ENHANCING GRAPH GENERATION WITH FIRST-ORDER LOGIC RULES

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing graph generative models produce graphs that are often quite realistic, but sometimes miss domain-specific patterns. Enhancing graph learning with domain knowledge is one of the current frontiers for neural models of graph data. In this paper, we propose a new approach to enhancing deep graph generative models with knowledge that is represented by first-order logic rules. First-order logic provides an expressive formalism for representing interpretable knowledge about relational structures. Our conceptual contribution is a new first-order semantic loss function for training a graph generative model on relational data: maximize the model likelihood subject to a *moment matching constraint*, namely that the expected instance count of each rule matches its observed instance count. Our algorithmic contribution is a novel method for computing the expected instance count of a first-order rule for a standard generative mixture model based on matrix multiplication. Empirical evaluation on seven benchmark datasets, both homogeneous and heterogeneous, shows that moment matching improves the quality of generated graphs substantially (by orders of magnitude on standard graph quality metrics), and improves predictive accuracy on the downstream task of node classification.

1 INTRODUCTION

Generative models for graphs based on graph neural networks (GNNs) have achieved great success in modeling complex graphs (Hamilton, 2020). One of the current research frontiers is enhancing graph learning with domain knowledge (Tian *et al.*, 2024) (Wang *et al.*, 2020) (Sun *et al.*, 2021) (Niresi *et al.*, 2024) (Yu *et al.*, 2023) (Agarwal *et al.*, 2022). Different enhancement methodologies are appropriate for different types of knowledge. In this paper, we consider leveraging knowledge in the form of a *first-order logic knowledge base* (Russell and Norvig, 2010), comprising a set of first-order (FO) formulas. FO formulas represent domain knowledge by specifying important patterns in a domain. Because formulas used in knowledge representation practice often take the form of if-then rules, we refer to our approach as rule-enhanced graph generation. An example rule would be “If person X works in city Y, then X lives in city Y (with probability p)".

Advantages. Logical formulas have several advantages for enhancing graph learning. (1) *Expressiveness*: First-order formulas are one of the most common formalisms for representing domain knowledge in AI and database systems (Russell and Norvig, 2010). (2) *Interpretability*: Logical formulas are easily understood by users and domain experts. (3) *Learnability*: The field of statistical-relational learning (SRL) has developed methods for learning relevant formulas from a heterogeneous training graph, known as *structure learning*. (4) *User Control*: Users can control the behavior of the final graph generation system in a mixed-initiative approach, by specifying and/or rejecting formulas. (5) *Graph Realism and Data Efficiency*: Matching first-order formulas leads to generating more realistic graphs, while requiring less training data.

Approach. Figure 1 shows our system components. We show how fundamental ideas from SRL (Raedt *et al.*, 2016) can be combined with deep graph generative models (GGMs). A fundamental concept of SRL is *moment matching* (Domingos and Lowd, 2019; Russell, 2015; Kuzelka *et al.*, 2018). The general idea is that a formula can be viewed as specifying a *motif* or subgraph pattern with an **instance count** in a given graph. Formula moment matching requires that for each formula,

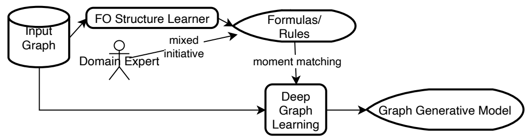

Figure 1: System Overview for Rule-Enhanced Graph Generation

the expected instance count for a model should match the observed instance count in a training graph. Our novel GGM training objective is to *maximize the GGM likelihood subject to moment matching*.

Our algorithmic contribution is a *differentiable new matrix multiplication method* for computing observed and expected instance counts. We show that for every conjunctive formula (satisfying a minor syntactic constraint), there is a corresponding sequence of adjacency matrices, such that i) the observed instance count is obtained by multiplying the data adjacency matrices, and ii) the expected instance count for a standard mixture model with conditionally independent links is obtained by multiplying expected adjacency matrices.

Evaluation. Our methodology uses an A-B design where we compare training a recent state-of-the-art variational graph auto-encoder (Mahmoudzadeh *et al.*, 2024), called VGAE+, with and without moment matching, on seven benchmark datasets. We find that rule-enhanced VGAEs score better than standard VGAEs on several metrics: (1) They generate *more realistic graphs*, by orders of magnitude, as measured by SOTA graph quality metrics (F1 MMD) (Thompson *et al.*, 2022; O'Bray *et al.*, 2022). (2) On the downstream task of *node classification*, the rule-enhanced VGAE node embeddings improve accuracy compared to standard VGAE. (3) Learning curves for node classification show that first-order domain knowledge often leads to more data efficient learning.

Contributions Our main contributions can be summarized as follows.

- A new semantic loss objective function for enhancing generative graph training with domain knowledge represented by first-order formulas: Maximize the data likelihood of a graph generative model, constrained so that the observed number of formula instances matches the expected number of formula instances.
- A new matrix multiplication algorithm for counting the number of formula instances in a graph.
- A proof that the matrix multiplication algorithm can also be used to estimate the expected number of instances for a standard mixture model. It can therefore be leveraged to compute the new semantic loss objective.
- Our new VGAE+R system uses the new objective function to train a VGAE+ model that matches formula instance counts.

2 Related Work

Our work falls under the heading of *neuro-symbolic AI*, a cutting-edge field of AI that aims to combine symbolic formalisms, such as first-order logic, with neural network learning; see Figure 2. For surveys of neuro-symbolic AI, please see (Raedt *et al.*, 2020; Garcez and Lamb, 2023), and Kautz's 2022 Engelmore lecture. Within Kautz's taxonomy, our approach belongs to the *semantic loss* frameworks (type 5) where symbolic knowledge is encoded into the network's loss function (Kautz, 2022; Xu *et al.*, 2018; Marra *et al.*, 2019). The trained system is a standard NN model that does not utilize rules at test time. In contrast, reasoning approaches typically perform symbolic inference (Raedt *et al.*, 2020; Qu *et al.*, 2021) at test time.

Compared to previous semantic loss approaches (Xu *et al.*, 2018), our main innovation is that *we incorporate knowledge expressed in first-order logic, rather than the less powerful formalism of propositional logic*. For example, a propositional rule would be "if a movie is a horror movie, it is not likely to be a romance". A first-order rule could be "if a user rates a horror movie, the user is most likely to be a man". Since first-order rules incorporate relationships, *first-order logic (FOL) can*

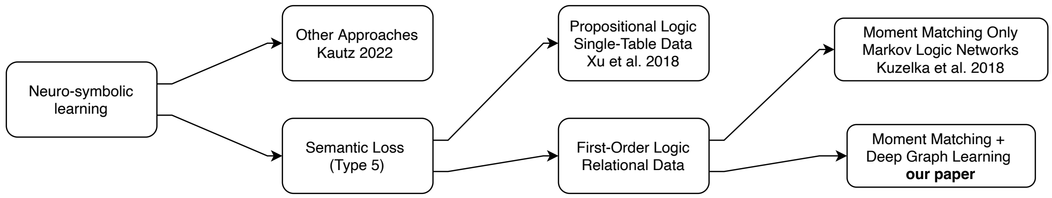

Figure 2: Within neuro-symbolic AI, we develop a new first-order semantic loss approach.

leverage the full power of relational data. We show that our semantic loss for FOL rules reduces to the loss of (Xu *et al.*, 2018) in the propositional case. Compared to previous FOL approaches (e.g., Marra *et al.* (2019)), we use standard FOL semantics (not fuzzy logic), and our computations do not require as input the full Cartesian product grounding over all domain elements (nodes).

Markov Logic Networks and Maximum Entropy Moment Matching. We use the same FO knowledge representation structure as the well-known **Markov Logic Network** (MLN) model, namely a set of FO formulas. The MLN formalism has been applied to represent knowledge in a number of domains, and it has sufficient expressive power to capture other FO formalisms, such as rule-based knowledge (Domingos and Lowd, 2019).

In terms of model training, Kuzelka *et al.* (2018) show that a distribution P over graphs maximizes entropy subject to moment matching if and only if P is defined by an MLN with maximum likelihood weights. Both the maximum entropy objective and our constrained likelihood objective capture the global graph statistics represented by instance counts. However, the GGM likelihood can in addition capture local graph patterns. For example, matching the number of observed triangles in a graph is unlikely to capture community structure, or which nodes have special properties such as centrality.

Deep graph generative models. The closest predecessor to our work is the constrained VGAE model of Ma *et al.* (2018) where a VGAE likelihood is maximized subject to a constraint of the form $g(\boldsymbol{\theta}) = 0$. While this general form covers moment matching, the work of Ma *et al.* does not incorporate FO logic for specifying graph patterns, nor does it address computing pattern counts.

In principle the moment matching likelihood objective can be used for maximum likelihood training with any deep graph generative model. We selected VGAEs as our base model for several reasons. (1) They are a well-established and widely used GGM. Mahmoudzadeh *et al.* (2024) show that their VGAE+ model is a strong multitask model that provides accurate predictions for a wide range of knowledge graph queries, based on inference from a single model. (2) They support learning from a single large graph, rather than from a set of graphs (Faez *et al.*, 2021). Rule learners also utilize the single-graph setting (Qian and Schulte, 2015; Meilicke *et al.*, 2024), so the VGAE input data are compatible with the rule learner input data. (3) As we show in this paper, the conditional link independence assumptions of VGAEs facilitates the computation of expected rule instance counts. Extending rule moment matching to other generative models is a fruitful topic for future research.

3 BACKGROUND ON FIRST-ORDER LOGIC

Attributed Heterogenous Graphs An attributed graph is a pair $\mathcal{G} = (V, E)$ where V is a set of nodes of size $|V| = n$ and $E \subseteq V \times V$ is a set of edges. Node features are summarized in $n \times f$ matrix $\mathbf{X}$ and node labels in a $n \times L$ matrix $\mathbf{L}$ where the u-th row of $\mathbf{L}$ is a one-hot encoding of the label of node u. Different edge types are represented by a set of adjacency matrices $\mathbf{A} = \{\mathbf{A}_1, \ldots, \mathbf{A}_T\}$. The notation $\mathbf{A}_r[u, v] = 1$ indicates that there is a link $u \rightarrow_r v$ of type r from node u to node v. Figure 3 shows part of the information in an attributed graph using the tabular SQL format.

User		
User Id	**Age**	**Gender**
3	0 (0.34)	M (0.55)
5	1 (0.43)	F (0.34)
7	2 (0.90)	M (0.84)
...	...	...

Rating		
User Id	**Movie Id**	**Rating**
3	The Dictator	1 (0.75)
5	Thor	4 (0.36)
5	The Dictator	3 (0.84)
7	BraveHeart	5 (0.98)
...	...	...

Movie			
Movie Id	**Action**	**Drama**	**Horror**
The Dictator	0 (0.38)	0 (0.73)	0 (0.85)
Thor	1 (0.49)	0 (0.66)	0 (0.4)
BraveHeart	1 (0.91)	1 (0.41)	1 (0.7)
...	...	...	...

Figure 3: Excerpt from a relational dataset. (a) An attributed graph represented in table format. (b) The probabilities assigned to each data entry specify a probabilistic graph (see below).

Conjunction ϕ	$n_\phi(\mathcal{G})$
$Age(User) = 0$	376
$Rating(User, Movie) = 1$	4701
$Age(User) = 0, Rating(User, Movie) = 1$	2524

Table 1: Conjunction Instance counts in the MovieLens database $\mathcal{G}$

First-Order Logic We follow previous work in SRL (Schulte and Gholami, 2017; Kimmig *et al.*, 2014). A **population** is a set of individuals of the same type (e.g., a set of $Users$, a set of $Movies$). Individuals are denoted by constants (e.g., $user_3$ and $thor$). An attributed graph specifies a set of individuals (nodes) for each type. A **node variable** ranges over a population, and is denoted in upper case such as $User, Movie, U, V$. A unary **functor** maps an individual to a value, and corresponds to a node attribute/label. A binary functor maps an ordered pair of individuals to a value, and corresponds to an edge/edge type. Functors are denoted f, f' etc.

A **first-order** term (FOT) is of the form $f(\boldsymbol{U})$ where each population variable U_i is of the appropriate type. FOT examples are $age(User)$ and $rating(User, Movie)$. A FOT can be instantiated with individual constants, much like an index in a plate model (Kimmig *et al.*, 2014). A **grounding** $\boldsymbol{U} = \boldsymbol{u}$ for a list of FOTs simultaneously replaces each population variable in the list by a constant. (We assume that different population variables are replaced by different constants.) A ground term, Python-style, assigns individuals as argument to node variables, then applies the functor to return a value. Examples are $age(User = user_5)$, and $rating(User = user_5, Movie = thor)$.

An FO **literal** is of the form $\ell \equiv f(\boldsymbol{U}) = v$. A **conjunction** is a list of literals $\phi = \ell_1, \ldots, \ell_s$. We write $\phi(\boldsymbol{U})$ for an FO conjunction and $\phi(\boldsymbol{U} = \boldsymbol{u})$ for a ground conjunction. A graph $\mathcal{G}$ **satisfies** a ground literal if the graph assigns value v to the ground term $f(\boldsymbol{U} = \boldsymbol{u})$, and satisfies the conjunction ϕ if it satisfies each ground literal in the conjunction. The **instance count** $n_\phi(\mathcal{G})$ in a graph $\mathcal{G}$ returns the number of ϕ-groundings satisfied by graph $\mathcal{G}$.

A **probabilistic graph** $\tilde{\mathcal{G}}$ assigns a probability $p_{\tilde{\mathcal{G}}}(\ell(\boldsymbol{U} = \boldsymbol{u}))$ to each ground literal. The **probabilistic instance count** of a conjunction (Kuzelka, 2023) is the probability product, summed over all conjunction groundings:

$$n_\phi(\tilde{\mathcal{G}}) = \sum_{\boldsymbol{U}=\boldsymbol{u}} \prod_{i=1}^{s} p_{\tilde{\mathcal{G}}}(\ell_i(\boldsymbol{U} = \boldsymbol{u})) \text{ for } \phi = \ell_1, \ldots, \ell_s \tag{1}$$

Examples. $Age(User) = 1$,$rating(User, Movie) = 4$ is an FO conjunction. Its grounding $age(User = user_5) = 1, rating(User = user_5, Movie = thor) = 4$ is satisfied by the data of Figure 3(a). In the probabilistic graph Figure 3(b), the probability of this conjunction is $0.43 \times 0.36 = 0.1548$.

Table 1 illustrates FO instance counts using the MovieLens dataset (Qian and Schulte, 2015). Movielens contains 376 users at age level 0. The number of user-movie pairs with a rating of 1 is 4701. The number of such pairs with the user at age level 0 is 2524. *An FO conjunction specifies a graph motif*, and the instance count is the motif count (Ma *et al.*, 2019) (see Figure 9 for illustration).

4 RULE-ENHANCED GRAPH GENERATION

This section considers how to enhance training a parametrized graph generative model (GGM) $P_{\boldsymbol{\theta}}$ on a training graph D, with a list of formulas $\phi_1, \ldots, \phi_k$. Our *semantic loss objective* maximizes the data likelihood $P_{\boldsymbol{\theta}}(D)$, subject to the FO moment matching constraint that $E_{\boldsymbol{\theta}}[n_i] = n_i(D)$, where $n_i(D)$ is the **data instance count** of formula ϕ_i, and $E_{\boldsymbol{\theta}}[n_i] \equiv \sum_{\mathcal{G}} P_{\boldsymbol{\theta}}(\mathcal{G}) n_i(\mathcal{G})$ is the **expected instance count** for the GGM. For a mixture model GGM, we derive the following Lagrangian ELBO.

Proposition 1. *Suppose that $P_{\boldsymbol{\theta}}(\mathcal{G}) = \int p(\mathcal{G}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$ is a mixture model. Then*

$$\ln P_{\boldsymbol{\theta}}(D) - \lambda/k \sum_{i=1}^{k} \rho(n_i(D), E_{\boldsymbol{\theta}}[n_i]) \geq \tag{2}$$

$$E_{\boldsymbol{z}\sim q_{\phi}(\boldsymbol{z}|D)}[\ln P_{\boldsymbol{\theta}}(D|\boldsymbol{z}) - KL\big(q_{\phi}(\boldsymbol{z}|D)||p(\boldsymbol{z})\big) \tag{3}$$

$$-\lambda/k \sum_{i=1}^{k} \rho(n_i(D), E_{\boldsymbol{\theta}}[n_i|\boldsymbol{z}])], \tag{4}$$

where $\rho(count_1, count_2) \geq 0$ is a differentiable count distance metric convex in $E_{\boldsymbol{\theta}}[n_i|\boldsymbol{z}])]$.

Proposition 1 says that the constrained likelihood Equation (2) can be approximated by our new **moment matching variational ELBO objective** (3). To compare an expected count to an observed count, our experiments use

$$\rho(n_i(D), E_{\boldsymbol{\theta}}[n_i|\boldsymbol{z}]) = |\ln n_i(D) - \ln E_{\boldsymbol{\theta}}[n_i|\boldsymbol{z}]|.$$

Conjunction counts grow exponentially with the number of node variables in the conjunction. Comparing expected counts on a log-scale decreases the impact of the number of node variables and improves numeric stability. With this choice of ρ, *the FO semantic loss Equation* (2) *reduces to the semantic loss of (Xu* et al.*, 2018) for a propositional formula ϕ*; see Appendix A.6 for details.

4.1 Implementing the Moment Matching ELBO

Our novel VGAE+R architecture extends the recent VGAE+ architecture (Mahmoudzadeh *et al.*, 2024) to match rules, including the new **motif loss** (4).

Encoder-Decoder Architecture. Figure 10 shows the VGAE+R architecture. The **encoder** model $q_{\phi}(\boldsymbol{z}|D)$ can be any GNN that maps a heterogeneous graph to node embeddings, such as RGCN. The VGAE+R **decoder** independently maps node embeddings to different graph components with three different decoders (Mahmoudzadeh *et al.*, 2024):

$$\ln P_{\boldsymbol{\theta}}(D|\boldsymbol{z}) = [\alpha \ln p_{\eta}(\mathbf{A}|\boldsymbol{z}) + \beta \ln p_{\psi}(\boldsymbol{X}|\boldsymbol{z}) + \gamma \ln p_{\phi}(\boldsymbol{L}|\boldsymbol{z})$$

where $p_{\eta} : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$ is a trainable **link decoder**, p_{ψ} is a trainable **feature decoder**, and p_{ϕ} is a trainable **label decoder** (see Figure 10). The hyperparameters α, β and γ weight the importance of different reconstruction tasks.

Computing Expected Instance Counts. Given a set of node embeddings $\boldsymbol{z}$, the **expected graph** $\tilde{\mathcal{G}}_{\boldsymbol{z}}$ is a probabilistic graph that assigns a probability to each ground literal by applying the decoder to the relevant links/node features/edge types. For examples see Figure 3 and Figure 11.

Proposition 2. *The expected instance count given a set of node embeddings can be computed as the instance count in the expected graph: $E_{\boldsymbol{\theta}}[n_i|\boldsymbol{z}] = n_i(\tilde{\mathcal{G}}_{\boldsymbol{z}})$.*

The proof is in the supplement. The upshot is that *FO moment moment matching can be implemented by performing (probabilistic) instance counting in a single graph.*

5 Matrix Multiplication for Instance Counting

SOTA MLN structure learners output a set of conjunctive formulas or if-then rules (Qian and Schulte, 2015; Khot *et al.*, 2011; Cui *et al.*, 2022; Potter *et al.*, 2024). We discuss instance counting for conjunctive formulas, which we can be extended to if-then rules by restricting counts to instances that match the antecedent (body); see Appendix A.4 for more details.

This section presents a novel matrix multiplication method for instance counting with conjunctive formulas, that is differentiable and applies to both discrete and probabilistic graphs. To illustrate the basic idea, consider the conjunction $R(U_1, V_1), R(V_1, V_2), R(V_2, U_1)$, whose instance count gives the number of triangles in an undirected graph represented by an adjacency matrix $\mathbf{A}$. It is well-known that the triangle count is given by $\sum_{u=1}^{n} \mathbf{A}^3_{u,u}$, the trace of the third power of the adjacency matrix. We generalize this approach to a large class of logical formulas. A **chain conjunction** of binary literals is of the form $\phi = \ell_1(U_1, V_1), \ldots, \ell_P(U_P, V_P)$ where $V_i = U_{i+1}$ for every i. Algorithm 1 maps each chain conjunction to a sequence of adjacency matrix multiplications, such that the conjunction's instance count can be found by executing the matrix multiplications.

Algorithm 1 Matrix Multiplication for Instance Counting

```
Input: Chain conjunction φ = {ℓ_1(U_1, V_1), ..., ℓ_P(U_P, V_P)}
Output: Instance count n_φ(G) or expected count n_φ(G̃_z)
{Initialize adjacency matrices A_{ℓ_k} for binary literals ℓ_k, k = 1, ..., P }
for k = 1 to P do
    if positive literal ℓ_k = R(U_k, V_k) = 1 then
        A_{ℓ_k} ← A_r
    else if ℓ_k = R(U_k, V_k) = 0 then
        A_{ℓ_k} ← ¬A_r where ¬A_r is the complement of A_r
    end if
end for
O_1 ← A_{ℓ_1}
for k = 1 to P − 1 do
    O_{k+1} ← O_k · A_{ℓ_{k+1}}
    if V_{k+1} = U_1 then
        Zero out the non-diagonal entries of O_{k+1}
    end if
end for
Return:
n_φ(G) = Σ(O_P(φ)) for input graph G
n_φ(G̃_z) = Σ(O_P(φ)) for expected graph G̃_z
```

Example. Consider the chain conjunction $\mathit{AdvisedBy}(\mathit{Student}, \mathit{Professor}),\ \mathit{Teaches}(\mathit{Professor}, \mathit{Course})\ \mathit{TakesCourse}(\mathit{Course}, \mathit{Student})$. Figure 4 shows the corresponding sequence of matrix multiplications in a sample graph.

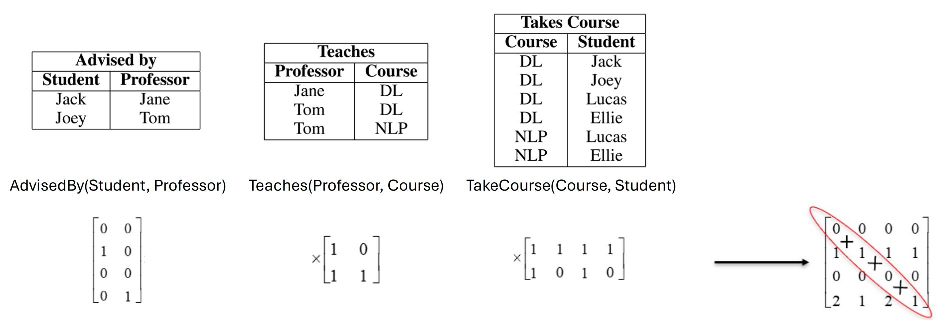

Figure 4: The matrix multiplication sequence for our example conjunction and sample graph data. The final result is 2, which is the number of satisfying groundings in the input graph.

Extensions. Unary literals can be included by omitting nodes from the input graph that do not satisfy them. Probabilistic instance counts can be obtained by using soft matrices $\tilde{A}, \tilde{X}, \tilde{L}$; see Appendix A.9.

Correctness. The next proposition shows that the instance count for the chain conjunction can be obtained through summing over the entries in the constructed matrix product.

Proposition 3. *Let ϕ be a centered chain conjunction of length k, i.e., the first node variable is the only one that appears twice non-consecutively.*

1. *For an input graph $\mathcal{G}$, the (u, v)-th entry of O_k counts the number of groundings of ϕ in $\mathcal{G}$ where $U_1 = u$ and $V_P = v$. Therefore $n_\phi(\mathcal{G}) = \sum(O_k(\phi))$.*

2. *For an expected graph $\tilde{\mathcal{G}}_{\boldsymbol{z}}$, the (u, v)-th entry of O_k counts the expected number of groundings of ϕ where $U_1 = u$ and $V_P = v$. Therefore $n_\phi(\tilde{\mathcal{G}}_{\boldsymbol{z}}) = \sum(O_k(\phi))$.*

In our experiments, we found that all learned rules were centered. The Appendix extends the matrix multiplication method to non-centered chains.

Computational Complexity Algorithm 1 translates a logical formula into a sequence of matrix multiplications in time linear in the length of the formula. The number of binary literals is small enough to be treated as a constant $k \leq 5$. The bottleneck is scaling a k-fold adjacency matrix product to large graphs, especially the dense expected adjacency matrices.

6 Evaluation

We detail our methodology and discuss our empirical results. Appendix A.5 provides training details.

6.1 Experimental Design

We describe our benchmark datasets, comparison methods, and how evaluation metrics are computed.

Datasets We use datasets from previous studies of GGMs (Mahmoudzadeh *et al.*, 2024; Yun *et al.*, 2019; Hao *et al.*, 2020). Cora, ACM, and CiteSeer are citation networks, IMDb is a movie dataset, and UW represents an academic department. Appendix A.1 presents dataset and preprocessing details. We report results for homogeneous versions of ACM and IMDb in the main paper, and for heterogeneous versions in Appendix A.2.

Evaluation Metrics We compare rule-enhanced VGAE+R training with plain VGAE+ training, using three main metrics. In the following, we refer to a complete dataset as the **input graph**. Our evaluation measures *graph realism*—the quality of generated graphs—and the downstream task of node classification.

Count Distance. Given the training graph D, we sample one node embedding matrix $\boldsymbol{z}$ from the encoder posterior $q_\phi(\boldsymbol{z}|D)$ and then apply the decoder model eq. (5) to $\boldsymbol{z}$ to obtain the expected graph $\tilde{D}_{\boldsymbol{z}}$. We report the mean squared distance $(1/k \sum_{i=1}^{k}[n_i(D) - n_i(\tilde{D}_{\boldsymbol{z}})]^2)^{1/2}$—between the observed motif counts and the expected motif counts in the reconstructed graph—as the **count distance** (CD), where k is the number of formulas.

Graph Realism measures how similar graphs generated by the model are to observed graphs. How to quantitatively assess generated graphs has been studied in recent papers. We adapt the SOTA approach that compares graph embeddings of the training graph to embeddings of generated graphs using Maximum Mean Distance (MMD) (O'Bray *et al.*, 2022; Thompson *et al.*, 2022; Shirzad *et al.*, 2022); see Appendix A.10 for details. The MMD metric is independent of the training objective.

Node Classification To compute a node classification score, we randomly divide the nodes in the input graph into training, test and validation nodes (70%/20%/10%). The training graph is the input graph but with the test node labels removed. At test time, we run the encoder on the input graph to obtain node embeddings for all nodes, then apply the decoder to predict node labels for the test nodes.

6.2 Experimental Results

Count Distance and Graph Realism are the most important metrics for us since they directly pertain to graph generation quality. Our graph generation baseline is the VGAE+ model trained without moment matching (i.e., $\lambda = 0$). To obtain formulas, we used the SOTA MLN structure learning system Factorbase (Qian and Schulte, 2015) with default settings (Appendix A.4). To illustrate, in the UW dataset, the learned formulas capture several patterns that express university domain knowledge, such as the following. (1) Whether a person teaches a course correlates with whether they have a position. (2) Course teachers are more likely to be professors. (3) A person's program phase predicts their years in the program.

Table 2: Mean $\pm$ Std for Count Distance (CD$\downarrow$) and Graph Realism (MMD$\downarrow$) with Improvements (in scientific notation, so "e" represents $\times 10^n$).

Dataset	Count Distance (MSE)			Graph Realism (MMD)		
	VGAE+	VGAE+R	Improv. (%)	VGAE+	VGAE+R	Improv. (%)
Cora	$5.14e4 \pm 8.07e3$	$2.68e4 \pm 9.79e3$	47.81	$4.03e18 \pm 3.21e18$	$6.84e17 \pm 9.15e17$	83.03
CiteSeer	$3.40e4 \pm 1.06e3$	$2.47e4 \pm 1.38e3$	27.35	$1.20e18 \pm 3.15e17$	$6.14e16 \pm 3.65e16$	94.88
Computers	$4.63e5 \pm 2.67e4$	$3.54e5 \pm 6.44e4$	23.61	$3.80e25 \pm 1.86e25$	$1.08e24 \pm 1.45e24$	97.16
Photo	$2.01e5 \pm 1.96e4$	$1.24e5 \pm 1.37e4$	38.01	$1.16e24 \pm 8.30e23$	$1.66e22 \pm 7.63e21$	98.57
IMDb	$5.86e5 \pm 1.96e4$	$3.23e5 \pm 1.34e5$	44.81	$2.94e23 \pm 6.29e22$	$4.58e22 \pm 3.25e22$	84.42
UW	$9.92e5 \pm 8.87e4$	$9.48e5 \pm 5.69e4$	4.51	$4.72e13 \pm 1.90e13$	$2.32e13 \pm 1.94e13$	50.74
ACM	$1.24e5 \pm 4.86e3$	$2.81e4 \pm 3.54e3$	77.34	$3.31e20 \pm 1.32e20$	$3.29e18 \pm 3.08e18$	99.01

6.2.1 COUNT DISTANCE AND GRAPH REALISM

Table 4 shows the difference between expected and observe instance counts. Both methods show large absolute distances because a VGAE model tends to produce overly dense graphs (Orbanz and Roy, 2014). However we observe a *very large improvement in the match between expected and observed counts*, at least 23% on all datasets, except for the small graph UW with an improvement of 4.51%. On the graph realism metric, Table 4 again shows large absolute distances with the training set, and *very large improvements through FO moment matching, by an order of magnitude*. Overall we conclude that unconstrained VGAE training does not match the instantiation counts of the learned formulas and that enforcing moment matching has a large impact on generated graph realism. In addition, Section A.13 shows that VGAE + R outperforms VGAE + in statistic-based MMD metrics. Also, as discussed in Appendix A.14, we report results for Count Distance Evaluation based on prior embedding sampling. Moreover, there is a report on robustness to noisy or incomplete rules for Cora dataset in Appendix A.16.

6.2.2 NODE CLASSIFICATION

Since SOTA performance on node classification is nearly saturated, we do not claim that VGAE+R leads to uniformly best node classification. Instead we investigate two hypotheses:

1. Rule enhancement can improve GGM-based classification when the rules capture relevant domain knowledge.
2. The VGAE+R model is competitive with current baselines.

Table 3 *shows an improvement from rule enhancement (bold)* on 4 out 7 datasets, substantive for two of them (Cora and UW). The biggest improvement is on Cora, where moment matching increase the AUC score by 10%. Even when the rules are not very relevant for the class label, moment matching decreases classification performance only slightly.

Table 6 compares the rule-enhanced VGAE+R with the recent node classification baselines, described in Appendix A.3. Our VGAE+R model shows the best node AUC classification performance on 3/6 datasets (4/6 on F1). The biggest improvement is on CiteSeer where our baselines are far from SOTA performance. GiGaMAE is a strong baseline that achieves the best result on two datasets (Table 3). Our conclusion is that *rule-enhanced graph generation supports node classification that is competitive with recent baselines*.

Learning Curve We report a learning curve experiment to examine the effect of rule knowledge on data efficiency. The idea is to simulate the impact of a domain expert providing the model with a strong set of rules. We report the predictive accuracy on the test labels, after training the VGAE with and without rules on $x = 25\%, 50\%, 75\%, 100\%$ of training labels.

Figures 13 to 15 show that *moment matching improves data efficiency substantially on the CiteSeer, Cora, and Photos datasets*. For example on CiteSeer with 50% of node labels, moment matching achieves a 15% higher F1-score than baseline VGAE learning. The learning curves with and without

Dataset	Metric	VGAE+R	VGAE+	GiGaMAE
Cora	AUC	**0.965 ± 0.013**	0.865 ± 0.043	0.920
	F1 Score	**0.887 ± 0.016**	0.699 ± 0.103	0.856
UW	AUC	**0.960 ± 0.012**	0.889 ± 0.054	-
	F1 Score	**0.654 ± 0.031**	0.618 ± 0.030	-
CiteSeer	AUC	**0.903 ± 0.008**	0.891 ± 0.013	0.842
	F1 Score	**0.794 ± 0.042**	0.733 ± 0.058	<u>0.798</u>
Computers	AUC	0.915 ± 0.022	**0.920 ± 0.004**	<u>0.941</u>
	F1 Score	0.827 ± 0.047	**0.837 ± 0.005**	0.770
Photo	AUC	**0.991 ± 0.003**	0.980 ± 0.021	0.963
	F1 Score	**0.972 ± 0.002**	0.946 ± 0.052	0.569
ACM	AUC	0.761 ± 0.077	**0.775 ± 0.074**	<u>0.823</u>
	F1 Score	**0.525 ± 0.014**	0.523 ± 0.009	0.440
IMDb	AUC	0.829 ± 0.006	**0.828 ± 0.011**	<u>0.890</u>
	F1 Score	**0.697 ± 0.008**	0.687 ± 0.014	0.457

Table 3: Node classification results for graph generation with and without rule enhancement. The recent GiGAMAE system is a strong baseline. Bold indicates the best VGAE score, underline the best GiGAMAE score. Standard deviations are reported for five random weight initializations.

moment matching are similar for the datasets ACM, IMDb and Computers because their rules affect node classification little.

Impact of Rules on Training Figure 19 shows the node label loss component of decoder training Equation (2) for the CiteSeer dataset. Rule matching adds a difficult new component to the VGAE+ objective, which initially causes a spike in the label loss component. After the VGAE+R model has encoded the background knowledge in its weights, it learns to optimize the other components, including the node label loss. This shows that *rule matching is a strong regularizer* that takes the network to a very different part of weight space compared to the baseline VGAE+ loss, and supports better generalization. The supplement Appendix A.12 illustrates this pattern in loss curves.

7 Conclusion, Limitations and Future Work

We proposed a new semantic loss objective function for training a deep graph generative model (GGM) to incorporate FO domain knowledge expressed by logical formulas: Maximize the data likelihood subject to a moment matching constraint, which requires *the expected formula instance counts under a model to match the observed instance count.* Our main algorithmic contribution is a *new differentiable matrix multiplication method for computing both observed and expected counts.* In empirical evaluation, we found that moment matching improves the quality of the graphs generated by a Variational Graph Auto-Encoder (VGAE) model by an order of magnitude or more, both with respect to instance counts and with respect to a standard metric of graph realism. Applying the trained GGM to the downstream task of node classification, moment matching improved classification accuracy on all but one of our benchmark datasets. The domain knowledge incorporated in the model is often effective in improving predictions from small datasets, as shown in learning curves.

Limitations. As our paper is the first to combine deep graph generation with a first-order semantic loss, it leaves several aspects open for future developments. (1) Scaling the matrix multiplication algorithm for expected counts is a challenge (Section 5). There is a report on scalability and runtime analysis in Appendix A.15. A possible solution are approximation algorithms from the related problem of weighted model counting (van Bremen and Kuzelka, 2020). (2) An incorrect or incomplete set of rules limits the effectiveness of the semantic loss function. We did not explore methods for validating the knowledge expressed in formulas, such as human-in-the-loop. (3) Because rule learners (MLN structure learners) assume a single dataset, we did not explore enhancing GGMs other than VGAEs (e.g., auto-regressive, diffusion, and matching flow models (cf. Section 2).

In sum, moment matching presents a novel semantic loss approach to neuro-symbolic AI that combines logical rules with deep graph learning. Our experiments show great potential for enhancing deep graph generative models with rule-based knowledge.

REFERENCES

Ankush Agarwal, Raj Gite, Shreya Laddha, Pushpak Bhattacharyya, Satyanarayan Kar, Asif Ekbal, Prabhjit Thind, Rajesh Zele, and Ravi Shankar. Knowledge graph - deep learning: A case study in question answering in aviation safety domain. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6260–6270, June 2022.

Shan Cui, Tao Zhu, Xiao Zhang, and Huansheng Ning. Mcla: Research on cumulative learning of markov logic network. *Knowledge-Based Systems*, 242:108352, 2022.

Pedro Domingos and Daniel Lowd. Unifying logical and statistical ai with markov logic. *Communications of the ACM*, 62(7):74–83, 2019.

Pedro Domingos and Matthew Richardson. Markov logic: A unifying framework for statistical relational learning. In *Introduction to Statistical Relational Learning*. MIT Press, 2007.

Faezeh Faez, Yassaman Ommi, Mahdieh Soleymani Baghshah, and Hamid R Rabiee. Deep graph generators: A survey. *IEEE Access*, 9:106675–106702, 2021.

Artur d'Avila Garcez and Luis C Lamb. Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review*, pages 1–20, 2023.

William L Hamilton. *Graph representation learning*, volume 14. Morgan & Claypool Publishers, 2020.

Yu Hao, Xin Cao, Yixiang Fang, Xike Xie, and Sibo Wang. Inductive link prediction for nodes having only attribute information. *arXiv preprint arXiv:2007.08053*, 2020.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*, pages 4116–4126. PMLR, 2020.

Manfred Jaeger and Oliver Schulte. A complete characterization of projectivity for statistical relational models. In *International Joint Conferences on Artificial Intelligence IJCAI-20*, pages 4283–4290, 2020.

Henry Kautz. The third AI summer: AAAI Robert S. Engelmore Memorial Lecture. *AI magazine*, 43(1):105–125, 2022.

Seyed Mehran Kazemi, David Buchman, Kristian Kersting, Sriraam Natarajan, and David Poole. Relational logistic regression. In *International Conference on the Principles of Knowledge Representation and Reasoning, KRR 2014*, 2014.

Hassan Khosravi, Oliver Schulte, Jianfeng Hu, and Tianxing Gao. Learning compact Markov logic networks with decision trees. *Machine Learning*, 89(3):257–277, 2012.

Tushar Khot, Sriraam Natarajan, Kristian Kersting, and Jude W. Shavlik. Learning Markov logic networks via functional gradient boosting. In *ICDM*, pages 320–329. IEEE Computer Society, 2011.

Angelika Kimmig, Lilyana Mihalkova, and Lise Getoor. Lifted graphical models: a survey. *Machine Learning*, 99(1):1–45, 2014.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Ondrej Kuzelka, Yuyi Wang, Jesse Davis, and Steven Schockaert. Relational marginal problems: Theory and estimation. In *AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 1–8, 2018.

Ondrej Kuzelka. Counting and sampling models in first-order logic. In *International Joint Conference on Artificial Intelligence, IJCAI*, pages 7020–7025, 2023.

Tengfei Ma, Jie Chen, and Cao Xiao. Constrained generation of semantically valid graphs via regularizing variational autoencoders. In *Advances in Neural Information Processing Systems, NeurIPS 2018*, pages 7113–7124, 2018.

Chenhao Ma, Reynold Cheng, Laks VS Lakshmanan, Tobias Grubenmann, Yixiang Fang, and Xiaodong Li. Linc: a motif counting algorithm for uncertain graphs. *VLDB Endowment*, 13(2):155–168, 2019.

Erfaneh Mahmoudzadeh, Parmis Naddaf, Kiarash Zahirnia, and Oliver Schulte. Deep generative models for subgraph prediction. In *European Conference on Artificial Intelligence, ECAI 2024*, volume 392, pages 3128–3136, 2024.

Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, and Marco Gori. Lyrics: A general interface layer to integrate logic inference and deep learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD*, pages 283–298, 2019.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. *ACM*, 2015.

Christian Meilicke, Melisachew Wudage Chekol, Patrick Betz, Manuel Fink, and Heiner Stuckeschmidt. Anytime bottom-up rule learning for large-scale knowledge graph completion. *The VLDB Journal*, 33(1):131–161, 2024.

Keivan Faghih Niresi, Lucas Kuhn, Gaëtan Frusque, and Olga Fink. Informed graph learning by domain knowledge injection and smooth graph signal representation. In *2024 32nd European Signal Processing Conference (EUSIPCO)*, pages 2467–2471, 2024.

Leslie O'Bray, Max Horn, Bastian Rieck, and Karsten Borgwardt. Evaluation metrics for graph generative models: Problems, pitfalls, and practical solutions. In *International Conference on Learning Representations, ICLR*, 2022.

Peter Orbanz and Daniel M Roy. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):437–461, 2014.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge university press, 2000.

George BG Potter, Gertjan Burghouts, and Joris Sijs. Incremental learning of affordances using markov logic networks. In *2024 Eighth IEEE International Conference on Robotic Computing (IRC)*, pages 46–53. IEEE, 2024.

Zhensong Qian and Oliver Schulte. Factorbase: Multi-relational model learning with sql all the way. In *Data Science and Advanced Analytics (DSAA)*, pages 1–10, 2015.

Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. RNNLogic: Learning logic rules for reasoning on knowledge graphs. In *International Conference on Learning Representations, ICLR*, 2021.

Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis lectures on artificial intelligence and machine learning*, 10(2):1–189, 2016.

Luc de Raedt, Sebastijan Dumančić, Robin Manhaeve, and Giuseppe Marra. From statistical relational to neuro-symbolic artificial intelligence. In *International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4943–4950, 2020.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010.

Stuart Russell. Unifying logic and probability. *Communications of the ACM*, 58(7):88–97, 2015.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *International conference on the semantic web (ESWC 2018)*, pages 593–607, 2018.

Oliver Schulte and Sajjad Gholami. Locally consistent Bayesian network scores for multi-relational data. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2693–2700, 2017.

Oliver Schulte and Zhensong Qian. Factorbase: Sql for learning a multi-relational graphical model, August 2015.

Oliver Schulte and Zhensong Qian. FACTORBASE: multi-relational structure learning with SQL all the way. *International Journal of Data Science and Analytics*, 7(4):1–21, 2018.

Yucheng Shi, Yushun Dong, Qiaoyu Tan, Jundong Li, and Ninghao Liu. Gigamae: Generalizable graph masked autoencoder via collaborative latent space reconstruction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2259–2269, 2023.

Hamed Shirzad, Kaveh Hassani, and Danica J Sutherland. Evaluating graph generative models with contrastively learned features. *Advances in Neural Information Processing Systems, NeurIPS*, 2022.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning–ICANN 2018*, pages 412–422. Springer, 2018.

Mengying Sun, Jing Xing, Huijun Wang, Bin Chen, and Jiayu Zhou. Mocl: Data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 3585–3594, 2021.

Rylee Thompson, Boris Knyazev, Elahe Ghalebi, Jungtaek Kim, and Graham W Taylor. On evaluation metrics for graph generative models. In *International Conference on Learning Representations, ICLR*, 2022.

Yijun Tian, Shichao Pei, Xiangliang Zhang, Wei Wang, Hanghang Tong, and Nitesh V. Chawla, editors. *Knowledge-enhanced Graph Learning*. Workshop at AAAI, 2024.

Timothy van Bremen and Ondrej Kuzelka. Approximate weighted first-order model counting: Exploiting fast approximate model counters and symmetry. In *International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4252–4258, 7 2020.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. 2023.

Daisy Zhe Wang, Eirinaios Michelakis, Minos Garofalakis, and Joseph M Hellerstein. Bayesstore: managing large, uncertain data repositories with probabilistic graphical models. In *VLDB*, pages 340–351, 2008.

Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. Pm2.5-gnn: A domain knowledge enhanced graph neural network for pm2.5 forecasting. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '20, page 163–166, 2020.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. In *International Conference on Machine Learning, ICML*, volume 80, pages 5502–5511, 2018.

Huafei Yu, Tinghua Ai, Min Yang, Jingzhong Li, Lu Wang, Aji Gao, Tianyuan Xiao, and Zhe Zhou. Integrating domain knowledge and graph convolutional neural networks to support river network selection. *Transactions in GIS*, 27(7):1898–1927, 2023.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Neural Information Processing Systems, NeurIPS*, 32, 2019.

Qin Zhang, Zelin Shi, Xiaolin Zhang, Xiaojun Chen, Philippe Fournier-Viger, and Shirui Pan. G2pxy: generative open-set node classification on graphs with proxy unknowns. In *International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 4576–4583, 2023.