Scalable Heterogeneous Graph Neural Networks for Predicting High-potential Early-stage Startups

Shengming Zhang[1], Hao Zhong[2], Zixuan Yuan[1], Hui Xiong[1*]
[1]Rutgers University, [2]ESCP Business School
shengming.zhang@rutgers.edu,hzhong@escp.eu,{zy101,hxiong}@rutgers.edu

ABSTRACT

It is critical and important for venture investors to find high-potential startups at their early stages. Indeed, many efforts have been made to study the key factors for the success of startups through the topological analysis of the heterogeneous information network of people, startup, and venture firms or representation learning of latent startup profile features. However, the existing topological analysis lacks an in-depth understanding of heterogeneous information. Also, the approach based on representation learning heavily relies on domain-specific knowledge for feature selections. Instead, in this paper, we propose a *Scalable Heterogeneous Graph Markov Neural Network* (SHGMNN) for identifying the high-potential startups. The general idea is to use graph neural networks (GNN) to learn effective startup representations through end-to-end efficient training and model the label dependency among startups through Maximum A Posterior (MAP) inference. Specifically, we first define different metapaths to capture various semantics over the heterogeneous information network (HIN) and aggregate all semantic information into a summated graph structure. To predict the high-potential early-stage startups, we introduce GNN to diffuse the information over the summated graph. We then adopt an MAP inference over Hinge-Loss Markov Random Fields to enforce label dependency. Here, a pseudolikelihood variational expectation-maximization (EM) framework is incorporated to optimize both MAP inference and GNN iteratively: The E-step calculates the inference, and the M-step updates the GNN. For efficiency concerns, we develop a GNN with a lightweight linear diffusion architecture to perform graph propagation over web-scale heterogeneous information networks. Finally, extensive experiments and case studies on real-world datasets demonstrate the superiority of SHGMNN.

CCS CONCEPTS

• **Applied computing → Business intelligence**; • **Computing methodologies → Neural networks**; **Maximum a posteriori modeling**.

KEYWORDS

Business Intelligence, Startup Success Prediction, Graph Neural Networks, Heterogeneous Information Networks, Representation Learning, Graph Embedding, Graph Mining, Markov Random Fields

ACM Reference Format:
Shengming Zhang[1], Hao Zhong[2], Zixuan Yuan[1], Hui Xiong[1]. 2021. Scalable Heterogeneous Graph Neural Networks for Predicting High-potential Early-stage Startups. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3447548.3467383

1 INTRODUCTION

Startups are *Big Smalls*, small scales yet with big potentials. As pointed out by GEM[1], entrepreneurship is a truly powerful engine for economic and social development, generating incomes and jobs while enabling and enriching individuals and communities. As a major driving force to job creation, high-growth startups accounted for only a few percentage of the firms' population but tended to create about 60% of new jobs across most countries and sectors [34]. Compared with incumbents, who are more likely to invest on existing technologies and incremental innovations, startups are more inclined to promoting disruptive and revolutionary innovations. Meanwhile, venture capital investors are typically anticipating significant financial returns by investing on the startups with great potential to grow and successfully exit. However, early-stage startups are immature and fragile in the sense that only a significant small portion can survive and grow. It has never been an easy task to identify high-potential startups despite its great economic value and vital societal meanings.

Over the past decades, scholars from various research communities have made great efforts on addressing high-potential startups prediction problem. Their methodologies can be roughly grouped into two categories: (i) *Qualitative approaches* that process startups' profile information into features based on knowledge from interviews, surveys and experts' inputs [16, 17, 23, 24], typically followed by downstream supervised-learning tasks [3, 5, 32, 33, 35, 44]. These study outcomes rely heavily on people's retrospection which is subject to relationalization, resulting in post-hoc biases. The downsides of these approaches include the requirement of in-depth domain-specific knowledge and overlooking the interactions between different entities involved in entrepreneurial activities. (ii) *Topological approaches* that are using interactions between objects to construct an information network and further extract network topological features for identifying high-potential early-stage startups [4, 15, 18, 42]. However, by using solely the topological features,

[*] Hao Zhong and Hui Xiong are both Corresponding Authors. This work was partially supported by the National Science Foundation through awards III-2006387, IIS 1814510, and IIS-2040799.

KDD '21, August 14–18, 2021, Virtual Event, Singapore.
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467383

[1] Global Entrepreneurship Monitor 2019/2020 Global Report, Global Entrepreneurship Research Association, 2020

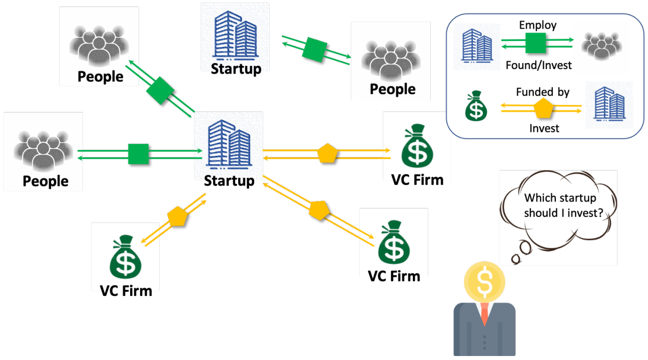


Figure 1: Illustrations of the semantic structure of our Heterogeneous Business Information Network (HBIN).

these approaches fail to fully employ abundant intrinsic information over the network.

Instead, we propose to approach the problem by utilizing Heterogeneous Graph Neural Network (HGNN) model due to its successful applications on a wide-range of research problems involving Heterogeneous Information Networks (HIN) [36, 40, 43]. Regardless of its strength in distilling information from heterogeneous graphs, one essential drawback is that the predictions are based solely on the aggregation of node attributes, overlooking the joint dependency among object labels. To alleviate this problem, [27] proposes a Graph Markov Neural Network (GMNN) that utilizes Statistical Relational Learning (SRL) to model the object label dependency. However, directly applying GMNN for identifying high-potential startups remains at least three challenges: (i) GMNN is initially designed for homogeneous graphs, while most real-world business networks are heterogeneous with multiple types of entities involved (as one example shown in Figure 1). (ii) GMNN uses Graph Convolutional Network (GCN) to model object label dependency, whereas previously reported improvement of using GCN for label capacity is minor, near to that by a mean-pooling heuristic. (iii) Another key issue of using GCN to model label capacity is its "short sight" - aggregating the information from the nodes' neighbors with order up to the number of stacked layers. Its incapability of balancing between scalability and capacity restricts its broader applications on web-scale business network problems.

To better overcome the aforementioned deficits of GMNN, we propose a *Scalable Heterogeneous Graph Markov Neural Network (SHGMNN)* to address the proposed high-potential early-stage startups prediction problem. Specifically, we regard the prediction of high-potential startups as a node classification task on the heterogeneous information network (HIN). We use metapath instance counts between startups as closeness measures, and apply trainable weight parameters to learn the relative importance between different metapaths. The parameterized summated adjacent graph structure is used as input of GNN. We conduct a Maximum A Posterior (MAP) inference on the Hinge-Loss Markov Random Fields (HL-MRFs) to predict unlabeled nodes, then the inference results are to assist GNN parameters updating. This framework can be efficiently optimized using the variational EM framework, alternating between an inference procedure (E-step) and a learning procedure (M-step). In the E-step, the unlabeled nodes are predicted based on MAP inference, while in the M-step, both labeled and unlabeled nodes are fed into GNN for parameter updating. To improve the scalability of our model, we develop a lightweight structured GNN with only one linear diffusion layer and adopt a fast convex optimization algorithm for MAP inference.

Finally, we unify all the components into the SHGMNN framework for high-potential early-stage startup prediction. The main technical contributions of our work are as follows:

- We propose a technique that generates the *Parameterized Summated Adjacent Matrix* to integrate and capture the relative importance of multiple semantic metapaths.
- We propose an MAP inference over Hinge-Loss Markov Random Fields for label capacity that can be calculated through fast convex optimization algorithms.
- We propose a general SHGMNN framework that could take benifit of both GNN and MAP inference under large-scale heterogeneous graph settings.
- Extensive experiments and case studies are conducted on real-world datasets, demonstrating the effectiveness of our proposed SHGMNN model to handle the unique properties of the startup success prediction problem.

2 RELATED WORKS

Business Success Prediction: The problem of predicting early-stage startups' financing success has long been a hot topic in finance and management research communities. Early studies in those fields investigated various factors which might have potential impacts on new ventures' success [39]. These works laid a solid foundation for further feature engineering approaches [9, 22]. Typically, these studies start from crawling a real-world business dataset with a properly-defined problem, followed by carefully engineered features using domain-specific knowledge or heuristic guidelines. The handcrafted features are then fed into extant machine learning models to generate predictive results, such as decision tree [3, 33], logistic regression [5], naive Bayesian network [22], neural networks [32], etc. There are also studies aiming to seek signals of business success using topological features in the network, where most common topological features are centrality-like ones [4, 15, 18].

Heterogeneous Graph Neural Network: A Graph Neural Network focuses on learning effective object representations for label prediction problems. In particular, a Heterogeneous Graph Neural Network (HGNN) is a special type of GNNs designed for Heterogeneous Information Networks (HIN) which can essentially exploit the characteristics of heterogeneity [19, 31, 36, 37, 40, 43]. There is one specific type of HGNN that uses metapath as a tool for information integration on HINs. Conceptually, metapaths in HIN can be utilized in three different manners: (i) using the metapath-based neighbors as adjacency information for the node type of interest [12, 36], (ii) applying a metapath-based sampling technique on the heterogeneous information network [10], and (iii) designing the algorithm to automatically find important metapaths [19, 40]. Note that there is also another strand of research focusing on enhancing GNN's scalability to large-scale real-world data [6, 7, 20, 29, 41, 45].

Table 1: Mathematical Notations

Symbol	Description
B	$B = \{B_1, B_2, ..., B_M\}$ Set of objects
M	Number of types of objects
$G = (V, E)$	A graph with nodes V and edges E
P	$P = \{P_1, P_2, ...P_T\}$ Set of metapaths
T	Number of defined metapaths
R	$R = \{R_1, R_2, ..., R_S\}$ Relationships of objects on G
S	Number of relationships between objects
$p(u, v)$	A metapath instance connecting node u and v
$pc(u, v)$	Number of metapath instances between u and v
PC	$PC = \{PC_1, PC_2, ..., PC_T\}$ Metapath count matrices
ψ	Potential function defined over edges
Y_V, Y_L, Y_U	Label for all/labeled/unlabeled nodes
X_V	Attributes for all the nodes
Θ, Ω	GNN model parameters
N_i, N_i^+, N_i^-, N_i^u	All/positive/negative/unlabeled neighbors of i
$[X_i] \mid\mid [X_j]$	Concatenation of two vectors
p_ϕ	Parameters to be updated in M-Step.
q_θ	Parameters to be updated in E-Step.
α	$\alpha = \{\alpha_1, \alpha_2, ..., \alpha_T\}$ Weight for each metapath
$\sigma(), \xi()$	Activation functions
$\mathcal{A}$	Parameterized summated adjacent matrix

Statistical Relational Learning: In the study of statistical relational learning (SRL), most research methods model label dependency using probabilistic graphical models, such as Markov networks [11, 13, 14, 21, 28]. Typically, the following formulation is employed for modeling the node label distribution Y_V conditioned on the node features X_V with conditional random fields:

$$p(Y_V|X_V) = \frac{1}{Z} \prod_{(u,v)\in E} \psi_{u,v}(y_u, y_v, X_V), \tag{1}$$

where $(u, v) \in E$ is an edge in graph $G(V, E)$, and $\psi_{u,v}(y_u, y_v, X_V)$ is the potential score defined on the edge and Z is the normalization factor. The potential score is typically defined as linear combinations of variables and given values, such as logical formulae[2]. With this formulation, predicting the labels for unlabeled objects becomes an inference problem, i.e., inferring the posterior label distribution of the unlabeled objects $p(Y_U|Y_L, X_V)$, in which Y_U and Y_L are the sets of unlabeled and labeled nodes, respectively. Exact inferencing is usually infeasible due to the complex structures between object labels, whereas approximate inferencing methods are favored.

In sum, SRL methods are solid in modeling label dependency over the entire graph and keeps high interpretability. However, given its inefficiency of inferencing over complex relational structures, SRL models may suffer the performance-scalability issue.

3 PRELIMINARIES

In this section, we give formal definitions of some important terminologies pertinent to our work. Table 1 summarizes the methematical notations appeared in this paper.

Definition 3.1. **Heterogeneous Information Network:** Given a list types of objects $B = \{B_1, B_2, ..., B_M\}$, where each type B_i contains n_i nodes: $\{b_{i,1}, b_{i,2}, ...b_{i,n_i}\}$. Graph $G =< V, E >$ is called a *heterogeneous information network* on types B, if $V(G) = B$ and $E(G) = \{< b_i, b_j >\}$, where $b_i, b_j \in B$.

Definition 3.2. **Metapath:** A metapath P is defined as a path that describes a composit relation between different types of nodes. A metapath of length l has a form of $B_1 \xrightarrow{R_1} B_2 \xrightarrow{R_2} ... \xrightarrow{R_l} B_{l+1}$, denoting there is a composition of relations $R = R_1 \diamond R_2 \diamond ... \diamond R_l$ between node type B_1 and B_{l+1}. Each relation $R_i^{|B_i|\times|B_{i+1}|}$ is the adjacent matrix between object B_i and B_{i+1}. The total number of relations l is the length of the metapath.

Definition 3.3. **Metapath Instance:** A metapath instance $p = (u, v)$ represents a specific path instance of metapath P that connects node u and v.

Definition 3.4. **Metapath Count:** A metapath count $pc(u, v)$ under metapath $P = B_1 \xrightarrow{R_1} B_2 \xrightarrow{R_2} ... \xrightarrow{R_l} B_{l+1}$ is the total number of metapath instances connecting node u and v.

Definition 3.5. **Metapath Count Matrix:** A metapath count matrix PC under metapath $P = B_1 \xrightarrow{R_1} B_2 \xrightarrow{R_2} ... \xrightarrow{R_l} B_{l+1}$ is an adjacent matrix between B_1 and B_{l+1}, where $PC_{u,v} = pc(u, v)$. A metapath count matrix can be calculated as multiplications of relations along the metapath:

$$PC = R_1 \times R_2 \times ... \times R_l. \tag{2}$$

Specifically, if B_1 and B_{l+1} are of the same type of node, the metapath count is a symmetric square matrix.

Graph Neural Network: In the view of statistical learning, GNN-based methods model the joint distribution as:

$$p_\Theta(Y_V|X_V) = \prod_{v\in V} p_\Theta(y_v|X_V). \tag{3}$$

For a one-layer GNN, the conditional probability for each node $p(y_v|X_V)$ is:

$$p_\Theta(y_v|X_V) = \sigma\left(\sum_{s\in N_i} Softmax(\Theta[X_n] \mid\mid [X_s])\right), \tag{4}$$

where Θ is a trainable model parameter, $||$ is the concatenation operator and $\sigma()$ is an activation function. GNN can be trained in an end-to-end manner which updates the model parameter Θ given a subset of labeled nodes. From the equation we can see that the prediction of GNN is based on only aggregation of nodes and its neighbors' attributes, ignoring label dependency.

4 SHGMNN FRAMEWORK

In this section, we introduce a novel Scalable Heterogeneous Graph Markov Neural Network (SHGMNN) model for supervised node classifications on HINs. The key idea is to take the advantages of both GNN and SRL: SRL ensures label capacity that GNN neglects, while GNN helps improve predictability over SRL-based methods. In particular, we propose to address two major chanllenges here:

- **C1**: Although extant heterogeneous GNN approaches can capture heterogeneous network information, it is inevitably confronted with the scalability issue especially on large-scale networks. Thus we state Challenge 1 (**C1**): **How to**

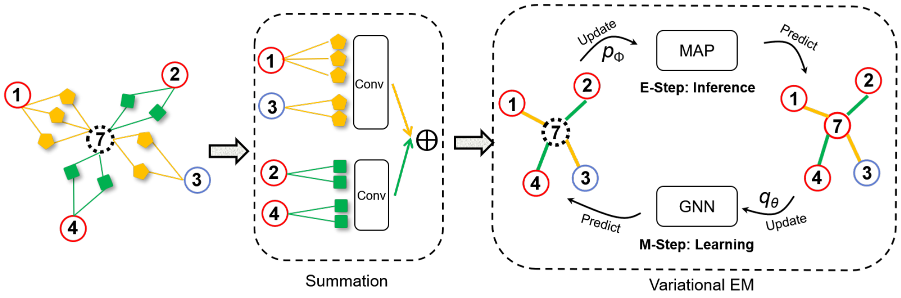


Figure 2: The architecture of the Scalable Heterogeneous Graph Markov Neural Network (SHGMNN). Red circled nodes are labeled as positive, blue circled nodes are labeled as negative and black-dotted circled nodes are unlabeled. The original heterogeneous graph is first transferred into a parameterized summated adjacent graph structure using metapath-count and 1-d convolutions, followed by a variational EM framework. In E-Step, the MAP Inference predicts unlabeled nodes, and pass result to GNN. In M-Step, the GNN updates parameters based on E-Step's output, and make predictions for MAP's parameters update.

incorporate heterogeneity while keeping model complexity under control?

- **C2**: We expect SRL can model label capacity and potentially outperform GCN. However, inferencing on a network with complex node relations can be computationally expensive. Thus we state Challenge 2 (**C2**): **How to properly model label capacity using low-cost SRL methods?**

4.1 Overview

Our proposed SHGMNN framework models a joint distribution $p(Y|G,\phi)$ of object labels Y conditioned on the heterogeneous network G and model parameters ϕ. In order to optimize the objective, we use a pseudolikelihood variational EM framework. In E-step, a Maximum A Posterior (MAP) inference over Hinge-Loss Markov Random Fields (HL-MRFs) is used to ensure dependency of object labels. In M-step, a GNN is used to integrate heterogeneity as well as to leverage the object attributes. The predicted labels in E-step together with the ground-truth labels are employed for GNN's parameter updating. By using a linear diffusion operator technique and a fast convex optimization computing algorithm, our SHGMNN framework can be deployed on large-scale heterogeneous information networks. Figure 2 shows the overall architecture of SHGMNN. We will present the details of SHGMNN in the following sections.

4.2 Pseudolikelihood Variational EM:

We rewrite Equation (2) into a parameterized joint distribution of object labels conditioned on node attributes: $p_\phi(Y_V|X_V)$, where ϕ denotes model parameters. For simplicity, we ignore the specific formulation of potential functions for now. It is difficult to directly learn ϕ by maximizing the log-likelihood function $logp_\phi(Y_L|X_V)$, since the object labels are partially observed. Instead, we optimize the evidence lower bound (ELBO) of the log-likelihood function:

$$E_{q_\theta(Y_U|X_V)}\left[logp_\phi(Y_L,Y_U|X_V) - logq_\theta(Y_U|X_V)\right], \tag{5}$$

where $q_\theta(Y_U|X_V)$ can be any distributions over Y_U. According to [25], the ELBO can be optimized using variational EM algorithm that alternates between a variational E-step and an M-step. In the variational E-step, the goal is to fix p_ϕ and update the variational distribution $q_\theta(Y_U|X_V)$ to approximate the true posterior distribution $p_\phi(Y_U|Y_L,X_V)$, in which the objective is:

$$O_\theta = \sum_{n\in U} E_{p_\phi(y_n|X_V)}\left[logq_\theta(y_n|X_V)\right] + \sum_{n\in L} logq_\theta(y_n|X_V). \tag{6}$$

In the M-step, q_θ is fixed and p_ϕ is updated to maximize the following objective function:

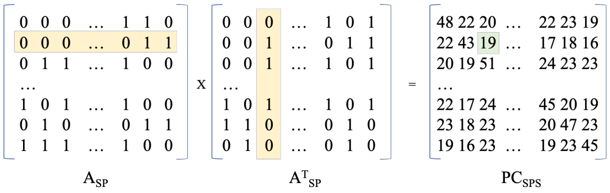
$$O_\phi = \sum_{n\in U} logp_\phi(q_\theta(y_n|X_V)|X_V) + \sum_{n\in L} logp_\phi(y_n|X_V), \tag{7}$$

where U and L are the sets of unlabeled and labeled nodes respectively. For labeled nodes, the supervised log-likelihood objective function is optimized, while for unlabeled nodes, the log-likelihood is approximated by the inference resulted from the E-step. It is worth to note that if θ is empty, the variational E-step will become a regular E-step in which no parameters updates are needed.

4.3 M-step: GNN with Parameterized Adjacent Matrix Using Meta-path Count-Based Heterogeneity Summation

Some of the existing HGNN methods [19, 40, 43] groups all types of nodes on the heterogeneous information network, generating a very large node space. However, in most cases only a certain type of nodes are of our interests, corresponding to a very small proportion of the entire node space. Our startup success prediction problem is a typical example that only *Startup* nodes are of our particular concern. Accordingly, a meta-path count-based summation strategy is introduced to effectively reduce the node space.

Consider a heterogeneous information network $G(V,E)$ with two types of nodes: *Startup* (S) and *People* (P). A metapath starts from a *Startup* node, traverses through its *People* neighbors, then ends at connected *Startup* nodes, i.e. $S \xrightarrow{R_1} P \xrightarrow{R_2} S$. Here R_1 can represent the relation of a startup hiring a person, while R_2 can denote a startup founded by a person. Such a metapath can be denoted in short as "SPS". As introduced in Section 3, we could use matrix multiplications to calculate the total number of metapath instance counts between different nodes, which generates the



$$\underset{A_{SP}}{\begin{bmatrix} 0 & 0 & 0 & \dots & 1 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 1 \\ 0 & 1 & 1 & \dots & 1 & 0 & 0 \\ \dots & & & & & & \\ 1 & 0 & 1 & \dots & 1 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 0 & 0 \end{bmatrix}} \text{x} \underset{A^{T}_{SP}}{\begin{bmatrix} 0 & 0 & 0 & \dots & 1 & 0 & 1 \\ 0 & 0 & 1 & \dots & 0 & 1 & 1 \\ 0 & 0 & 1 & \dots & 1 & 0 & 1 \\ \dots & & & & & & \\ 1 & 0 & 1 & \dots & 1 & 0 & 1 \\ 1 & 1 & 0 & \dots & 0 & 1 & 0 \\ 0 & 1 & 0 & \dots & 0 & 1 & 0 \end{bmatrix}} = \underset{PC_{SPS}}{\begin{bmatrix} 48 & 22 & 20 & \dots & 22 & 23 & 19 \\ 22 & 43 & 19 & \dots & 17 & 18 & 16 \\ 20 & 19 & 51 & \dots & 24 & 23 & 23 \\ \dots & & & & & & \\ 22 & 17 & 24 & \dots & 45 & 20 & 19 \\ 23 & 18 & 23 & \dots & 20 & 47 & 23 \\ 19 & 16 & 23 & \dots & 19 & 23 & 45 \end{bmatrix}}$$

Figure 3: Demonstration of the metapath count process using matrix multiplications. The path-count matrix PC_{SPS} is derived from the multiplications between A_{SP} (The adjacent matrix from *Startup* nodes to *People* nodes) and its transposed matrix.

metapath count matrix:

$$PC_{SPS} = A_{SP} \times A_{SP}^{T}, \tag{8}$$

where A_{SP} is the adjacent matrix from *Startup* nodes to *People* nodes. Figure 3 illustrates the procedure of calculating the meta-path count matrix (PC) of metapath "SPS" with adjacent matrices multiplications. We select a set of symmetric metapaths which starts and ends at nodes of our interests, generating a set of symmetric metapath count matrices. Each matrix corresponding to one type of metapath can be treated as a closeness adjacency measure between nodes of interest under a certain semantic meaning. To leverage the weights of different semantic meta-paths, we introduce a parameterized summation over the normalized meta-path count matrices:

$$\tilde{PC}_{soft} = \sum_{t \in T} \alpha_t \times \tilde{PC}_t, \tag{9}$$

where $\tilde{PC}_t = D^{\frac{1}{2}} PC_t D^{-\frac{1}{2}}$ is the normalized metapath count matrix, $D_{ii} = \sum_j PC_{ij}$ is the row-summated diagnal matrix. T is the set of selected metapaths and α is a trainable relative weight parameter constrained by a softmax function among all the meta-path count matrices. To simplify the notation, we let $\mathcal{A} = \tilde{PC}_{soft}$. The above operation is equivalent to applying a 1-d convolutional layer over the normalized adjacent matrices, resulting in a parameterized weighted adjacent matrix to represent the overall adjacency between each nodes. The parameterized summated adjacent matrix can be input of any form of Graph Neural Network architectures and it empowers any GNN to handle heterogeneity. In the M-step of variational EM framework, we seek to update the parameters of the GNN model together with the weight parameters that maximizes the objective O_ϕ. More specifically, we will use both groundtruth labels and inference results from E-Step to update the parameters $\phi = \{\alpha, \Theta\}$, Θ denotes to the parameters of GNN.

4.4 E-Step: Maximum A-Posterior (MAP) Inference

At this stage, we form a Hinge-Loss Markov Random Fields [1] (HL-MRFs) over the graph. A Hinge-Loss Markov Random Fields can be formulated as a log-concave conditional probability density function over the graph:

$$P(Y_A|X_A) = \frac{1}{Z} exp(-\sum_{r=1}^{R} \lambda_r \psi_r(Y_A, X_A)), \tag{10}$$

where λ represents the set of weight parameters and Z a normalization constant. The potential function $\psi_r(Y, X)$ for HL-MRFs is specifically defined as:

$$\psi_r(Y, X) = (\max(l_r(Y, X), 0))^p, \tag{11}$$

where l_r is a linear function of Y and X while $p \in \{1, 2\}$. We define specifically three types of potential functions in our problem settings: *positive*, *negative* and *unknown* potentials. The *positive* potential function is defined as:

$$\psi_P(Y, X) = \sum_{i \in Y_U} \sum_{j \in N_i^+} \mathcal{A}(i, j)(\max(1 - y_i, 0))^p, \tag{12}$$

where $\mathcal{A}(i, j)$ represents the edge weight of i, j in the parameterized summated adjacent matrix. N_i^+ is the set of positively labeled neighbors of i. This potential function assigns higher probabilities for any unlabeled nodes with positive neighbors, since the higher y_i becomes, the smaller the "*distance-to-satisfaction*" is, i.e. non-negative hinge-loss decreases. Similarly, we have negative constraint potential function:

$$\psi_N(Y, X) = \sum_{i \in Y_U} \sum_{j \in N_i^-} \mathcal{A}(i, j)(\max(y_i - 0, 0))^p, \tag{13}$$

where N_i^- is the set of negatively labeled neighbors of i.

The above two potential functions jointly constrains the label capacity given nodes' first-order neighbors, which is equivalent to a single layer GCN. Previous work[27] needs multiple stacks of GCN layers to consider neighbors of higher orders. We solve by using a third potential function to constrain the unlabeled nodes not only with labeled neighbors, but also with unlabeled neighbors by forcing neighboring unlabeled nodes to be close in probabilities:

$$\begin{aligned} \psi_U(Y, X) &= \sum_{i \in Y_U} \sum_{j \in N_i^u} \mathcal{A}(i, j)(\max(y_i - y_j, 0))^p + \max(y_j - y_i, 0))^p \\ &= \sum_{i \in Y_U} \sum_{j \in N_i^u} \mathcal{A}(i, j)|y_i - y_j|^p, \end{aligned} \tag{14}$$

where N_i^u is the set of unlabeled neighbors of i. By adding the third potential function, the label capacity can be propagated to the entire graph. More rigorously, the conditional probability density function of HL-MRFs based on the three pre-defined potential functions can be formally written as:

$$\begin{aligned} P(Y|X) = \frac{1}{Z} exp(-\sum_{i \in Y_U} (\lambda_P & \sum_{j \in N_i^+} \mathcal{A}(i, j) \times max(1 - y_i, 0)^p \\ & + \lambda_N \sum_{j \in N_i^-} \mathcal{A}(i, j) \times max(y_i - 0, 0)^p \\ & + \lambda_U \sum_{j \in N_i^u} \mathcal{A}(i, j) \times |y_i - y_j|^p)), \end{aligned} \tag{15}$$

where λ_P, λ_N and λ_U are the weight parameters corresponding to each potential function. The *Maximum A-Posterior* (MAP) inference can be calculated by maximizing the conditional probability density function, which is equivalent to minimizing the sum potentials. Given the hinge-loss potentials are convex, the inference can be optimized to reach the global optimal solution efficiently. The predicted labels will be used by GNN in the back-propagation stage, updating the parameters. Using MAP inference, we can ensure label dependency which is overlooked by the GNN model and therefore properly address Challenge 2 (**C2**).

4.5 Scalable Optimization

Till now, we have fully presented the general architecture of our proposed SHGMNN framework. In this subsection, we will introduce how to optimize our SHGMNN in a scalable setting so that it is applicable to large-scale real-world datasets. It consists of two parts: efficient GNN model design and fast MAP inference calculation.

Shallow-structure GNN with Linear Diffusions: Previous work [38] demonstrated that a "shallow" model with single GCN layer can have the performance on a par with other "deep" models. Meanwhile, [29] introduced a strategy of concatenating multiple diffusion matrices in one linear layer in order to capture the importance between nodes' neighbors under different diffusion operators. We adopt the similar idea by using three diffusion operators: simple adjacent matrix, Personalized PageRank-based adjacency, and triangle-based adjacency matrices. All diffusion matrices (together with their powers) can be pre-calculated before the training stage. Given only one single linear layer, both the training batch time and inference time are significantly shorter than most of the sampling-graph-based approaches, such as ClusterGCN [7] and GraphSAINT [41]. Formally, we have:

$$\begin{aligned} Z &= \sigma([X\Theta_0, A_1X\Theta_1, ..., A_rX\Theta_r]), \\ Y &= \xi(Z\Omega). \end{aligned} \tag{16}$$

Here, $A_1, A_2, ..., A_r$ are linear diffusion matrices calculated based on $\mathcal{A}$, $\Theta_0, ..., \Theta_r$. Ω is a set of learnable parameters. σ and ξ are activation functions. Note that the product of the diffusion matrices and node features can be pre-calculated to avoid redundant computations.

In addition, our meta-path count-based summation strategy significantly reduces the node space by preserving only the nodes of interest, generating a parameterized adjacent matrix that can be fed into any form of GNN architectures. Since the 1-d convolutional and the linear diffusion contain very few parameters, our proposed network structure is rather efficient in comparison with other HGNN approaches, effectively addressing Challenge 1 (**C1**).

Parallel Convex Optimization Using ADMM Algorithm: The design of our MAP Inference over Hinge-Loss Markov Random Fields makes the inference objective convex, which allows acceleration using fast convex optimization algorithms. In order to calculate the MAP convex inference over a large-scale graph, we adopt the *Alternating Direction Method of Multipliers* (ADMM) algorithm which divides the overall convex optimization problem into small fractions, which are subsequently addressed in parallel manner and forced to an agreement progressively.

5 DATASET

5.1 Data Source

To carry out our experiments, we start by collecting information from the selected business dataset and transfer them into graph-structured data. Among several potential business data sources, such as Crunchbase[2], Owler[3], Preqin[4] and S&P Capital IQ[5], Crunchbase is widely recognized as the leading database of business and investment activities [8], especially for early-stage startups. Crunchbase data is sourced mainly through two channels: large investor network and community contributors. More than 3,000 global investment firms upload monthly portfolio updates to Crunchbase, in exchange for free data access [8]. In a benchmark of comparison between multiple business data sources, Chrunchbase demonstrated its fairly comprehensive coverage. Therefore, we target Crunchbase as the primary data source for constructing our business information network. Specifically, Crunchbase data contains information about venture capital investments, startup founding members and individuals in leadership positions, mergers and acquisitions (M&A) events, media news, industry trends, etc. Among them, we gather historical investment records, startup firmographics, and members' profiles to build our Heterogeneous Business Information Network.

5.2 HBIN Construction

In order to construct the Heterogeneous Business Information Network $G = (V, E)$, we target three types of entities commonly involved in entrepreneurial activities: *VC Firm*, *Startup* and *People*. These entities are regarded as different types of *nodes* in our heterogeneous business information network, i.e. $G(V) = \{S, V, P\}$. The interactions between entities are *edges*. We extract three business interactions between different entities:

- *VC Investments:* If a *Startup s* was funded by a *VC Firm v*, there is an edge $(s, v) \in E$.
- *People Investment:* If a *Startup s* receives investment from a *Person p*, there is an edge $(s, p) \in E$.
- *Employment:* If a *Startup s* hires a *Person p*, or a *Person* founded a $Startup\ s$, there is an edge $(s, p) \in E$.

Label Assignment: We model the early-stage startups success prediction problem as node classifications on the constructed HBIN, bringing up two questions:

1. *Which stage in startup's life cycle is regarded as early-stage?*
2. *What is the proper indicator of success for early-stage startups?*

Extensive studies have conducted on studying the life cycle of startups [26, 30]. According to [26], the development of a typical startup can be divided into multiple stages, such as pre-startup stage, early stage, growth stage, etc. Early-stage startups are generally referred to those who have received at least one funding round, e.g. seed round, yet without any profit. When startups start to demonstrate their capability of generating revenue, such as launching their first products, VC investors will typically consider further rounds of investments which are typically called series funding rounds. The first round of such equity crowdfunding is *Series-A* funding, then *Series-B,C,D*, etc. As our targets, the early-stage startups are defined as those companies who have *received at least one round of investment but no Series-A funding*. Meanwhile, given that receiving series-A funding is a significant milestone for an early-stage startup signifying its growth and expansion, we therefore define *whether an early-stage startup could reach Series-A funding round*

[2] https://www.crunchbase.com/
[3] https://corp.owler.com/
[4] https://www.preqin.com/
[5] https://www.spglobal.com

Table 2: Statistics of two datasets.

	CB-2013	CB-2021
#Startups	7,374	6,741
#Positively Labeled Startups	1,697	1,200
#Negatively Labeled Startups	5,677	5,541
#People	29,991	20,250
#VC Firms	2,509	6,260
#S-P	35,515	22,544
#S-V	7,052	14,313

as an indicator of success. Note that in rare cases, the companies who has not received Series-A funding but eventually got acquired by other companies are also considered “successful”.

5.3 Dataset Description

From Crunchbase, we gather two snapshots of data samples for our experiments. The first is a snapshot CB-2013 that contains all investment activities occurring no later than Dec 31th, 2013. Meanwhile, to evaluate our model on the latest business dynamics, we have collected up-to-date data from Crunchbase website as our second dataset CB-2021. In the aim of tracking startup behaviors consistently, we excluded the startups founded over 10 years ago in the respective snapshots (i.e. those earlier than Jan 1, 2004 for CB-2013 and those earlier than Feb 1, 2011 in CB-2021). On the other hand, we observed in CB-2013, 86.54% of startups receive *Series-A* investment within 2 years since establishment, and likewise 81.23% of startups in CB-2021. We therefore decided to exclude startups founded less than 2 years to ensure necessary performance period. To avoid information leakage, we ruled out all the investment records after *Series-A* (included) investment rounds. The statistics of the two snapshots are shown in Table 2.

6 EXPERIMENTS

6.1 Experimental Settings

6.1.1 Baselines. To better demonstrate the performance of our **SHGMNN** model, we include a wide range of state-of-the-art baselines algorithms. They can be divided into four groups: (i) methods using only node attributes or topological features (**Random, Centrality, Metapath2Vec**) (ii) metapath-based heterogeneous neural network methods (**HAN, GTN**), (iii) methods using SRL for label capacity (**GMNN, MAP**), and (iv) methods with scalable settings (**SGC, SIGN**). The descriptions of baselines are presented below. Detailed evaluation protocols and experimental setup can be found in the appendix. We also make our code publicly available at GitHub[6].

- **Random:** Randomly assign value to unlabeled nodes with respect to the label distribution in the training set.
- **Closeness Centrality [4]:** It is a topological feature-based approach using closeness centrality for node ranking. This represents the state-of-the-art method for graph-based startup success prediction. We also assign labels with respect to the label distribution in the training set.
- **Metapath2Vec [10]:** It is a metapath-based skip-gram model for heterogeneous network embedding.
- **Heterogeneous Graph Attention Network (HAN) [36]:** It is a metapath-neighbor-based graph neural network with two attention layers.
- **Graph Transformer Networks (GTN) [40]:** It is a graph neural network that automatically extracts metapaths with multiple graph transformer layers.
- **Graph Markov Neural Network (GMNN) [27]:** This approach uses SRL to assist GNN with label capacity on the homogeneous graph. This represents stat-of-the-art SRL-assisted GNN method.
- **MAP Inference (MAP) [1]:** It is a pure SRL method that infers unlabeled nodes using MAP inference on Hinge-Loss Markov Random Fields.
- **Simplified Graph Convolution (SGC) [38]:** It is a simplified graph convolutional neural network for scalable setting.
- **Scalable Inception Graph Neural Networks(SIGN) [29]:** It is a scalable GNN model that precomputes the diffusions under different operators over graph.

6.2 Experimental Results

6.2.1 Overall Performance. To demonstrate the effectiveness of our model, we first compare our **SHGMNN** with all the baseline methods on performing high-potential early-stage startup predictions. Table 3 presents the experimental results, from which we can see some interesting facts. First, the performance of **SHGMNN** surpasses the baseline methods on all evaluation metrics on CB-2013. It demonstrates the superior capability of our proposed framework on predicting high-potential early-stage startups. Second, our **SHGMNN** obtains much higher overall Precision@K than other baselines, which exhibits strong ranking ability of our model on potential candidates, and thus more applicable for real-world scenarios. Third, regarding the comparison between **HAN** and **GTN**, we can see that different strategies of using metapaths influencing the final performance differently, which manifests the necessity of new heterogeneity integration techniques.

6.2.2 Ablation Study. We also conduct some ablation experiments to check how each part of our model affects the final results. We experiment by ablating different parts of our framework, and included the results in Table 3. To clarify, **SHGMNN-S** is **SHGMNN** preserving parameterized summation and ruling out the MAP inference while **SHGMNN-M** is **SHGMNN** without parameterized summation, but keeping only the MAP inference. From the results, we can draw the following arguments: (1) it is clear that both weight learning and MAP inference help improve the model performance to certain degrees; (2) the MAP inference provides important ranking support for most-likely candidates, indicated by a significant increase of Precision@K values; (3) using a parameterized summation strategy, the overall performance in classification correctness is certainly improved, including both accuracy and AUPR score.

6.2.3 Time Efficiency. In order to compare the convergence speed for GNN-based algorithms, we show their validation accuracy on CB-2021 dataset as a function of runtime in Figure 4. We observe that **SHGMNN** does not only reach a higher validation accuracy,

[6] https://github.com/mmichaelzhang/SHGMNN

Table 3: Overall performance.

Dataset	CB-2013							CB-2021						
	Classification Correctness				**Precision@K**			**Classification Correctness**				**Precision@K**		
Algorithm	*F1*	*AUC*	*AUPR*	*Accuracy*	*P@10*	*P@50*	*P@100*	*F1*	*AUC*	*AUPR*	*Accuracy*	*P@10*	*P@50*	*P@100*
Random	23.14%	49.64%	24.75%	64.20%	26.00%	26.60%	26.90%	19.36%	50.01%	19.09%	67.74%	15.00%	18.80%	18.90%
Centrality	23.53%	49.03%	24.26%	60.97%	40.00%	28.00%	25.00%	20.14%	48.14%	18.06%	65.17%	10.00%	16.00%	16.00%
Metapath2Vec	9.85%	51.38%	25.84%	73.47%	20.00%	22.00%	28.00%	2.22%	51.17%	19.83%	81.59%	40.00%	14.00%	20.00%
HAN	33.93%	59.96%	31.61%	77.29%	71.00%	65.8%	55.20%	23.12%	51.89%	20.45%	82.06%	53.00%	40.30%	35.50%
GTN	32.78%	59.20%	34.78%	79.12%	78.00%	76.60%	64.20%	17.72%	70.19%	34.13%	81.91%	57.00%	**52.60%**	**46.72%**
GMNN	31.12%	68.82%	35.22%	78.63%	74.00%	65.60%	55.70%	13.11%	69.39%	33.93%	81.13%	48.00%	47.20%	43.80%
MAP	32.36%	50.88%	24.95%	53.70%	83.00%	72.00%	36.00%	24.33%	51.16%	19.10%	58.37%	62.00%	46.20%	42.80%
SGC	30.15%	68.85%	33.78%	77.83%	65.00%	58.00%	56.90%	11.58%	68.23%	35.38%	81.69%	47.00%	44.20%	38.30%
SIGN	32.29%	59.21%	34.41%	78.49%	70.00%	61.40%	57.80%	13.34%	68.60%	33.78%	80.89%	42.00%	43.60%	43.10%
SHGMNN-S	30.02%	59.94%	35.09%	78.65%	67.00%	63.00%	59.60%	14.08%	68.86%	34.42%	**82.39%**	42.00%	44.20%	44.70%
SHGMNN-M	31.01%	70.59%	41.67%	81.53%	85.00%	75.50%	61.70%	9.47%	62.29%	30.08%	81.67%	**65.00%**	37.40%	30.30%
SHGMNN	**33.94%**	**70.60%**	**42.29%**	**82.76%**	**86.00%**	**79.80%**	**63.30%**	**26.70%**	**71.04%**	**35.60%**	82.28%	52.00%	47.40%	41.80%

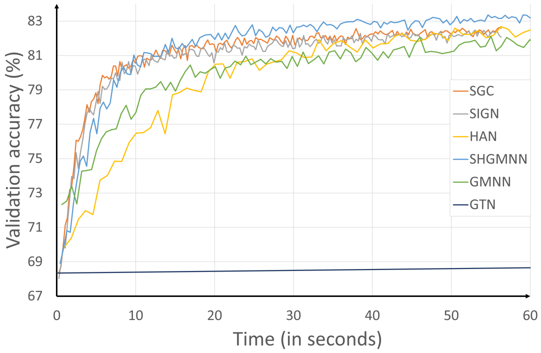


Figure 4: Convergence speed comparison of different GNN-based methods on CB-2021.

but also converge faster than **GMNN** and **HAN**. Meanwhile, its convergence speed is on a par with the algorithms designed specifically for scalable training, such as **SIGN** and **S-GCN**. From the figure we can tell that **GTN** takes significantly longer time for training since it is not equiped with any node space reduction process. Due to the two-layer attention structure, **HAN** also converges slower than other baselines. The underperformed results from both **GTN** and **HAN** demonstrate our model's improvement in scalability when dealing with heterogeneous information networks.

6.3 Case Study

To further illustrate the managerial insights from our SHGMNN model, we pick four representative startups from the test set of CB-2013 for case study. Specifically, we sort all startups in the test set according to their predicted scores, and pick one of the most representational startups from each of the following categories in confusion matrix: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) and *False Negative* (FN), as shown in Table 4.

Table 4: Selected startups for each category.

	Positive	**Negative**
Predicted (+)	*TP*: Gauss Surgical	*FP*: Official.fm
Predicted (-)	*FN*: Airbrite Inc.	*TN*: Camiloo, CellCap, Setgo

The **Gauss Surgical** is ranked NO.1 in our test set. Regarding its funding records, we can see an initial seed round of $805,000 dollars after 10 months of its establishment, followed by another seed round of over $1.4 million dollars. Up until now, this company has received a total of 7 funding rounds with in sum $51.5 million dollars. Undoubtedly, it is a startup with very high potential, and our model does its job well. Furthermore, by checking its founders, both CEO and CTO are graduates from the School of Medicine at Stanford University. It also manifests, from a different perspective, how the semantic meta-path SPS plays a vital role in identifying high-potential startups.

Another interesting case is **Airbrite Inc.**, ranked No.2 in our test set, but the ground truth label is indeed *negative*, given its fact that no *Series-A* funding before 2013 year-end snapshot date. However, this company received a second funding round of over $2 million dollars in two months after the snapshot date, and eventually got acquired by Indiegogo on May 10, 2016. Our model still foresees its great potential under the undesired circumstances of data limitations.

The **Camiloo**, which only received one seed-round investment in our CB-2013 dataset, along with its two direct neighbors, (**CellCap** and **Setgo**), do not receive further funding rounds until recently. Our model makes correct predictions for all these three startups. Last but not least, **Official.fm** lies at the bottom of our ordered startup list (below the bottom 0.5% percentile). According to the records, it did received *Series-A* round of investment (labeled as positive), but no further reported fundings up until current moment. This case indicates that our model learns knowledge beyond predicting further funding rounds, and could potentially see long-term performance of startups.

7 CONCLUSIONS

In this paper, we investigated the high-potential early-stage startup prediction problem which is of great economic value and managerial meanings. Specifically, we constructed the Heterogeneous Business Information Network (HBIN) and proposed the Scalable Heterogeneous Graph Markov Neural Network (SHGMNN) framework to identify high-potential startups. We adopted metapath count-based summations over different semantic metapaths and used a 1-d convolutional operator to leverage the relative weights, generating a parameterized summated adjacent graph structure that can be input of any GNNs. The GNN architecture we designed features at its lightweight linear diffusion structure, which is naturally ready for scalable training. To keep the label capacity which GNN ignores, we introduced a convex MAP inference over Hinge-Loss Markov Random Fields that can be optimized using fast parallel convex solvers. A variational EM framework is adopted to jointly optimize GNN and MAP inference. The proposed SHGMNN model provided a general solution to process large-scale heterogeneous information networks. Finally, extensive experimental results and case studies on two real-world datasets demonstrated SHGMNN's superiority over all other baselines for high-potential early-stage startup predictions.

REFERENCES

[1] Stephen Bach, Bert Huang, Ben London, and Lise Getoor. 2013. Hinge-loss Markov random fields: Convex inference for structured prediction. *arXiv preprint arXiv:1309.6813* (2013).

[2] Stephen H Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2015. Hinge-loss markov random fields and probabilistic soft logic. *arXiv preprint arXiv:1505.04406* (2015).

[3] Falco J Bargagli-Stoffi, Jan Niederreiter, and Massimo Riccaboni. 2020. Supervised learning for the prediction of firm dynamics. *arXiv preprint arXiv:2009.06413* (2020).

[4] Moreno Bonaventura, Valerio Ciotti, Pietro Panzarasa, Silvia Liverani, Lucas Lacasa, and Vito Latora. 2020. Predicting success in the worldwide start-up network. *Scientific Reports* 10, 1 (2020), 1–6.

[5] Diego Camelo Martines. 2019. Startup Success Prediction in the Dutch Startup Ecosystem. (2019).

[6] Jie Chen, Tengfei Ma, and Cao Xiao. 2018. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247* (2018).

[7] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. 2019. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 257–266.

[8] Jean-Michel Dalle, Matthijs Den Besten, and Carlo Menon. 2017. Using Crunchbase for economic and managerial research. (2017).

[9] Dominik Dellermann, Nikolaus Lipusch, Philipp Ebel, Karl Michael Popp, and Jan Marco Leimeister. 2017. Finding the unicorn: Predicting early stage startup success through a hybrid intelligence method. (2017).

[10] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 135–144.

[11] Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. 1999. Learning probabilistic relational models. In *IJCAI*, Vol. 99. 1300–1309.

[12] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*. 2331–2341.

[13] Samuel Gershman and Noah Goodman. 2014. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, Vol. 36.

[14] Lise Getoor, Nir Friedman, Daphne Koller, and Benjamin Taskar. 2001. Learning probabilistic models of relational structure. In *ICML*, Vol. 1. 170–177.

[15] Peter A Gloor, Pierre Dorsaz, Hauke Fuehres, and Manfred Vogel. 2013. Choosing the right friends–predicting success of startup entrepreneurs and innovators through their online social network structure. *International Journal of Organisational Design and Engineering* 3, 1 (2013), 67–85.

[16] Paul Gompers, Anna Kovner, and Josh Lerner. 2009. Specialization and success: Evidence from venture capital. *Journal of Economics & Management Strategy* 18, 3 (2009), 817–844.

[17] Paul Gompers and Josh Lerner. 2000. The determinants of corporate venture capital success: Organizational structure, incentives, and complementarities. In *Concentrated corporate ownership*. University of Chicago Press, 17–54.

[18] Beth Hadley, Peter A Gloor, Stephanie L Woerner, and Yuhong Zhou. 2018. Analyzing VC influence on startup success: A people-centric network theory approach. In *Collaborative Innovation Networks*. Springer, 3–14.

[19] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*. 2704–2710.

[20] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. 2018. Adaptive sampling towards fast graph representation learning. *arXiv preprint arXiv:1809.05343* (2018).

[21] Stanley Kok and Pedro Domingos. 2005. Learning the structure of Markov logic networks. In *Proceedings of the 22nd international conference on Machine learning*. 441–448.

[22] Amar Krishna, Ankit Agrawal, and Alok Choudhary. 2016. Predicting the outcome of startups: less failure, more success. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, 798–805.

[23] Miona Milosevic. 2018. Skills or networks? Success and fundraising determinants in a low performing venture capital market. *Research Policy* 47, 1 (2018), 49–60.

[24] Ramana Nanda, Sampsa Samila, and Olav Sorenson. 2020. The persistent effect of initial success: Evidence from venture capital. *Journal of Financial Economics* (2020).

[25] Radford M Neal and Geoffrey E Hinton. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*. Springer, 355–368.

[26] Jeannette Paschen. 2017. Choose wisely: Crowdfunding through the stages of the startup life cycle. *Business Horizons* 60, 2 (2017), 179–188.

[27] Meng Qu, Yoshua Bengio, and Jian Tang. 2019. Gmnn: Graph markov neural networks. *arXiv preprint arXiv:1905.06214* (2019).

[28] Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine learning* 62, 1-2 (2006), 107–136.

[29] Emanuele Rossi, Fabrizio Frasca, Ben Chamberlain, Davide Eynard, Michael Bronstein, and Federico Monti. 2020. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198* (2020).

[30] Aidin Salamzadeh and Hiroko Kawamorita Kesim. 2015. Startup companies: Life cycle and challenges. In *4th International conference on employment, education and entrepreneurship (EEE), Belgrade, Serbia*.

[31] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. Springer, 593–607.

[32] Boris Sharchilev, Michael Roizner, Andrey Rumyantsev, Denis Ozornin, Pavel Serdyukov, and Maarten de Rijke. 2018. Web-based startup success prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2283–2291.

[33] Cemre Ünal. 2019. *Searching for a Unicorn: A Machine Learning Approach Towards Startup Success Prediction.* Master's thesis. Humboldt-Universität zu Berlin.

[34] OECD/European Union. 2019. The Missing Entrepreneurs 2019: Policies for Inclusive Entrepreneurship. (2019). https://doi.org/10.1787/3ed84801-en

[35] Joosua Virtanen. 2019. Predicting high-growth firms with machine learning methods. (2019).

[36] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference*. 2022–2032.

[37] Yueyang Wang, Ziheng Duan, Binbing Liao, Fei Wu, and Yueting Zhuang. 2019. Heterogeneous attributed network embedding with graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 10061–10062.

[38] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.

[39] B Yankov. 2012. Overview of Success Prediction Models for New Ventures. In *International Conference Automatics and Informatics*, Vol. 12. 13–16.

[40] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. In *Advances in Neural Information Processing Systems*. 11983–11993.

[41] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. 2019. Graphsaint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931* (2019).

[42] Charles Zhang, Ethan Chan, and Adam Abdulhamid. 2015. Link prediction in bipartite venture capital investment networks. *CS224-w report, Stanford* (2015).

[43] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 793–803.

[44] Hao Zhong. 2019. *Venture capital investment: from rule of thumb to data science*. Ph.D. Dissertation. Rutgers University-Graduate School-Newark.

[45] Difan Zou, Ziniu Hu, Yewen Wang, Song Jiang, Yizhou Sun, and Quanquan Gu. 2019. Layer-dependent importance sampling for training deep and large graph convolutional networks. *arXiv preprint arXiv:1911.07323* (2019).

A APPENDIX

Evaluation Protocols To simulate a real-world scenario, the startups in our datasets are segmented according to their founded dates. In the 10-year period, we group the startups founded in the first 5 years into *training* set, the ones founded in the 6-th year into *validation* set, and those founded in the 7-8th years into *testing* set. (recall that we reserve 2 years as the performance window). As a result, in CB-2013, there are 2,413 nodes in the *training* set, 1,097 nodes in the *validation* set and 3,864 nodes in the *testing* set, while in CB-2021, there are 5,066 nodes in the *training* set, 719 nodes in the *validation* set and 956 nodes in the *test* set. Note that two datasets with different *training/validation/testing* distribution can help evaluate the robustness of our model.

We compare the results on two categories of evaluation metrics. The first is classification correctness, including accuracy, F1-score, AUC and AUPR. The other group is ranking metrics, including the Precision@10, @50, and @100. Note that the Precision@K is a popular evaluation metric used by business success prediction-related papers [4, 32].

Experimental Setup We use Metapath2Vec on the HBIN to generate the node attributes with parameters setting: window size = 7, walks per node = 1,000, walk length = 100, attribute dimension = 128. Two semantic metapaths are defined: SPS and SVS. We use the same node attributes for all the baselines and employ two metapaths: SPS and SVS. The same set of metapaths are incorporated for **Metapath2Vec** and **HAN**. For **GMNN**, **SIGN** and **SGC**, the input adjacent matrix is the mean average of two metapath count matrices to ensure unbiased comparisons. We implement our model using PyTorch (For GNN training) and Matlab (For MAP Inference). The hidden layer dimensions are set at 32. In the process of model training, we use the Adam optimizer for parameter optimization. We set learning rate at 0.01 and mini-batch size at 32. The parameters of the baselines are set up similarly as our method and carefully tuned to ensure fair comparisons.