Exploratory Analysis of Graph Data by Leveraging Domain Knowledge

Di Jin
University of Michigan, Ann Arbor
dijin@umich.edu

Danai Koutra
University of Michigan, Ann Arbor
dkoutra@umich.edu

***Abstract*—Given the soaring amount of data being generated daily, graph mining tasks are becoming increasingly challenging, leading to tremendous demand for summarization techniques. Feature selection is a representative approach that simplifies a dataset by choosing features that are relevant to a specific task, such as classification, prediction, and anomaly detection. Although it can be viewed as a way to summarize a graph in terms of a few features, it is not well-defined for exploratory analysis, and it operates on a set of observations jointly rather than conditionally (i.e., feature selection from many graphs vs. selection for an input graph *conditioned* on other graphs).**

In this work, we introduce EAGLE (Exploratory Analysis of Graphs with domain knowLEdge), a novel method that creates interpretable, *feature-based*, and *domain-specific* graph summaries in a fully automatic way. That is, the same graph in different domains—e.g., social science and neuroscience—will be described via different EAGLE summaries, which automatically leverage the domain knowledge and expectations. We propose an optimization formulation that seeks to find an interpretable summary with the most representative features for the input graph so that it is: *diverse*, *concise*, *domain-specific*, and *efficient*. Extensive experiments on synthetic and real-world datasets with up to $\sim 1M$ edges and ~ 400 features demonstrate the effectiveness and efficiency of EAGLE and its benefits over existing methods. We also show how our method can be applied to various graph mining tasks, such as classification and exploratory analysis.

I. INTRODUCTION

Technological advances have led to a tremendous increase in the collected data at a finer granularity than ever, including scientific data from different domains that has the potential to lead to new knowledge. Graphs are prevalent in scientific and other data, as they naturally encode various phenomena like structural or functional brain connectivity in neuroscience [8], compounds in chemistry, protein interactions in biology, symptom relations in healthcare [23], behavioral patterns in social sciences, mobility patterns in transportation engineering, and more. However, the size and complexity of these graphs call for statistical and programmatic tools that can harness them. Motivated by this need, we focus on the problem of summarizing graph data in a scalable and domain-aware way, enabling the extraction of intelligible information.

The typical first step of exploring a new graph dataset (e.g., brain connectome; social, technological, or communication network) often involves plotting, fitting, seeking for outliers in, and summarizing the distributions of various graph invariants (or features) such as degree, PageRank, radius, local clustering coefficient, eigenvectors, node attributes, and many more. Univariate and bivariate distributions are often used in graph mining to discover anomalous patterns at the node or graph level ([3], [16], [14]). However, the features to be explored are usually determined in a feature engineering approach, which heavily depends on the analyst's knowledge, intuition, and prior studies. For example, in connectomics, typical features for comparing healthy and non-healthy populations include the average degree, clustering coefficient, path length [6], [8].

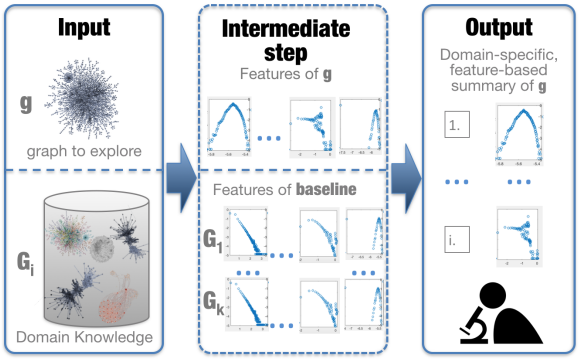


Fig. 1: Overview of EAGLE: Given an input graph g and a set of K baseline graphs G_i that encode the domain knowledge, we seek to find a domain-specific, feature-based summary of g that is diverse, concise, and interpretable. The summary consists of univariate feature distributions (e.g., degree, PageRank).

Moreover, the features selected in existing techniques are determined by the choice of evaluation metrics and are task-dependent. For example, highly correlated features are more likely to be chosen in clustering; independent features are more likely to be chosen for classification. Recent developments in representation learning study latent feature representations via optimization frameworks. Although they are promising and remove the ad-hoc property of feature engineering, they return latent representations which are hard to interpret and are mostly suited for specific tasks such as link prediction and multi-label classification. Therefore, there is need for a general summarization or feature selection technique for exploring graph properties independent of specific tasks.

Proposed Approach: Motivated by these observations, our proposed method, EAGLE, aims to model the exploratory analysis of graph data as a mathematically rigorous feature selection problem which is automatically *guided by* and, thus, *conditioned on the domain* of the data. Throughout the paper, *features* is used to refer to a combination of graph invariants, or structural node attributes (discrete or continuous—e.g., degree, PageRank, clustering coefficient), and categorical or numerical node attributes. Each feature is represented by its (univariate) distribution over the nodes in the graph. Specifically, EAGLE

seeks to summarize an input graph g with the aid of a small set of features by leveraging the information encoded in a set of "baseline" graphs G_i for $i \in \{1, 2, \ldots, k\}$, which, in combination with their invariant distributions, represent the *domain knowledge*.

For instance in Fig. 1, let the input graph be a new social network (g) and the domain contain well-established social networks (G_i). A 'surprising' summary of g would consist of a small set of features including the degree distribution (the leftmost distribution in the central box) which follows the Gaussian distribution, while in the domain a power-law distribution is expected. Our approach can be seen either as feature-based graph summarization, or domain-specific feature selection that seeks to choose some features for an input graph *conditioned on* the features of the baseline graphs. This conditional property sets our work apart from traditional feature selection methods that jointly operate on a set of observations (e.g., select features from multiple graphs).

We formalize the problem as an optimization model that outputs an interpretable, feature-based summary satisfying four important properties: diversity, conciseness, domain specificity, and efficiency. Application-wise, we consider the cases where the number of features in the summary (i) can be defined via prior knowledge or domain expertise, or (ii) need to be defined automatically. Our main contributions are:

• **Novel Formulation:** We propose a new mathematical formulation of graph exploration as a *conditional* feature selection problem over structural or other node attributes. The goal of our proposed constrained optimization framework is to find a diverse, succinct, domain-specific summary for the input graph, which is also interpretable.

• **Scalable Algorithms:** We propose EAGLE-FIX and EAGLE-FLEX, two efficient methods for obtaining the desired summaries. To speed up our methods, we carefully handle the correlations between graph features by systematically investigating their affinities in a data-driven way.

• **Experiments:** We compare EAGLE with baseline approaches on a variety of real-world datasets (including social networks, citation networks, and human connectomes) and show that it satisfies all the desired properties and it is scalable. Although our approach is task-independent, we show that it can be applied to traditional graph mining tasks, such as classification.

For reproducibility, the source code is available at `https://github.com/DerekDiJin/Domain_Knowledge`.

II. Related Work

Our work is related to several research directions:

Feature selection. The process of feature selection consists of two parts: a search technique for proposing new feature subsets, and a measure for evaluating these different feature subsets. Search techniques vary from exhaustive [12] to improved ones, such as greedy hill climbing. Evaluation metrics are divided into three categories: wrappers (which use predictive models to score feature subsets, e.g., [19]), filters (which use measures, such as pointwise mutual information [27]), and embedded methods (which perform selection as part of the model construction process [4]). Our proposed method, EAGLE, is the first approach searching for features greedily based on the domain knowledge and expectations and specifically targeting the graph setting. Moreover, while the above methods select features by jointly learning from all the available observations, our method performs a 'customized' feature selection for a given graph *conditioned* on observations from a set of baseline graphs. Though EAGLE is used for summarizing a dataset with desired properties and there is no particular task guiding its evaluation, we showcase how to adapt it for task-dependent evaluation too.

Pattern mining and Summaries. Mining static graphs often involves analyzing the distributions of specific graph invariants (e.g., skewed degree distribution [9] in numerous settings, small-worldness in connectomics [6], [8]), and speeding up their computations (e.g., betweenness centrality [5]). Moreover, systems [3], [16] have been proposed for anomaly detection via analyzing specific distributions of graph invariants, and spam detection on bivariate distributions. These methods focus on modeling manually-chosen distributions of invariants and potentially finding outliers in them, while our work aims to *automatically detect the features* that summarize a given graph *depending on its domain*. Moreover, we assume that fast methods are used prior to applying EAGLE in order to obtain the distributions of various node invariants. Although EAGLE finds feature-based summaries for an input graph, our work differs significantly from graph summarization [18], [17], which typically seeks to find a compact representation of a network with fewer nodes/links.

Similarity/Distance and Interestingness measures. An excellent review of existing distance/similarity measures for distributions is given in [7]. Attempts to define the interestingness of a plot or distribution by studying its geometric properties [11] include: SCAGNOSTICS [26], which ranks and guides the interactive exploration of bivariate distributions, and motif-based interestingness measures for local patterns in scatterplots [21]. However, unlike our work, these methods are unaware of the domain and introduce generic measures that define the 'interestingness' of each plot independently.

III. Proposed Method

Motivated by the large amounts of graph data and the prevalent need for exploratory analysis in various areas (e.g., neuroscience, social science), we focus on generating interpretable graph summaries by leveraging the domain knowledge:

DEFINITION 1. [Domain Knowledge] We refer to the *expected patterns* (or *laws*) for the distributions of node invariants or other attributes in a specific area as the domain knowledge.

Examples of graph invariants include global structural statistics such as the degree and PageRank; local structural statistics such as the egonet size, interactions to neighbors, and properties revealed by different algorithms such as community detection. In social science, examples of categorical and numerical attributes are the gender and age of a user, respectively.

Our assumption is that the domain expectations are implicitly encoded in a set of *baseline* graphs which belong to that domain. For example, in social networks many distributions of structural attributes (e.g., degree variants, PageRank) are *expected* to follow a power law [9], while in functional connectomes that are produced via neuroimaging techniques more uniform distributions are expected. Based on this definition, we state the problem that we tackle as follows:

PROBLEM. [Exploratory Analysis of Graph Data using Domain Knowledge] Given the node features of a plain or attributed input graph g and a set $\mathcal{G}$ of baseline graphs G_i, $i = 1, \ldots, K$, we seek to find a *domain-specific* summary consisting of a *small set* of *representative* and *interpretable* features in an *efficient way*.

If g is attributed, the features consist of invariants and node attributes. Otherwise, the features include only node invariants. Our main idea is to formulate the exploratory analysis of graphs as an optimization model that will produce as an output a feature-based summary with four desired properties:

- **P1. High Diversity / Coverage.** The summary is required to 'cover' the information or patterns or laws encoded in the baseline graphs: the features in the summary should provide *diverse* aspects of the domain knowledge. We measure *diversity* between the features through the concept of "similarity", so the features in the summary should have trivial dependence.
- **P2. Conciseness.** Although diversity is crucial for good summaries, it connives the "greed" to select features: the most diverse summary should contain many features. To avoid duplication and verbosity, *conciseness* indicates that the number of features in the summary should be small.
- **P3. Domain-specificity.** Based on the information of the baseline graphs $\mathcal{G}$, the summary of g should be related or contrasted to the features of the baseline graphs. For example, if a 'contrasted' summary is required and all the baselines follow a power law degree distribution (e.g., social networks) while g does not, the degree distribution should be included in the summary. However, a 'contrasted' summary in a different domain (e.g., neuroscience) would include different features.
- **P4. Efficiency.** Given the soaring amount of data being generated daily, the computation of the summary must be efficient and scale to large amounts of data.

Moreover, an informal desired property is that the selected features are interpretable and easy-to-understand. To that end, unlike network embedding or factorization-based methods, we seek summaries that do *not* rely on latent features. Next we introduce our proposed optimization framework. For reference, we list the major symbols in Table I.

A. Proposed Formulation

We propose to model the Exploratory Analysis of Graph Data problem as an optimization problem that encodes the above-mentioned desired properties and selects the features to add in the summary such that:

$$\arg\min_{\mathbf{f}} \lambda_1 \underbrace{\mathbf{f}^T \mathbf{S_F} \mathbf{f}}_{\text{1st term}} + \lambda_2 \underbrace{\|\mathbf{f}\|_0}_{\text{2nd term}} + \lambda_3 \cdot \underbrace{\phi(g, G_1, G_2, \ldots, G_K)}_{\text{3rd term}} \quad (1)$$

TABLE I: Table of symbols

Symbol	Definition
$\mathcal{G}$	a collection of baseline graphs, $\mathcal{G} = \{G_1, G_2, \ldots, G_K\}$
g	input graph
K	total number of baseline graphs
F	size of feature space
B	number of buckets in a distribution
$\lambda_{1,2,3}$	regularization parameters
$\mathbf{f}$	$F \times 1$ indicator vector for selected features, $\mathbf{f} \in \{0,1\}^F$
$\mathbf{S_F}$	$F \times F$ pairwise feature relevance matrix for the baseline graphs $\mathcal{G}$
$\mathbf{S_F}_i$	$F \times F$ pairwise feature relevance matrix for baseline graph G_i
$\mathbf{w}$	$K \times 1$ weight vector for the baseline graphs in $\mathcal{G}$, $\sum_i^K w_i = 1$
$\mathbf{h}$	$F \times 1$ vector denoting similarity / distance between equivalent marginal distributions (e.g., degree) of g and $\mathcal{G}$
$s(,), d(,)$	similarity and distance between two objects o_1 and o_2, resp.
$\phi(\cdot)$	coupling function of the input graph g and the baseline graphs $\mathcal{G}$

where $\mathbf{f} \in \{0,1\}^F$ is the vector indicating the selected features; $\mathbf{S_F}$ is the aggregated matrix that represents the pairwise feature relevance in the domain of interest, as encoded in the baseline graphs $\mathcal{G}$; $\|\mathbf{f}\|_0$ is the $l0$-norm of the indicator feature vector; $\phi()$ is a function that couples the input graph g and the baseline graphs, thus grounding the summary to domain; and $\lambda_1, \lambda_2, \lambda_3$ are regularization parameters which are set so that the three terms are comparable (cf. Sec. IV-A).

Intuitively, the **first** quadratic **term**, $\mathbf{f}^T \mathbf{S_F} \mathbf{f}$, forces the selected features to be diverse. It uses the baseline graphs to establish the 'norms' in the domain of interest and uses them to capture the relevance between all pairs of graph invariants. Specifically, $\mathbf{S_F}$ represents the aggregate of the 'correlation' or relevance between all F features over the baseline graphs $\mathcal{G}$, while the quadratic term evaluates the sum of relevance scores of selected features. The regularization parameter λ_1 is set to a positive number (discussed later). Unlike existing work, this term quantifies the relevance between *different* graph invariants (e.g., PageRank and local clustering coefficient) in the domain by harnessing the information in the baseline graphs.

The **second term**, $\|\mathbf{f}\|_0$, which is multiplied by a positive regularization parameter λ_2, requires that the summary is concise, i.e., it consists of a few features. Although, ideally, the $l0$-norm encodes this requirement, we will later relax this constraint to the $l2$-norm which is mathematically tractable.

The **last term**, $\phi(g, G_1, \ldots, G_k)$, is crucial because it couples the input graph g and the domain knowledge. It can be interpreted as the term that forces the features that will be selected for the summary to come as close (or far) as possible to those of the baseline graphs. That way, it can be tuned to provide an 'ordinary/expected' summary or a 'surprising' summary. This is useful when an analyst who knows the information that is being captured in the baseline graphs (e.g., connectomes of subjects with depression) wants to see a holistic overview of the feature-based similarities and possible differences of a newly obtained graph (e.g., connectome of a new subject). When $\phi()$ is a positive, increasing function of $\mathbf{f}$, we have the so-called "0 pit" problem of Equation (1):

DEFINITION 2. [The 0-pit problem] When the three terms of Equation (1) are positive, the solution is $\mathbf{0}^{F \times 1}$ irrespectively of the input and baseline node invariants, i.e., the objective function falls into a "pit" with optimal value 0.

To handle this problem, we add constraints to our optimiza-

tion problem. We elaborate more on the design choices of this term and the additional constraints in Section III-C.

The efficiency of computing the summary comes from our proposed framework, which we discuss in Section IV. The additional (informal) requirement for interpretability follows from our feature representation in $\mathbf{f}$. As opposed to latent representations that are hard to interpret, in our work the selected features correspond to node invariants (e.g., degree, PageRank) or node attributes, which depend on the domain. Throughout our formulation, we assume that the graph features are represented by their PDFs (Probability Density Function) and adapt appropriate measures to quantify their relevance/dissimilarities.

B. Proposed Model for Feature Diversity

As we mentioned above, the first term in our proposed optimization function enforces diversity in the selected features so that they are not correlated. In this subsection we discuss how we design $\mathbf{S_F}$ in order to capture the 'correlation' between the node invariants per baseline graph. Assuming that only the PDFs of the node invariants are provided, computing the correlation between the corresponding invariants is not feasible (more information per node would be needed). Thus, we use feature relevance or similarity between different invariants as a surrogate correlation model.

In general, the features (node invariants) that are considered can be: discrete (e.g., degree distribution) or continuous (e.g., PageRank distribution). If we view each PDF i as a vector of length l_i, it can be seen that different invariants are represented by distribution vectors of different lengths, which leads to two main challenges: (i) What is the right length for each distribution vector, or, put differently, what is the proper size of buckets to be used in different node invariant distributions? and (ii) How can we compute the relevance between two PDFs of different lengths? We address these two questions next.

(i) A general feature representation model. In order to compute the relevance between the features in the baseline graphs, we first need to define the feature model. As we mentioned, we view each feature i as the PDF of the corresponding invariant, which can be represented as a vector of length l_i or, equivalently, l_i 'buckets'. If the PDF is organized in a large number of buckets, the histogram "looks" uniform, while a small number of buckets results in information loss by aggregating many original values into one bucket.

Visualizing the feature distributions involves selecting the number of buckets l_i. For example, for a degree distribution, the number of buckets is equal to the number of unique node degrees, while for a PageRank distribution the number of buckets depends on the analyst and the data at hand. As we see in Fig. 2, the number of buckets is critical when computing the relevance between two features via their PDFs, as they can lead to different 'shapes' of distributions, and help with or prevent the detection of patterns (e.g., spikes). Fig. 2 indicates that a large number of buckets helps show the pattern of discrete PDFs such as the power-law of the out-degree distribution with 10^{-4} range in Fig. 2a, yet a small number of buckets fails to reflect the actual pattern and may miss the spikes that often indicate anomalies. On the contrary, for continuous PDFs, many buckets blur the patterns as the values in the distribution may differ slightly, while fewer buckets may address this problem. This is illustrated through the "uniform" distribution with unique bucketing in Fig. 2b.

We propose to find proper bucket sizing for *any* (discrete or continuous) PDF by adapting Scott's reference rule [20]:

$$\text{Bucket size} = 3.5 \cdot \hat{\delta}/n^{1/3} \tag{2}$$

where $\hat{\delta}$ is the sample standard deviation and n is the number of elements in the distribution. The distribution plots labeled "Scott" in Fig. 2 illustrate the effectiveness of Scott's rule by capturing not only the pattern, but also existing spikes. Scott's rule generates a flexible number of buckets for different PDFs, and it applies to both big and small graphs. There are several variants such as Sturges' formula [25] and Freedman–Diaconis' rule [10], all apply to different settings. For generality, we integrate all these rules including the fixed sizing in the proposed framework and use Scott's rule to conduct computation and experiments.

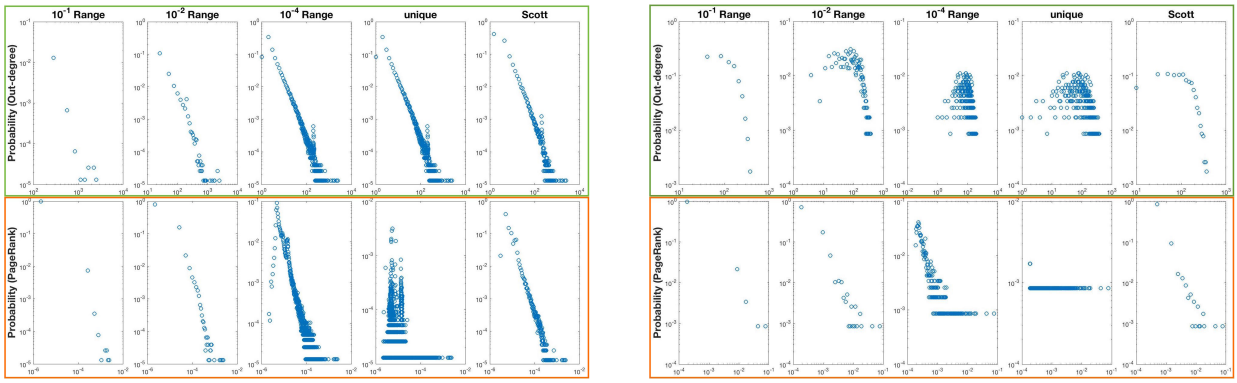
(ii) A surrogate feature correlation model. Assuming that only the PDFs of the node invariants are provided, computing the correlation between the corresponding invariants is not feasible (more information per node would be needed). Thus, we use feature relevance or similarity between different invariants as a surrogate correlation model. Other traditional distance-based measures [7] can be applied when two distribution vectors are of the same length, but, as we saw above, this is usually not the case when dealing with distributions of different invariants, e.g., degree vs. PageRank. For PDFs of different lengths, such as the ones generated by Scott's rule, those measures are not suitable unless they are normalized to have the same length. We discussed the challenges of such normalization above (a general feature representation model).

To emphasize the importance of 'shape' match between distributions of different invariants, and not point-wise match, we propose to leverage the dynamic time warping (DTW) algorithm. DTW is designed to calculate an optimal match between two given sequences by "warping" them non-linearly, so that the distance calculated is independent of variations in the warped dimension. For PDFs that denote the graph statistics distributions, DTW calculates the feature-by-feature distance independent of variations in the number of buckets, which can be converted to similarity in many ways, including $s = (1 + d)^{-1}$. DTW-based similarity works for both cases whether two PDFs are of the same or different lengths.

For generality, we integrate DTW and traditional distance-based methods in the proposed framework and primarily use DTW similarity in our experiments. Per baseline graph G_i, we compute the pairwise feature relevance matrix $\mathbf{S_F}_i$:

$$\mathbf{S_F}_i(f_j, f_l) = s(PDF_{G_i,f_j}, PDF_{G_i,f_l}) \tag{3}$$

where PDF_{G_i,f_j} is the PDF for the j^{th} feature of graph G_i, and $s()$ is the desired similarity between two distributions. By definition, the diagonal elements of each relevance matrix



(a) SOCIAL SCIENCE: SOC-SLASHDOT0811 [22] **(b) Neuroscience: Functional connectome**

Fig. 2: The discrete and continuous PDFs with different bucket sizing, from left to right, the bucket sizing is: $\frac{1}{10}$, $\frac{1}{100}$, $\frac{1}{10000}$ times the range of values; "unique" means the unique values in the PDF; "Scott" refers to the bucket sizing computed by Scott's rule.

are 1. We can obtain the aggregate pairwise feature relevance matrix as their weighted sum:

$$\mathbf{S_F}(f_j, f_l) = \sum_{i=1}^{K} w_i \cdot \mathbf{S_F}i(f_j, f_l) \tag{4}$$

where w_i is the weight or 'importance' of graph G_i in the computation, and $\sum_{i=1}^{K} w_i = 1$.

C. Proposed Model for Domain-Specificity

The **last term**, $\phi(g, G_1, \ldots, G_k)$, in Eq. (1) couples the input graph g and the domain knowledge. Unlike prior work in the literature which focuses on one graph only and assigns interestingness or anomaly scores to a distribution independently of the domain knowledge (e.g., Scagnostics [26]), the third term aims to find the distributions that bear the most or fewest number of similarities with other graphs in the domain.

We propose to model the domain specificity with a simple and intuitive linear formulation, $\phi(g, G_1, \ldots, G_k) = \mathbf{f}^T\mathbf{h}$, where h_j in $\mathbf{h} = [h_1, h_2, \ldots, h_F]$ is the aggregate relation score between the j^{th} marginal distributions (e.g., degree) of g and the baseline graphs G_i. The relation can be set to be a similarity or a distance measure resulting in an 'ordinary' or 'surprising' summary (Sec. IV-A). This choice is directly related to the "0 pit" problem: (i) If $\mathbf{h}$ is modeled as similarity, we need to force the solution of the optimization problem to make selections by adding constraints on $\mathbf{f}$; and (ii) If $\mathbf{h}$ is modeled as distance, the last term becomes negative (i.e., minimizing the 'negative' distance) by setting $\lambda_3 < 0$.

Unlike $\mathbf{S_F}$ which computes the relevance between different invariant distributions of a single graph G_i, $\mathbf{h}$ focuses on the relation between *equivalent* distributions of the input graph g and the baseline graphs G_i. The aggregate relation between the input g and the domain is computed as the weighted average of the relations between all the combinations of g and the baseline graphs G_i. We use h_{sj} to represent the j^{th} entry of the relation vector based on similarity:

$$h_{sj} = \sum_{i=1}^{K} w_i \cdot s(PDF_{g,f_j}, PDF_{G_i,f_j}) \tag{5}$$

Similarly, $\mathbf{h_d}$ represents the relation vector based on a distance measure, and is defined equivalently (by replacing $s()$ with a distance measure $d()$.

IV. EAGLE: PROPOSED ALGORITHM

Our proposed formulation in Optimization Problem 1 corresponds to a mixed-integer quadratic programming (MIQP) problem. The problem of 0–1 integer programming is NP-complete and the integral constraints bring challenges such as intractability and poorly-behaved derivatives, which make algorithms such as gradient descent unwarranted. To solve these challenges, we first explain how we approximate MIQP with a sequence of mixed-integer linear programming (MILP), and then propose two solutions to the "0 pit" problem by adding application-driven constraints in Section IV-A. We give the theoretical analysis on complexity in Section IV-B.

Although the $l0$-norm in Eq. (1) encodes the conciseness requirement, we relax it by using the $l2$-norm, which is mathematically tractable. By rewriting $\|\mathbf{f}\|_2^2 = \mathbf{f}^T\mathbf{f}$ and using the $F \times F$ identity matrix $\mathbf{I_F}$, the equation takes the form:

$$\arg \min_{\mathbf{f} \in \{0,1\}^{F \times 1}} \mathbf{f}^T \underbrace{(\lambda_1 \mathbf{S_F} + \lambda_2 \mathbf{I_F})}_{\mathbf{Q}} \mathbf{f} + \mathbf{f}^T \underbrace{\lambda_3 \mathbf{h}}_{\mathbf{r}}. \tag{6}$$

The integer vector $\mathbf{f}$ can be expressed as the linear constraint to Eq. (6) thus obtaining the form of MIQP:

$$\begin{aligned} \underset{\mathbf{f}}{\text{minimize}} \quad & \mathbf{f}^T\mathbf{Q}\mathbf{f} + \mathbf{r}^T\mathbf{f} \\ \text{subject to} \quad & 0 \le \textstyle\sum_i^F \mathbf{f}(i) \le F \\ & 0 \le \mathbf{f}(i) \le 1,\ i = 1, \ldots, F. \end{aligned} \tag{7}$$

We apply the cutting plane method [15] to convert Problem 7 to a series MILP by introducing a slack variable z:

$$\begin{aligned} \underset{\mathbf{f},z}{\text{minimize}} \quad & z + \mathbf{r}^T\mathbf{f} \\ \text{subject to} \quad & 0 \le \textstyle\sum_i^F \mathbf{f}(i) \le F \\ & 0 \le \mathbf{f}(i) \le 1,\ i = 1, \ldots, F. \\ & \mathbf{f}^T\mathbf{Q}\mathbf{f} - z \le 0,\ z \ge 0 \end{aligned} \tag{8}$$

Problem 8 gives the local MILP approximation to Problem 7 at one step. To further approximate the MIQP, we need to iteratively solve a series of MILP by updating the linear constraints until convergence. To update the linear constraints, we denote $\mathbf{f}$ at the t^{th} iteration as $\mathbf{f}_t$ such that $\mathbf{f}_t = \mathbf{f}_{t-1} + \boldsymbol{\delta}$, where $\mathbf{f}_{t-1}$ is the vector obtained in the previous iteration and $\boldsymbol{\delta}$ is a

variable vector. By using first-order Taylor approximation for the last constraint in Problem (8), we obtain:

$$\begin{aligned}\mathbf{f}_t^T\mathbf{Q}\mathbf{f}_t - z &= \mathbf{f}_{t-1}^T\mathbf{Q}\mathbf{f}_{t-1} + 2\mathbf{f}_{t-1}^T\mathbf{Q}\boldsymbol{\delta} - z + O(|\boldsymbol{\delta}|^2)\\ &= -\mathbf{f}_{t-1}^T\mathbf{Q}\mathbf{f}_{t-1} + 2\mathbf{f}_{t-1}^T\mathbf{Q}\mathbf{f}_t - z + O(|\mathbf{f}_t - \mathbf{f}_{t-1}|^2)\\ &\approx -\mathbf{f}_{t-1}^T\mathbf{Q}\mathbf{f}_{t-1} + 2\mathbf{f}_{t-1}^T\mathbf{Q}\mathbf{f} - z \le 0,\end{aligned}$$

where we omitted the second-order terms. Thus, to solve the MIQP of Problem (7), we need to solve a series of MILPs in Problem (8) combined with this updated linear constraint.

A. EAGLE: *Application-driven Constraints*

As we mentioned in Section III-A, the last term of Eq. (6) can be tuned to provide an 'ordinary/expected' summary or a 'surprising' summary by identifying features that are similar or dissimilar to the ones in the baseline graphs, respectively. During exploratory analysis, this allows for some flexibility about the type of relevance that is sought between the summary of g and the baseline graphs $\mathcal{G}$. Next, without loss of generality, we focus on surprising summaries, and introduce two application-driven constraints: (i) fixed, and (ii) flexible number of features in the summary. Our analysis can be easily extended to the case of ordinary summaries as well.

A1. EAGLE-FIX: Fixed number of features. In the case of creating a surprising summary for the input graph, the last term in Eq. (6) can be set such that $\mathbf{h}$ captures similarities between the features of g and G_i, i.e., it is computed based on Eq. (5) and denoted by $\mathbf{h_s}$. To solve the 0-pit problem, we introduce a capacity constraint for the summary, in addition to the constraints that are given in Problem (7), and set $\mathbf{r} = \lambda_3\mathbf{h_s}$:

$$\textstyle\sum_i^F \mathbf{f}(i) = c \qquad \text{[}\textit{new}\text{ capacity constraint]} \qquad (9)$$

To prevent the objective function from reaching the optimum with some desired properties overwhelming the others, $\lambda_{\{1,2,3\}}$ should be set such that the three terms in Optimization Problem 1 are comparable (i.e., of the same scale). The values of these normalization terms are primarily determined by the maximums of (i) $\mathbf{f}^T\mathbf{S_F}\mathbf{f}$, (i) $\|\mathbf{f}\|_2^2$, or $\mathbf{f}^T\mathbf{f}$ and (iii) $\mathbf{f}^T\mathbf{h}$. We discuss the parameter setting in the experiments (Sec. V).

Putting everything together, in the case of finding surprising summaries for a given input, we propose the EAGLE-FIX algorithm, for which we give the pseudocode in Algorithm 1.

A2. EAGLE-FLEX: Flexible number of features. In the case of creating a surprising summary for the input graph, we can search for a flexible number of features by setting the last term in Eq. (6) such that it captures the distances between the features of g and G_i (i.e., $\mathbf{h} = \mathbf{h_d}$) and $\lambda_3 < 0$. Note that a very small value λ_3 may render the third term smaller than other terms, which would lead the objective function to fall into the "0 pit". Therefore, to determine the regularization parameters in this case, we propose a different technique that obtains the range of λ_3 based on λ_1 and λ_2 values.

• *Upper bound for* λ_3. Suppose there are $c \ge 0$ selections in the solution $\mathbf{f}$. Then, the value of the relaxed objective function (6) can be calculated as:

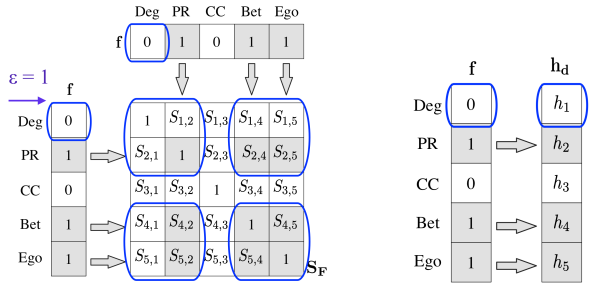
$$\lambda_1 \textstyle\sum_{i,j\in\mathcal{S}} S_F(i,j) + \lambda_2 c + \lambda_3 \sum_{i\in\mathcal{S}} h_d(i) \qquad (10)$$

Algorithm 1 EAGLE-FIX

Input: Graph g with F invariant distributions; Graph database with G_i $(i = 1 \dots K)$ graphs with their F invariant distributions
Output: Binary vector $\mathbf{f}$ of selected features in the summary of g

I. Preprocessing Phase: Computations over the Domain
for $i = 1 \dots K$
  // Step 1: Feature Representation Model
  for $j = 1 \dots F$
    $PDF^{new}_{G_i,f_j} = \text{Scott}(PDF_{G_i,f_j})$ ▷ Scott's rule, Eq. (2)
  // Step 2: Feature Diversity Model
  for $j = 1 \dots F$, and $l = j+1 \dots F$
    $\mathbf{S_F}_i(f_j, f_l) = s(PDF^{new}_{G_i,f_j}, PDF^{new}_{G_i,f_l})$ ▷ Eq. (3)
$\mathbf{S_F}(f_j, f_l) = \sum_{i=1}^K w_i \cdot \mathbf{S_F}_i(f_j, f_l)$ ▷ Eq. (4)

II. Query Phase: Summary Creation
Step 1: Domain-specificity Model
for $l = 1 \dots F$
  $h_{sl} = \sum_{i=1}^K w_i \cdot s(PDF^{new}_{g,f_l}, PDF^{new}_{G_i,f_l})$ ▷ Eq. (5)
Step 2: Feature Selection
$\mathbf{Q} = \lambda_1\mathbf{S_F} + \lambda_2\mathbf{I_F}$ ▷ Regularization parameters $\lambda_1, \lambda_2, \lambda_3$
$\mathbf{r} = \lambda_3\mathbf{h}_s$
$\mathbf{f} = \text{MIQP}(\mathbf{Q}, \mathbf{r})$ ▷ Solve Problem (7)



(a) First term: $\mathbf{f}^T\mathbf{S_F}\mathbf{f}$ **(b) Third term: $\mathbf{f}^T\mathbf{h_d}$**

Fig. 3: Example: $\mathcal{S} = \{2,4,5\}$, $\mathbf{f} = \{0,1,0,1,1\}$, and degree as the newly added feature (i.e., $\epsilon = 1$). (a) The sum of the shaded areas in $\mathbf{S_F}$ corresponds to the first term. After adding the degree, i.e., $\mathcal{S}' = \mathcal{S} \cup \{1\}$, the sum of the blue rectangles correspond to the first term. (b) Blue rounded rectangles in h_d indicate $h_d(\epsilon)$; The sum of its shaded cells gives the third term.

where $\mathcal{S}$ denotes the collection of the indices of selected features $\mathbf{f}$, which is explained in Fig. 3. When $c = 0$, $\mathcal{S} = \emptyset$ Similarly, when there are $c+1$ selections, the value of the objective function is:

$$\lambda_1 \textstyle\sum_{i,j\in\mathcal{S}'} S_F(i,j) + \lambda_2(c+1) + \lambda_3 \sum_{i\in\mathcal{S}'} h_d(i) \qquad (11)$$

where $\mathcal{S}' = \mathcal{S} \cup \{\epsilon\}$, and $\{\epsilon\}$ denotes the index of the newly selected feature. Our proposed Optimization Problem 1 will only select $c+1$ features if that further reduces the objective function, which implies that Eq. (10) > (11), or:

$$\begin{aligned}\lambda_3 &< -\frac{\lambda_1(\sum_{i,j\in\mathcal{S}'} S_F(i,j) - \sum_{i,j\in\mathcal{S}} S_F(i,j)) + \lambda_2}{\sum_{i\in\mathcal{S}'} h_d(i) - \sum_{i\in\mathcal{S}} h_d(i)} \Rightarrow\\ \lambda_3 &< -\frac{\lambda_1(\sum_{i\in\mathcal{S}} S_F(i,\epsilon) + \sum_{i\in\mathcal{S}} S_F(\epsilon,i) + 1) + \lambda_2}{h_d(\epsilon)}\end{aligned} \qquad (12)$$

By assuming that ϵ corresponds to the maximum entry in $\mathbf{h_d}$, we obtain the upper bound of λ_3:

$$\lambda_3 < -\frac{\lambda_1 + \lambda_2}{\max(\mathbf{h_d})} \qquad (13)$$

• *Lower bound for* λ_3. By requiring the three terms in the optimization problem to be comparable, we can obtain a lower bound for λ_3. Assuming that $\lambda_3 < 0$ and $c = |\mathcal{S}|$, the third term must be smaller or equal to the maximum of the others:

$$\lambda_3 > -\frac{\max\{\lambda_1 \sum_{i,j\in\mathcal{S}} S_F(i,j), \lambda_2|\mathcal{S}|\}}{\sum_{i\in\mathcal{S}} h_d(i)} \quad (14)$$

Inequality (14) indicates that the lower bound of λ_3 is determined by $\mathcal{S}$ (it is involved in all the terms of (14)). In order to find the exact lower bound, we need to consider all possible sets of $\mathcal{S}$ (or equivalently, all possible binary vectors $\mathbf{f}$), which are $O(2^F)$. Thus, to reduce the complexity of its computation we provide an empirical lower bound, which works well in practice:

$$\lambda_3 \geq -\lceil \frac{\lambda_1 + \lambda_2}{\max(\mathbf{h_d})} \rceil - 1 \quad (15)$$

We discuss the choices of λ_1, λ_2 and λ_3 more in Section V.

B. Complexity

The runtime of EAGLE consists of three parts: (1) computing $\mathbf{S_F}$, (2) computing $\mathbf{h}$, and (3) runtime of MIQP. In the first two parts, the runtime τ of computing similarity / distance between two PDFs is determined by the distance measure. Although τ can be affected by the lengths of PDFs, it is generally trivial.

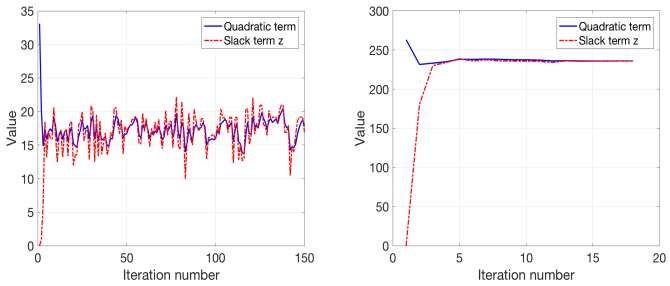
(1) Since $\mathbf{S_{F_i}}$ is symmetric with diagonal elements equal to 1, the number of similarity computations for one baseline graph is $O(F^2)$. $\mathbf{S_F}$ aggregates K of them, so the complexity for $\mathbf{S_F}$ is $O(\frac{KF(F-1)\tau}{2})$.

(2) The feature-by-feature relation between g and G_i constructs the $\mathbf{h_i}$ vector with $O(F)$ similarity computations. Then, $\mathbf{h}$ aggregates K of them, resulting in $O(KF\tau)$ complexity.

(3) The runtime complexity of MIQP depends on the speed of convergence between the quadratic term and its linear approximation. If the convergence criterion is not reached, EAGLE would run with every possible value of $\mathbf{f}$ to reach the minimum, which is $O(2^F)$. However, empirical experiments show that in general EAGLE takes about 20∼30 iterations to reach satisfying approximation, if not converging. This is illustrated in Fig. 4, where we set the maximum number of iterations to be 150. Interestingly, we observe that the MIQP runtime does not only depend on the length of vector $\mathbf{f}$, but also on the values of entries in $\mathbf{f}$: If the values are small and close to each other, MIQP would require more comparisons to find the path towards the optimum (Fig. 4a); On the contrary, if the values differ tremendously, this procedure becomes much faster (Fig. 4b).

V. EXPERIMENTS

In this section we provide thorough experimental analysis to evaluate our proposed approach. Specifically, we consider the evaluation metrics: **(1)** The satisfaction of the desired properties for exploratory analysis (P1-P3); **(2)** The scalability of EAGLE algorithm (P4); and **(3)** Its robustness to the required parameters. Moreover, we present an application of EAGLE to a graph mining task, namely the classification of patients (Schizophrenic) and healthy subjects based on fMRI data.



(a) `HepPh` **citation graph: 21 features** **(b)** `Slashdot0922` **social graph: 300 perturbed features**

Fig. 4: Convergence of two runs with MIQP.

A. Baselines

No systematic empirical research exists that addresses the problem of finding graph summaries by automatically leveraging domain knowledge. Moreover, as we discussed in Section II, unlike traditional feature selection methods that choose features by jointly operating on a set of observations, our method is 'conditional': It selects features for an *input* graph conditioned on observations from *other* graphs (domain knowledge). Despite these limitations in the literature, we evaluate the effectiveness of EAGLE against:

• **RANDOM:** This approach randomly selects a subset of features as the summary of the input graph. It is often used as the preliminary analysis given little or no prior knowledge.

• **SCAGNOSTICS [26]:** This method was proposed to summarize high-dimensional datasets by detecting anomalies in density, shape, and trend. Since it applies on bivariate distributions, we modified it to compute 9 measures (area of convex hull, skinniness, stringiness, straightness, monotonic score, skewness, clumpy score, striation, and binning score) on each one of F univariate distributions. Features with the top score in at least one measure are included in the summary.

• **SURPRISING:** This method is a special case of EAGLE with $\lambda_1 = \lambda_2 = 0$ and detects patterns that are different (or surprising) from the ones that appear in the baseline graphs.

B. Datasets

The real datasets that we used in our experiments are from three different domains: connectomics, citation networks, and social science. We give short descriptions of these datasets in Table II. The first two connectomes, `Brain-Voxel1` and `Brain-Voxel2`, were generated using the traditional network discovery [6] process: (i) computation of the pairwise correlations between the 3789 time series obtained during fMRI and (ii) application of threshold ($\theta = 0.9$) to keep the most significant associations and get sparse networks.

C. Experimental setup

EAGLE is implemented in MATLAB, and all the experiments were performed on a laptop equipped with an Intel Core i7-4870HQ Processor and 16GB memory.

EAGLE takes an arbitrary number of graph features as input and outputs a small set of representative features as a summary

TABLE II: Domains and graphs used in our experiments.

Domain	Name	Nodes	Edges	Description
Connectomics [1]	Brain-Voxel1	3 789	399 069	undirected unweighted
	Brain-Voxel2	3 789	148 648	undirected unweighted
	COBRE [2]	1 166	∼679 000	undirected unweighted
Citation networks [22]	HepTh	27 770	352 807	directed unweighted
	HepPh	34 546	421 578	directed unweighted
Social science [22]	Epinions	75 879	508 837	directed unweighted
	Slashdot0811	77 360	905 468	directed unweighted
	Slashdot0922	82 168	948 464	directed unweighted

based on the domain knowledge. The features used in the experiments include 28 common node- and structure-specific invariant distributions and other graph properties: The *node-specific features* that we used are in-degree, out-degree, PageRank, in-closeness, out-closeness, hubs, authorities, clustering coefficient, betweenness, top eigenvectors, network constraint, and roles [13]. The *structure-specific statistics* comprise egonet features, such as out-degree, out-neighbors, in-degree, in-neighbors, and size of egonets in edges and nodes. Moreover, we considered other features, such as the distribution of communities, weak / strong connected components, in / out-going community affiliations, motifs, community profiles, random left and right singular values, and hops [22].

Equation (6) defines the relationships between the regularization terms λ_1, λ_2, and λ_3: By observing that the maximum values of the three terms are F^2, F and F, respectively, we define the relationship between the regularization terms $F\lambda_1 = \lambda_2$ and $\lambda_2 = \lambda_3$. For EAGLE-FIX, we set $\lambda_1 = \frac{1}{F}$, $\lambda_2 = 1$, and $\lambda_3 = 1$; for EAGLE-FLEX, we set $\lambda_1 = \frac{1}{F}$, $\lambda_2 = 1$, and compute λ_3 according to Equation (15). We use the default value $\mathbf{w} = \{\frac{1}{K}\}^{F\times 1}$ to weigh the contribution of baseline graphs. However, if prior information is available, the weights can be set differently as long as $\sum_i^K w_i = 1$.

D. Satisfaction of Desired Properties

In this experiment, we quantitatively evaluate the satisfaction of the desired properties by EAGLE and the baselines. We obtain EAGLE summaries for two input graphs, HepPh and Slashdot0922, considering three domains with different sets of baseline graphs: (i) connectomics using Brain-Voxel1 and Brain-Voxel2; (ii) citation networks using HepTh; and (iii) social networks using Epinions and Slashdot0811. For fairness, we set all the methods to give the same number of features as SCAGNOSTICS, and thus run EAGLE-FIX. To evaluate the conciseness of our method, we present experiments in Section. V-F. For all the methods, we evaluate the diversity and domain-specificity ('surprising') of the selected features $\mathbf{f}$ via correlation: We compute the pairwise feature correlation matrix between the F univariate distributions (with the same binning) of the baseline graphs, $\mathbf{C}_\mathcal{G}$, and quantify diversity as $\mathbf{f}^T\mathbf{C}_\mathcal{G}\mathbf{f}$. Similarly, based on the correlation matrix $\mathbf{C}'_\mathbf{g}$ between the input graph and the baseline features, we quantify domain-specificity as $\mathbf{f}^T\mathbf{C}'_\mathbf{g}$. For completeness, we apply three different correlation coefficients: Pearson's, Kendall's Tau, and Spearman's Rank. Figure 5 illustrates the results for Pearson's correlation (dark shades for diversity, light for domain specificity). Similar patterns are

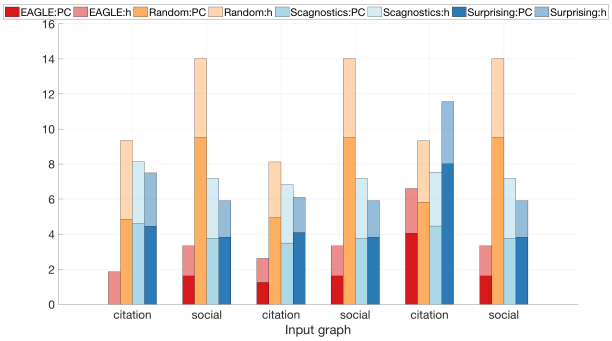


Fig. 5: Effectiveness in terms of diversity and domain-specificity evaluated using Pearson's correlation coefficient (low values are better). EAGLE achieves the best performance in every case.

detected by using the other two metrics, which are omitted for brevity (they can be found in our code repository).

Diversity. Diversity is measured using pairwise feature correlation in $\mathbf{C}_\mathcal{G}$, so lower values indicate higher diversity. The results show that EAGLE outperforms all the baselines in every case. We observe an extreme case: the summary of HepPh conditioned on citation networks yields almost 0 Pearson correlation value. This demonstrates the effectiveness of EAGLE in selecting features that are diverse especially when the baseline graphs and the input are very similar.

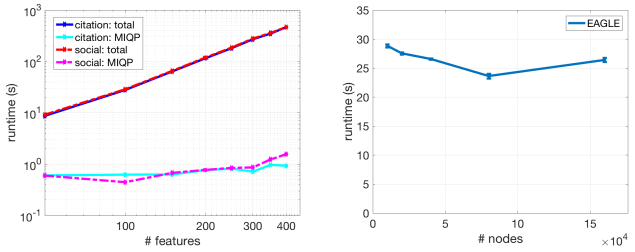
Domain-Specificity. Similar to diversity, we explore 'surprising' patterns of the input graph with respect to the baselines via the feature-wise correlation (low values correspond to high domain-specificity). Figure 5 shows that EAGLE outperforms all the baselines by up to $\sim 51.74\%$. Qualitatively, the clustering coefficient distribution and community size distribution are always selected when the input graph and the baseline graphs are from different domains. Intuitively, this is reasonable because the community structure differs in graphs from different domains and both properties are related to it.

E. Scalability

We evaluate the scalability of the proposed methods with regard to (a) number of features, and (b) size of the baseline graphs. Here we extend the feature space beyond the original 28 by creating 'perturbed' features with up to 30% random noise.

Number of features. We create a mixed domain containing the citation graph (HepTh) and two social graphs (Epinions and Slashdot0922) as baselines, and run EAGLE-FLEX to summarize two input graphs: HepPh and Slashdot0811, with the number of features (original and perturbed) varying from 50 to 400. In Fig. 6a, we observe that, for both input graphs, EAGLE-FLEX scales linearly and almost identically with the number of features in the semi-logarithmic plot, which indicates its quadratic complexity. Moreover, given identical number of features, the runtime of MIQP on different input graphs is almost the same.

Size of baseline graphs. In this experiment we test the scalability in terms of the size of baseline graphs for a fixed number of selected features. We create a series of synthetic datasets with feature space including 7 global invariant distributions and 13 perturbed invariants. The sizes of the



(a) Number of features **(b) Size of baseline graphs**

Fig. 6: Scalability of EAGLE-FLEX on two input graphs (citation and social). (a) In both cases, EAGLE-FLEX scales quadratically in terms of the number of features with similar behavior of MIQP (b) The runtime is independent of the size of baseline graphs.

synthetic graphs constructed are 10K, 20K, 40K, 80K, and 160K. The runtime of EAGLE-FLEX on these datasets is shown in Fig. 6b. We observe a relatively "flat" pattern in running time, which indicates that the optimization solver in EAGLE is independent of the size of baseline graphs. Note that there is some fluctuation in the curve: the running time of EAGLE-FLEX on 40K graphs is the shortest, while that on 10K is the longest. Despite the presence of randomness, this phenomenon points to one direction of our future work, which is to explore the behavior of MIQP in EAGLE on large-scale graphs.

F. Robustness to parameters

We run EAGLE-FLEX to evaluate the sensitivity of regularization parameters $\lambda_{\{1,2,3\}}$ and the corresponding conciseness of the summary. The baseline graphs are `HepTh`, `Slashdot0811` and `Brain-Voxel1`, and a total of 28 features are generated (no perturbation). Per regularizer, we perform a grid search over $\{\frac{1}{32}, \frac{1}{16}, \ldots, 16, 32\}$ times its default value (Sec. V-C), while keeping the other regularizers at their default values. We plot the number of selected features in the summary and the percentage of common selected features between that summary and the 'default' summary (based solely on the default values). These quantities are illustrated as a function of the regularizer values in Fig. 7. We note that we do not depict the percentage for the default value (marked with '*') in the blue curve, since it is 100% by definition.

The blue curves in Figures (7a)-(b) show that values of regularization around the default give relatively stable results, with 50%∼80% identical features to the default setting. Figure (7c) shows that the selection of features is stable up to the default value, but sensitive for larger λ_3 for which the last term dominates (and thus puts more emphasis on 'surprising' patterns). From the red curves, we observe that the default values lead to few selected features, indicating the conciseness (property P2) of the EAGLE summaries.

G. Case study: classification on brain graphs

How can EAGLE be applied to graph classification, a traditional data mining task? We focus on the domain of neuroscience, and use `COBRE` [2], a dataset from the NIH Center for Biomedical Research Excellence with resting-state fMRI data from 72 patients with schizophrenia and 76 healthy controls. From the 1166 fMRI time series (avg. length 100 timesteps), we created undirected, weighted graphs with $\theta = 0.6$ following the traditional method [6] (cf. Sec. V-B).

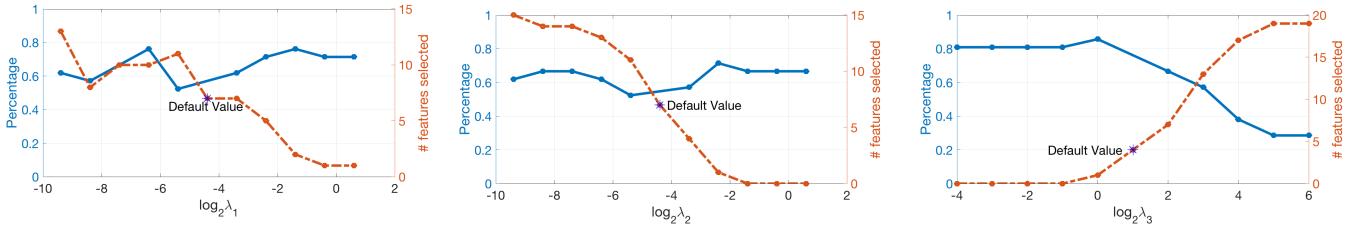
TABLE III: Classification on `COBRE`: AUC scores per method.

Method Category	`Unweighted`		`Weighted`	
	Ordinary	Surprising	Ordinary	Surprising
EAGLE-FLEX	0.6893	0.5499	0.7096	**0.7296**
EAGLE-FIX: 6	0.5114	0.5445	0.6961	**0.7371**
EAGLE-FIX: 8	0.6795	0.5904	**0.7216**	0.7079
EAGLE-FIX: 10	0.5003	0.4989	**0.7032**	0.6807
Full	-	-	0.6681	0.7147
Baselines	Baseline 1:	0.7028	Baseline 2:	0.1099

The task is to use the EAGLE summaries to classify the healthy controls and patients. We create the feature space by calculating the distributions of 11 features: weighted and unweighted degree, PageRank, closeness, eigenvector, clustering coefficient, betweenness, neighbors of the egonets, degree of the egonets, and sizes of egonets in edges and nodes. To obtain feature representations that can be used for classification, we used a random set of 36 healthy subjects as the baseline graphs, and ran both EAGLE-FIX (with $F = \{6, 8, 10\}$) and EAGLE-FLEX on the remaining graphs (40 controls and 40 patients) and obtained both 'surprising' and 'oridinary' summaries for them. We consider two vector representations for the summaries: (i) `Unweighted`: a binary vector $\mathbf{b}$ with 1s for the selected features by EAGLE; and (ii) `Weighted`: a real vector with the importance of each selected feature, i.e., $\mathbf{b} \odot \mathbf{h}$ where $\mathbf{h}$ is given in Eq. (5) (or its distance-based counterpart), and $\odot$ denotes component-wise multiplication. We also consider 'Full', which uses vector $\mathbf{h}$ as the representation of each connectome (without feature selection).

As baselines we considered two traditional methods in neuroscience: (a) per connectome, a vector representation with the mean of each feature distribution [8] and (b) a 'flat', vectorized ($1 \times N^2$) representation of the $N \times N$ adjacency matrix of the connectome [24]. For the classification task, we trained an SVM classifier that uses the RBF (radial basis function) kernel on the vector representations of our methods and the baselines, by conducting 10-fold cross validation. Table III gives the accuracy (AUC) of each method.

According to Table III, we have two observations: (1) Without knowing anything about the dataset, EAGLE-FLEX provides promising performance on the task of classification, although EAGLE-FIX outperforms EAGLE-FLEX with some explicit settings on F. The EAGLE-FLEX and EAGLE-FIX summaries lead to better performance than the baseline methods, indicating the fact that although not designed explicitly for this, features selected by EAGLE can be applied to specific tasks such as classification; (2) Compared with the use of all weighted features (Full) and selection (EAGLE-FLEX), we observe that the latter improves the performance over the former by eliminating the noise contained in the dataset, which demonstrates the effectiveness of selected features. Qualitatively, among the 11 features, PageRank is the most frequently picked feature by EAGLE-FLEX. Weighted and



(a) Sensitivity to λ_1. **(b) Sensitivity to λ_2.** **(c) Sensitivity to λ_3.**

Fig. 7: Robustness of EAGLE to the regularization parameters. Left y axis: percentage of identical selected features between λ and its default value. Right y axis: total number of invariant distributions included in the summary.

unweighted degree are the most distinguishable features when running EAGLE-FIX that are never picked in summaries for controls, but are selected for patients.

VI. CONCLUSION

We propose a novel way to summarize a graph using a set of informative, interpretable features, resulting in a diverse, concise, domain-specific, and efficient-to-compute summary. Our novel formulation targets early data exploration and provides an alternative to the feature engineering process that is often a part of graph mining tasks. We frame the problem as constrained optimization, based on 'conditional feature selection, which is tailored to the domain expectations and knowledge, in contrast to existing work which views each graph as a unit independent of its domain or many graph observations as a whole. We also introduce two efficient algorithms, EAGLE-FIX and EAGLE-FLEX, which handle the correlations between graph features and find summaries that are fixed or flexible in size. Our experiments show that the EAGLE variants are effective, their summaries satisfy all the desired properties, outperform alternative approaches that can be cast to solve this problem, and they are effective in data mining tasks such as classification despite not being tailored to it. Future work may explore extensions to more complex design choices or bivariate distributions of features (often used in spam detection), as well as scaling the method up more.

ACKNOWLEDGEMENTS

The authors thank Dr. Chandra Sripada for sharing the brain network data and the anonymous reviewers for their insightful comments. This material is based upon work supported by the National Science Foundation under Grant No. IIS 1743088 and the University of Michigan. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] Penn dataset. http://www.humanconnectome.org/ccf/.
[2] Center for Biomedical Research Excellence. http://fcon_1000.projects.nitrc.org/indi/retro/cobre.html, 2012.
[3] L. Akoglu*, D. H. Chau*, U. Kang*, D. Koutra*, and C. Faloutsos. OPAvion: Mining and Visualization in Large Graphs. In *SIGMOD*, pages 717–720, 2012.
[4] F. R. Bach. Bolasso: model consistent lasso estimation through the bootstrap. In *ICML*, pages 33–40. ACM, 2008.
[5] D. A. Bader, S. Kintali, K. Madduri, and M. Mihail. Approximating betweenness centrality. In *WAW*, pages 124–137, 2007.
[6] E. Bullmore and O. Sporns. Complex Brain Networks: Graph Theoretical Analysis of Structural and Functional Systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.
[7] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *J MMMAS*, 1(2):1, 2007.
[8] Z. Dai and Y. He. Disrupted structural and functional brain connectomes in mild cognitive impairment and alzheimer's disease. *Neuroscience Bulletin*, 30(2):217–232, 2014.
[9] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-law Relationships of the Internet Topology. *SIGCOMM*, pages 251–262, 1999.
[10] D. Freedman and P. Diaconis. On the histogram as a density estimator: L 2 theory. *Probability theory and related fields*, 57(4):453–476, 1981.
[11] L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3), Sept. 2006.
[12] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR*, 3(Mar):1157–1182, 2003.
[13] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, and L. Li. RolX: Structural Role Extraction & Mining in Large Graphs. In *KDD*, pages 1231–1239, 2012.
[14] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang. Catchsync: Catching synchronized behavior in large directed graphs. In *KDD*, pages 941–950, 2014.
[15] J. E. Kelley, Jr. The cutting-plane method for solving convex programs. *J Appl Math*, 8(4):703–712, 1960.
[16] D. Koutra, D. Jin, Y. Ning, and C. Faloutsos. Perseus: an interactive large-scale graph mining and visualization tool. *VLDB Endowment*, 8(12):1924–1927, 2015.
[17] D. Koutra, U. Kang, J. Vreeken, and C. Faloutsos. VOG: Summarizing and Understanding Large Graphs. In *SDM*. SIAM, 2014.
[18] Y. Liu, A. Dighe, T. Safavi, and D. Koutra. Graph Summarization: A Survey. *CoRR*, abs/1612.04883, 2016.
[19] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE TPAMI*, 27(8):1226–1238, 2005.
[20] D. W. Scott. On optimal and data-based histograms. *Biometrika*, pages 605–610, 1979.
[21] L. Shao, T. Schleicher, M. Behrisch, T. Schreck, I. Sipiran, and D. A. Keim. Guiding the exploration of scatter plot data using motif-based interest measures. In *BDVA*, pages 1–8, 2015.
[22] SNAP. http://snap.stanford.edu/data/index.html#web.
[23] P. Sondhi, J. Sun, H. Tong, and C. Zhai. SympGraph: a framework for mining clinical notes through symptom relation graphs. In *KDD*, pages 1167–1175, 2012.
[24] C. S. Sripada, D. Kessler, R. Welsh, M. Angstadt, I. Liberzon, K. L. Phan, and C. Scott. Distributed effects of methylphenidate on the network structure of the resting brain: a connectomic pattern classification analysis. *Neuroimage*, 81:213–221, 2013.
[25] H. A. Sturges. The choice of a class interval. *Journal of the American Statistical Association*, 21(153):65–66, 1926.
[26] L. Wilkinson, A. Anand, and R. L. Grossman. Graph-theoretic scagnostics. In *INFOVIS*, volume 5, page 21, 2005.
[27] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420, 1997.