2 primarily for the ease of making fair comparisons between the evaluated relations and the ground truth. To the best of our knowledge, there is no widely used database that characterizes entity relations in the form of hypergraphs. Although the evaluated hypergraphs cannot be directly compared with an established ground truth, the recovered relations align with our common knowledge. The PTMs are only asked to identify the two most related entities in Section 7.2. We slightly adapt the prompts to make them recover the full hypergraph structures, instructing the PTMs to directly output the lists of hyperedges for the extracted entities in the prompts. We find that the PTMs are capable of reconstructing relational hypergraph structures that align with our existing knowledge. Figures 25 - 34 are the evaluation results. The correspondences between the entities and the letters used in the figures are summarized in Tables 2 and 3.



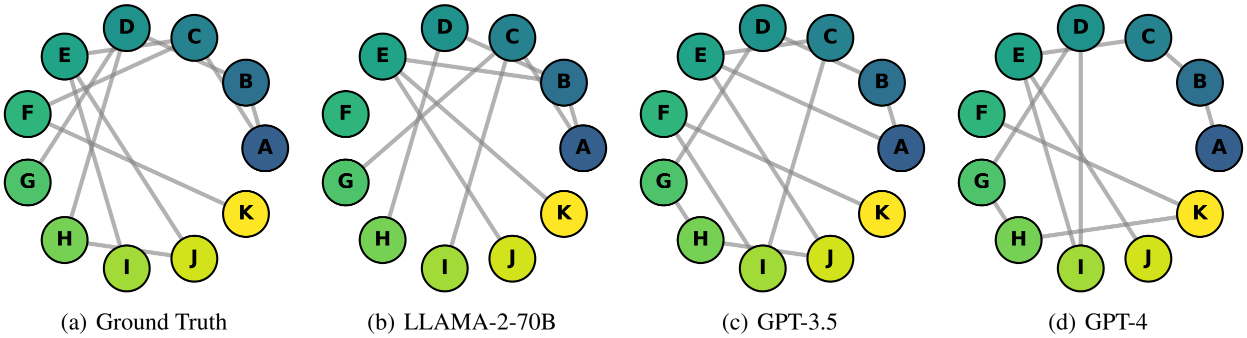
(a) Ground Truth (b) LLAMA-2-70B (c) GPT-3.5 (d) GPT-4

Figure 10. Fly.



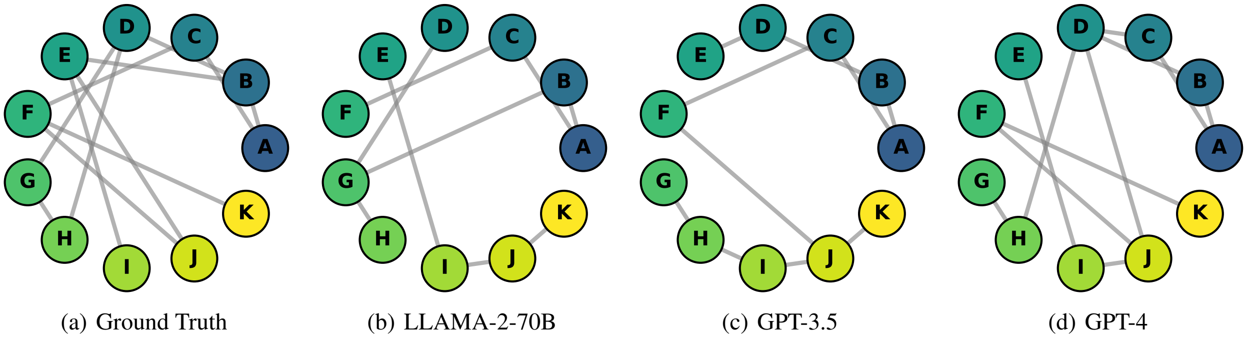
(a) Ground Truth (b) LLAMA-2-70B (c) GPT-3.5 (d) GPT-4

Figure 11. Human.



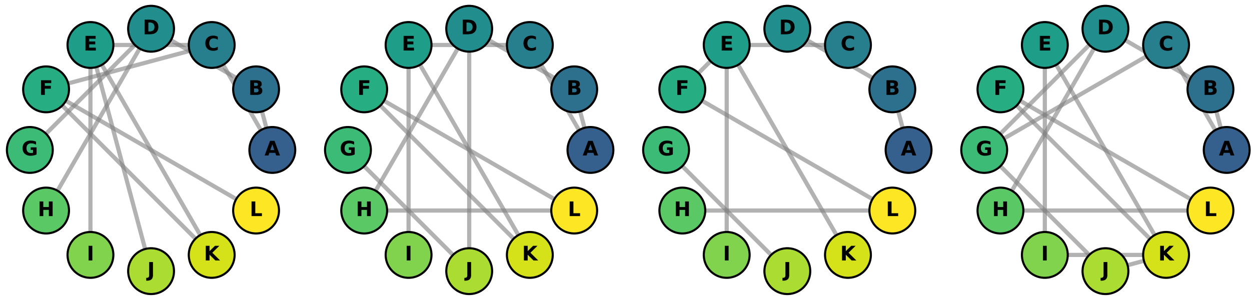
(a) Ground Truth (b) LLAMA-2-70B (c) GPT-3.5 (d) GPT-4

Figure 12. Jacket.



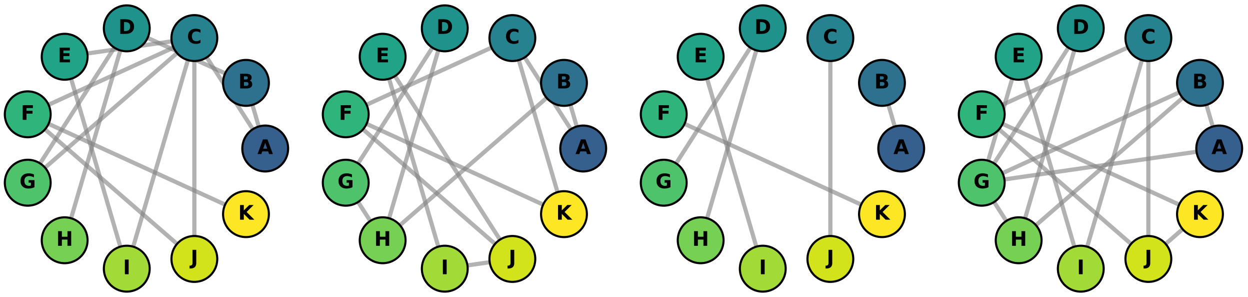
(a) Ground Truth (b) LLAMA-2-70B (c) GPT-3.5 (d) GPT-4

Figure 13. Orange.



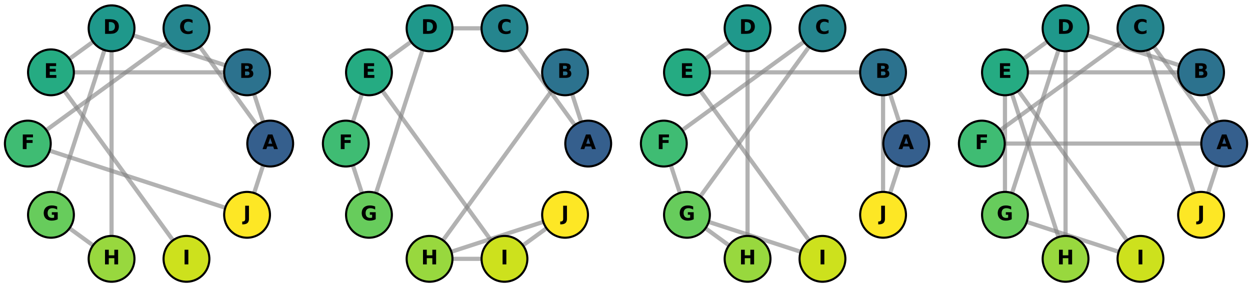
(a) Ground Truth (b) LLAMA-2-70B (c) GPT-3.5 (d) GPT-4

Figure 14. Paper.



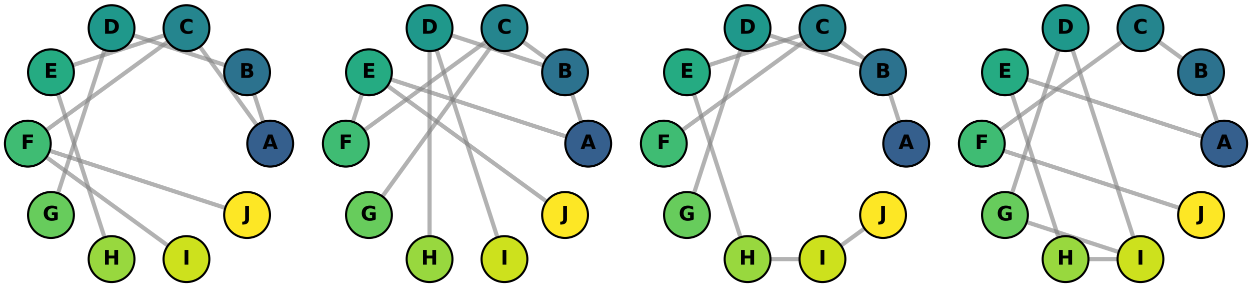
(a) Ground Truth (b) LLAMA-2-70B (c) GPT-3.5 (d) GPT-4

Figure 15. Sea.



(a) Ground Truth (b) LLAMA-2-70B (c) GPT-3.5 (d) GPT-4

Figure 16. Table.



(a) Ground Truth (b) LLAMA-2-70B (c) GPT-3.5 (d) GPT-4

Figure 17. Zoo.

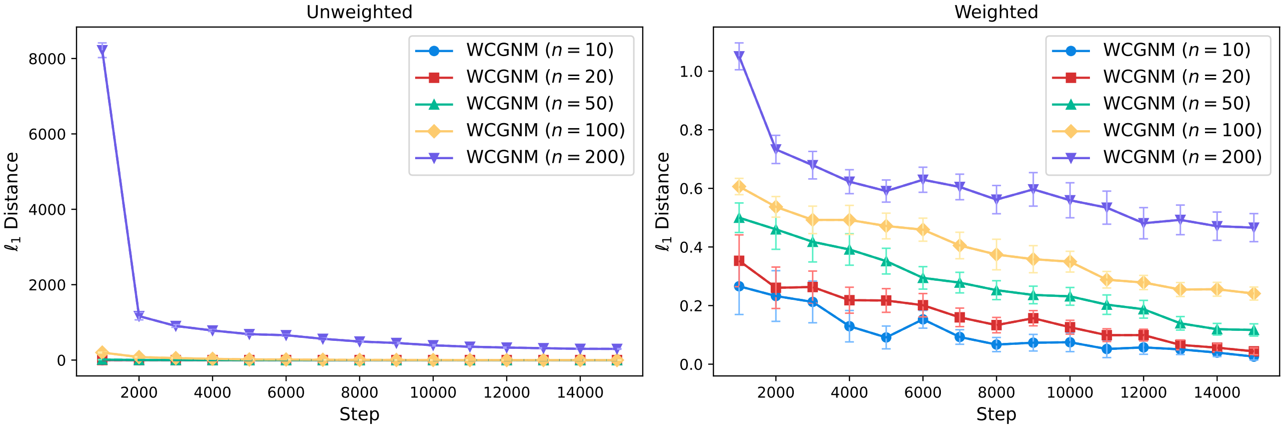


Figure 18. Synthetic relation learning in WCGNMs with different magnitudes.

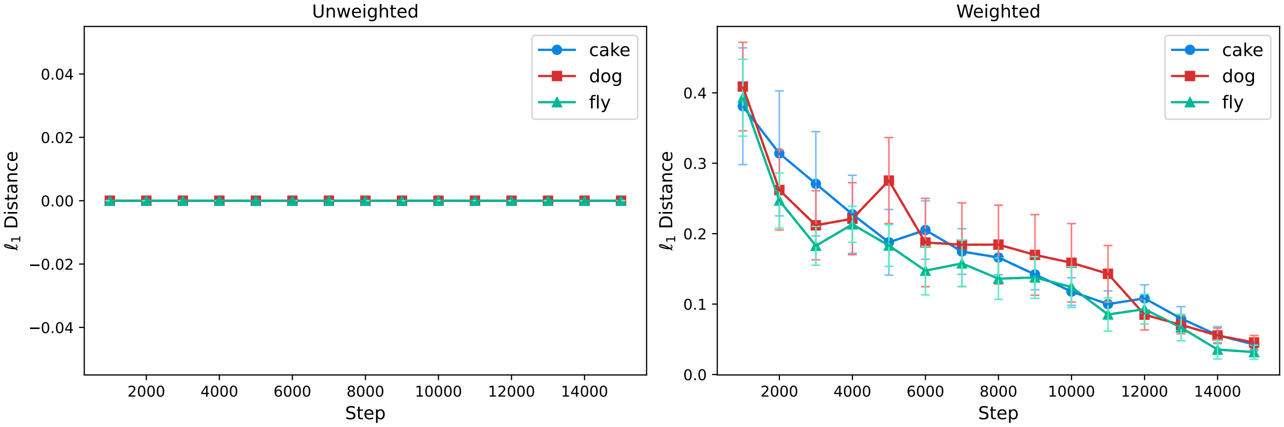


Figure 19. Synthetic relation learning in the subgraphs extracted from ConceptNet.

Table 4. The correspondences between the entities and the letters for WordNet (Part 1).

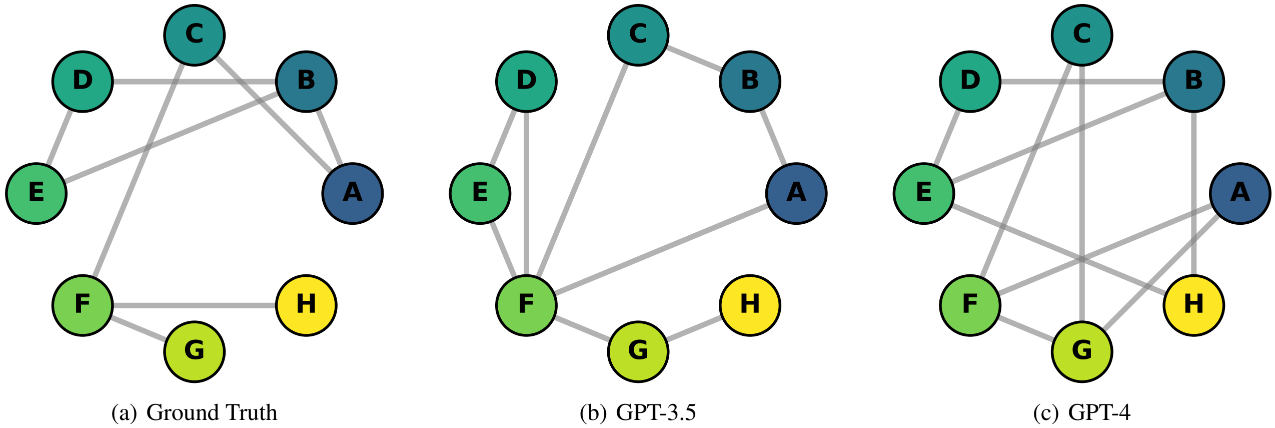
	A	B	C	D	E	F
CAKE	CAKE	BAR	PATTY	BARROOM	SALOON	DISH
DOG	DOG	FRUMP	CAD	BOUNDER	BLACKGUARD	-
FLY	FLY	TENT-FLY	RAINFLY	-	-	-
PAPER	PAPER	NEWSPAPER	COMPOSITION	NEWSPRINT	COMPOSING	CONSTITUTION
ZOO	ZOO	MENAGERIE	FACILITY	COLLECTION	INSTALLATION	ADEPTNESS

Table 5. The correspondences between the entities and the letters for WordNet (Part 2).

	G	H	I	J	K	L
CAKE	DISHFUL	SMASHER	-	-	-	-
DOG	-	-	-	-	-	-
FLY	-	-	-	-	-	-
PAPER	PLACEMENT	ESTABLISHMENT	FORMATION	-	-	-
ZOO	AGGREGATION	ACCUMULATION	INSTALLING	INSTALLMENT	ADROITNESS	DEFTNESS

Table 6. Some relation evaluation results in WordNet. Similarly, the subgraphs are generated from different source entities with $k = 2$ and $d = 3$. The dissimilarity measure is the same as that in ConceptNet.

	CAKE	DOG	FLY	PAPER	ZOO
GPT-3.5	1.33	1.00	0.00	0.75	1.33
GPT-4	1.33	1.00	0.00	1.00	1.00

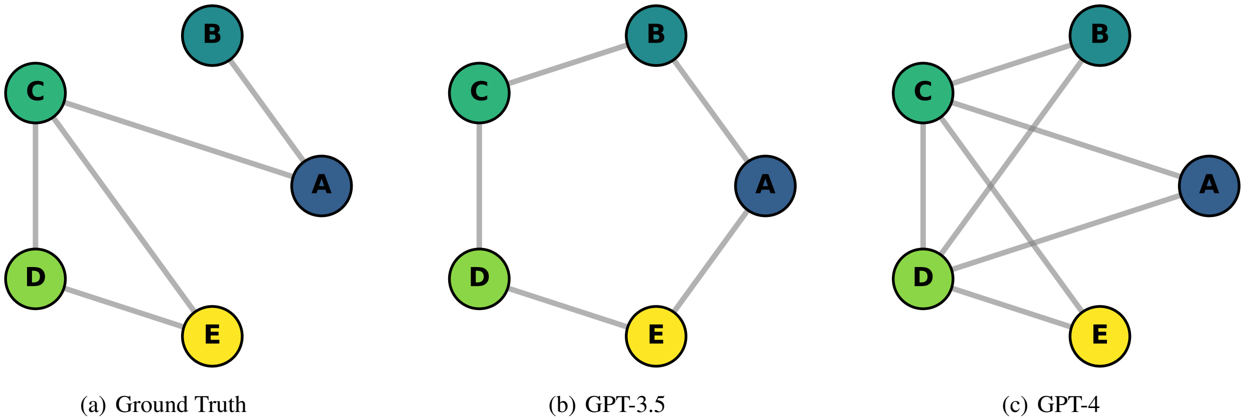


(a) Ground Truth

(b) GPT-3.5

(c) GPT-4

Figure 20. Cake (WordNet).

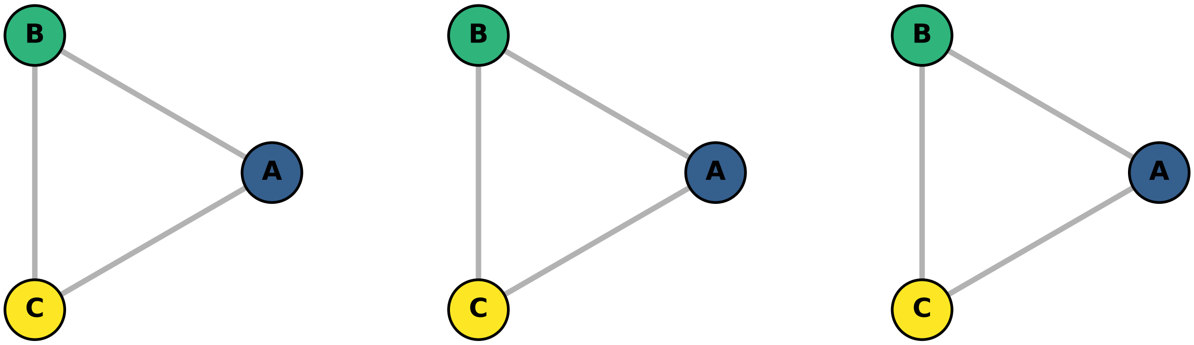


(a) Ground Truth

(b) GPT-3.5

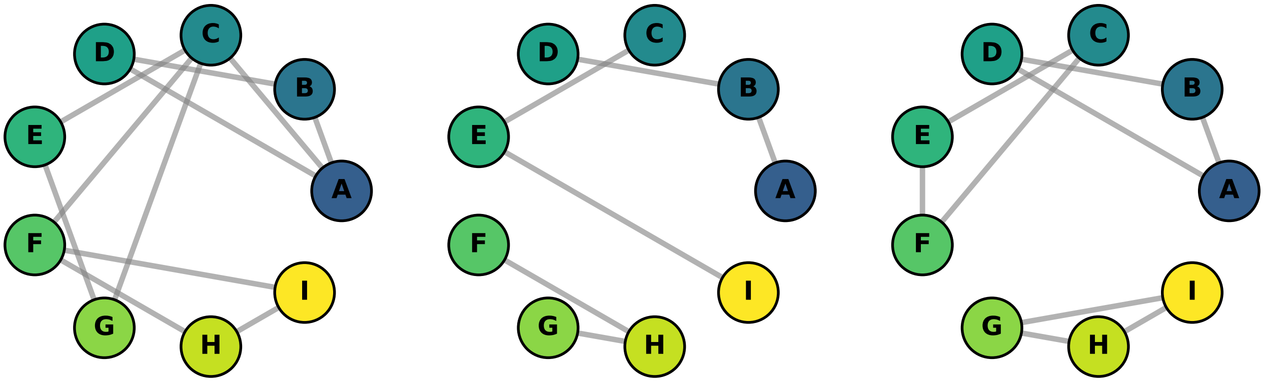
(c) GPT-4

Figure 21. Dog (WordNet).



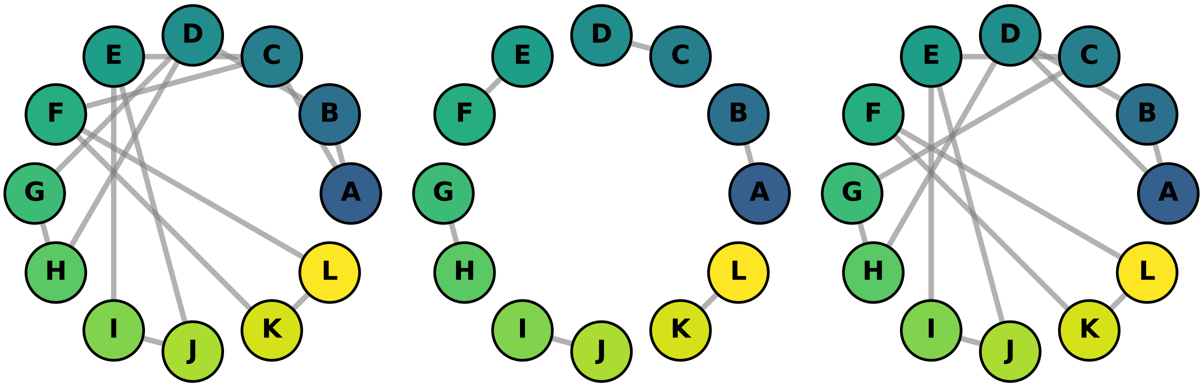
(a) Ground Truth (b) GPT-3.5 (c) GPT-4

Figure 22. Fly (WordNet).



(a) Ground Truth (b) GPT-3.5 (c) GPT-4

Figure 23. Paper (WordNet).



(a) Ground Truth (b) GPT-3.5 (c) GPT-4

Figure 24. Zoo (WordNet).

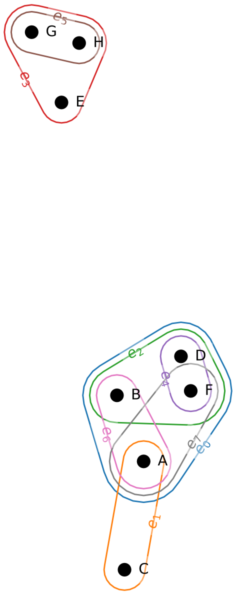


Figure 25. Cake.

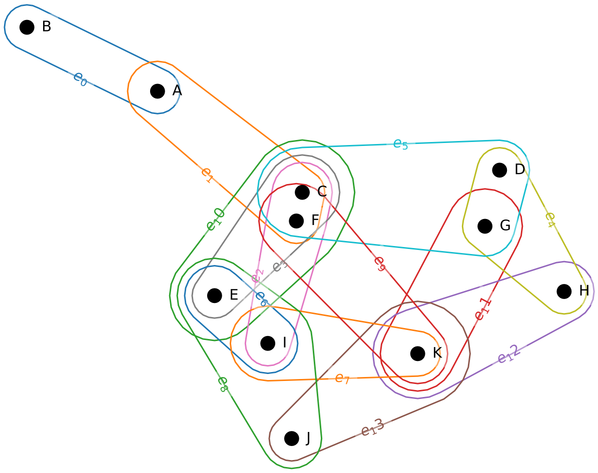


Figure 26. Dog.

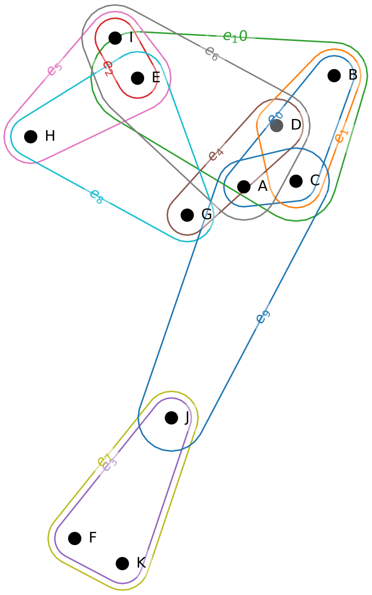


Figure 27. Fly.

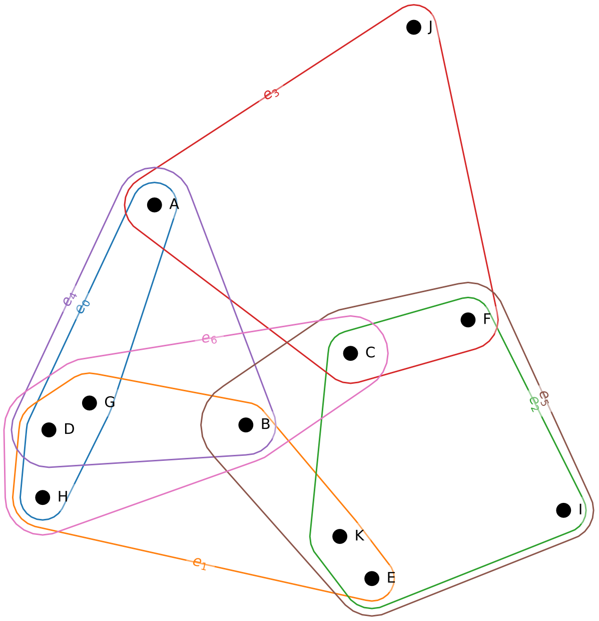


Figure 28. Human.

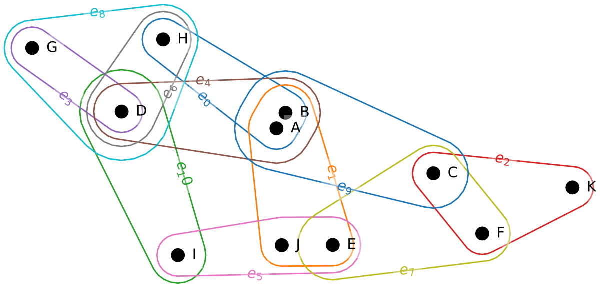


Figure 29. Jacket.

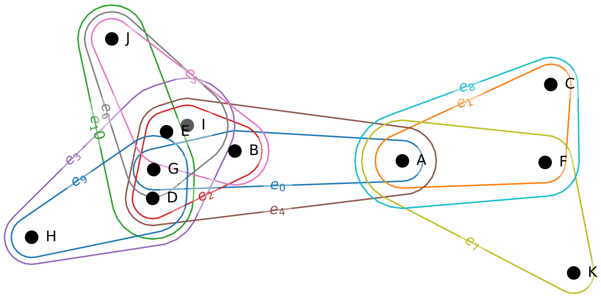


Figure 30. Orange.

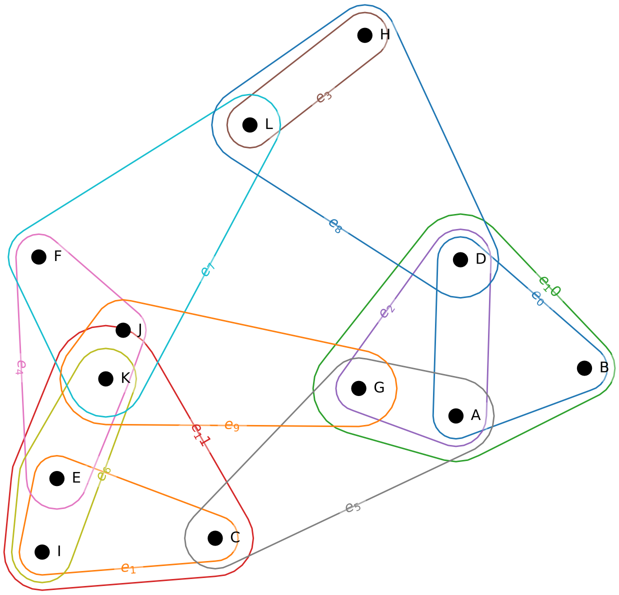


Figure 31. Paper.

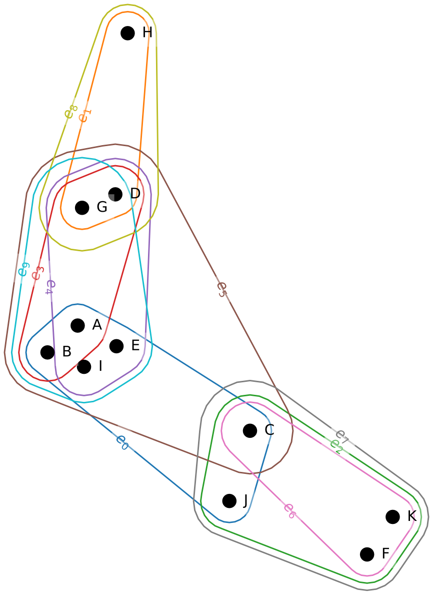


Figure 32. Sea.

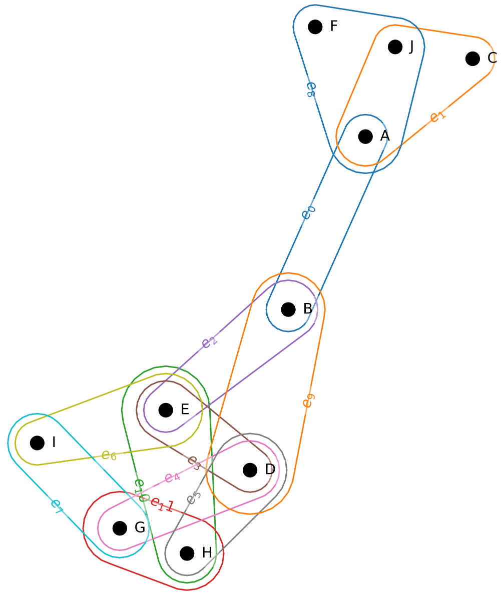


Figure 33. Table.

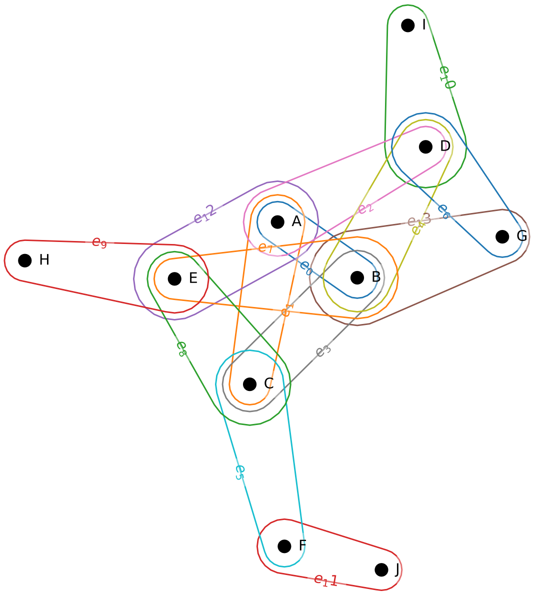


Figure 34. Zoo.