Smart Roles: Inferring Professional Roles in Email Networks

Di Jin*
University of Michigan
dijin@umich.edu

Mark Heimann*
University of Michigan
mheimann@umich.edu

Tara Safavi
University of Michigan
tsafavi@umich.edu

Mengdi Wang
University of Pittsburgh
mew133@pitt.edu

Wei Lee, Lindsay Snider
Trove AI
{wei,lindsay}@trove.com

Danai Koutra
University of Michigan
dkoutra@umich.edu

ABSTRACT

Email is ubiquitous in the workplace. Naturally, machine learning models that make third-party email clients "smarter" can dramatically impact employees' productivity and efficiency. Motivated by this potential, we study the task of *professional role inference* from email data, which is crucial for email prioritization and contact recommendation systems. The central question we address is: Given limited data about employees, as is common in third-party email applications, can we infer *where* in the organizational hierarchy these employees belong based on their email behavior?

Toward our goal, in this paper we study professional role inference on a unique new email dataset comprising *billions* of email exchanges across thousands of organizations. Taking a network approach in which nodes are employees and edges represent email communication, we propose EMBER, or EMBedding Email-based Roles, which finds email-centric embeddings of network nodes to be used in professional role inference tasks. EMBER automatically captures behavioral similarity between employees in the email network, leading to embeddings that naturally distinguish employees of different hierarchical roles. EMBER often outperforms the state-of-the-art by 2-20% in role inference accuracy and 2.5-344× in speed. We also use EMBER with our unique dataset to study how inferred professional roles compare between organizations of different sizes and sectors, gaining new insights into organizational hierarchy.

ACM Reference Format:
Di Jin*, Mark Heimann*, Tara Safavi, Mengdi Wang, Wei Lee, Lindsay Snider, and Danai Koutra. 2019. Smart Roles: Inferring Professional Roles in Email Networks . In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3292500.3330735

1 INTRODUCTION

Email is indispensable and ubiquitous in the workplace. According to a 2018 Radicati Group study [32], despite the recent rise of team communication tools (e.g., Slack, Microsoft Teams), email is the most widespread tool for both internal and external business communication. By the end of 2018, over 281 billion emails were sent and received *per day*, and there were over 3.8 billion email users versus a few million active users of Slack [23]. Because email is so central to the workplace, email interactions contain a wealth of information and insights into professional relationships, organizational structure, and employee behavior.

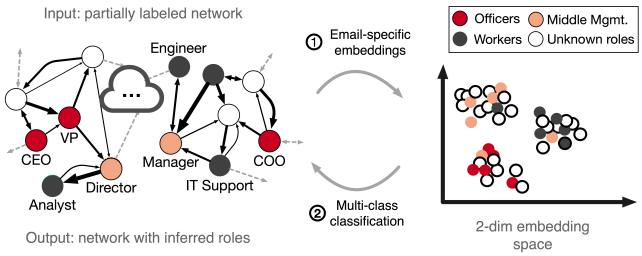


Figure 1: EMBER leverages communication volume and reciprocity to (1) compute email-specific structural embeddings, and then (2) infer professional roles via multi-class classification.

In this work, we study the problem of professional role inference, or inferring employees' roles in an organizational hierarchy, using a unique new dataset comprising *billions of emails across thousands* of organizations collected by Trove'semail-based application [1]. We motivate this task by the multitude of existing third-party email clients and applications that leverage emails to help recommend contacts, suggest responses, and organize and filter inboxes. While such applications typically have access to limited metadata about user emails, such as the sender and received time, they often *do not have complete information about the users themselves.* Therefore, inferring characteristics about users, such as their professional roles, can inform the personalization of "smart" email applications. For example, an email client that correctly identifies a C-suite executive (CEO, CTO, etc) can suggest to employees that the executive's emails be prioritized.

Our approach to professional role inference relies on the inherent network structure of an email corpus, wherein employees are nodes in the email graph and weighted, directed edges capture email exchanges. Importantly, to ensure a high level of user privacy, our email network is totally anonymized. It does not use any sensitive data from the email corpus, such as text, sent/received time, or subject line. Using this generalized email network, we build on recent advances in network representation learning, which have been shown to be state-of-the-art in difficult supervised learning tasks on networks. Specifically, we propose **EMBER**, or **Embedding Email-based Roles**. Our approach efficiently learns employee representations based on their email exchange behaviors, which are

* Authors contributed equally to this work.

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330735

captured in the *structure* of the email network, and predicts their professional roles via multi-class classification, as shown in Figure 1.

The contributions of this work include:

- **Email-centric embeddings**. We propose EMBER, a powerful and fast approach for embedding nodes, which correspond to employees in email networks, toward professional role inference. EMBER combines initial discoveries on our email corpora with the power of network representation learning.
- **Analysis and insights.** We show that EMBER is effective and efficient in inferring professional roles on several large-scale email corpora. We also apply EMBER on our new email corpus to gain insights into hierarchical differences across companies. By comparing the email behaviors of employees across organizations, we demonstrate that the role and function of employees is highly dependent on the size and sector of an organization.
- **Unique dataset**. We study professional role inference on a new email dataset collected by the Trove email application. Our dataset is unique and large-scale, comprising several *billion* email exchanges that span *multiple* organizations and sectors, unlikely previously analyzed email corpora. We use this characteristic to our advantage in our analyses and discoveries.

Next we discuss the work that is most relevant to ours.

2 RELATED WORK

Relevant areas of work include email network analysis, embeddings, and semi-supervised learning over networks. We qualitatively compare related methods that can be applied to the problem we consider in Table 1, and also compare them experimentally in § 5.

Email network analysis. User behaviors in email networks have been studied for modeling [3, 19, 34, 37], spam and fraud detection [2, 21, 33], and email ranking [39] purposes. Most works leverage textual features such as email addresses, body sentiment words, length of subjects [37], recipients [19], reply time, and email size [24] to characterize email behaviors. While analysis of this data is possible on a few email corpora made publicly available under special circumstances [10], such style of analysis would violate user privacy in the real-world scenarios that interest us. Therefore, we avoid methods that rely on textual features of email data.

Another direction involves the computation of network centralities. For example, Zhu et al. [39] propose Inner- and Outer-Pagerank centrality to distinguish nodes that mainly interact within and across communities. Aliabadi et al. [3] classifies professional roles based on graph centralities including in-/out-/total degree, clustering coefficient, PageRank, HITS, and betweenness. Some of these centralities are not node specific in that they do not naturally lend themselves to local computation. There are other works [28, 30] combining textual (e.g., mean response time) and network features (e.g., hubs, authorities, cliques). We compare such networked approaches to our own in § 5.

Network embedding. Node representation learning or embedding methods have grown significantly in popularity [12, 14]. Such methods intuitively try to embed similar nodes close in vector space. Most often similarity is defined in terms of node proximity in the network (i.e., LINE [31], DeepWalk [25], node2vec [13], and DNGR [5]). Unlike these, our proposed approach, EMBER, preserves *structural node similarity* instead of proximity. More related to our work are struc2vec [27] and GraphWave [7]. The former captures *structural* node similarity by degrees in local neighborhoods, although it assumes an unweighted and undirected graph and is not efficient on large datasets. The latter derives structural embeddings from heat wavelet diffusion patterns, but is also relatively inefficient on large networks in practice. However, they report improved results over RolX [18], a matrix factorization method for extracting structural role that uses hand-crafted features.

Table 1: Qualitative comparison of EMBER to alternatives. (1-2) Directionality & connection strength: Can the method handle directed and weighted edges? (3) Node specific: Can it embed only a subset of nodes? (4) Proximity independence: Is it independent of node proximity? (5) Scalable: Is it subquadratic in the number of nodes?

	Direction-ality	Conn. strength	Node specific	Prox. indep.	Scalable
SNA [3]	✓	✓	?	✗	✗
Rolx [18]	✓	✓	✗	✓	✓
LINE [31]	✓	✓	✗	✓	✓
node2vec [13]	✓	✓	✗	✗	✓
struc2vec [27]	✗	✗	✗	✓	✗
GraphWave [7]	✓	✓	✗	✓	?
DNGR [5]	✓	✓	✗	✗	✗
LinBP [11]	✗	✓	✗	✗	✓
EMBER	✓	✓	✓	✓	✓

Semi-supervised learning. Professional role inference in email networks (from a technical standpoint, multi-class classification) can also be modeled as semi-supervised learning [4, 40] or belief propagation [20, 38]. The key idea is to leverage not only labeled, but also unlabeled data, during the classification task. One related work from this domain is LinBP [11], a linearized version of belief propagation that can handle a mix of homophily and heterophily in multi-class settings. It should be noted, though, that such methods require explicitly specifying the amount of homophily between connected nodes, which may not be known in advance.

Recent methods based on graph neural networks [**?**] have demonstrated state-of-the-art performance for semi-supervised node classification on some datasets. However, their neighborhood aggregation mechanisms rely strongly on homophily, not necessary the a correct assumption for the task of role inference. In our application, their effect would be to propagate features among (and thus learn similar embeddings for) users within a company rather than users that have the same roles.

3 DATA

In this section, we introduce our datasets, discuss how we standardized and cleaned them, and give preliminary descriptive analyses that motivate our methodology in Section 4.

3.1 Email corpora

3.1.1 New email dataset. Our new dataset, collected by the Trove AI email application, consists of over 568 *million* emails in the year 2017 from ~130 000 users and their contacts. As far as we know, this is the first dataset studied in email network analysis that contains both *intra-* and *inter*-organization emails: exchanges between employees of the same company and exchanges between employees

Table 2: Overview of our datasets, consisting of sub-networks of Trove and Enron. We give the number of employees (nodes), connections (unweighted, undirected edges), email exchanges (weighted, directed edges), and the ground-truth distribution of roles (§ 3.2).

	Employees	Connections	Email exchanges	# Officers | Mid. mgmt | Workers
Trove-19	19	47	274	4 | 10 | 5
Trove-98	98	101	1769	53 | 32 | 13
Trove-141	141	1 242	9565	23 | 79 | 39
Trove-183	183	3136	21 655	16 | 133 | 34
Trove-318	318	1026	12 643	30 | 210 | 78
Trove-2K	2 414	16 281	183 443	495 | 1 300 | 620
Trove	9 989 507	40 290 044	568 678 419	495 | 1 300 | 620
Enron	75 416	319 935	2 064 442	31 | 44 | 41

of different companies, respectively. Per record, we retain only a datestamp and the anonymized sender and receiver IDs. We also collected ground-truth organizational roles by gathering email-to-organizational role mappings using an email signature parsing tool and information from a third-party data provider, with the consent of app users. This information is used only for evaluating EMBER.

We construct several weighted, directed email subnetworks from Trove's email corpus. In each network, each node is an employee and directed, weighted edges represent the number of emails from the sender to the receiver. We give some descriptive statistics of the following subnetworks in Table 2:

- Trove: All email exchanges between employees from several thousand companies during 2017.
- Trove-19, ..., Trove-318: Each of the five subnetworks captures the internal (intra-organization) emails during 2017 within one company. The number after the dash indicates the number of employees in the respective dataset.
- Trove-2K: All email exchanges between the employees of the five companies (Trove) and all their contacts (within and across organizations) in 2017.

3.1.2 Established email dataset. We also use the well-studied Enron email dataset. This dataset consists of email exchanges in 1999-2002 between the 116 Enron staff [15, 29] and their external contacts, for a total of 75 416 email users in the network. This is the only publicly available email corpus containing employee role information. The basic statistics of the Enron corpus are given in Table 2.

3.2 Professional roles

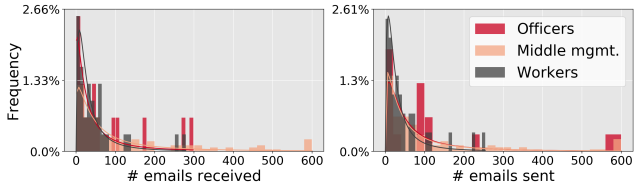
3.2.1 Standardization. While the categorization of professional roles may differ by organization and domain area, we follow established literature [6, 16] in organizational studies to define three hierarchical professional roles. We adopt the terminology of Cole and Bruch [6] in particular, and classify all employees as one of:

- **Officers**: These are "C-Suite" employees, meaning top-level officers such as CEO, COO, and other executives. We also grouped co-founders of organizations into this class.
- **Middle management**: These are middle-level managers responsible for coordinating the vision of officers by directing lower-level employees [6]. We included all non-officer employees with titles including "Manager" in this class.
- **Workers**: These are employees who directly contribute to the day-to-day work of the company. As to be expected, the titles in this category are more diverse, and include associates, assistants, engineers, salespeople, etc.

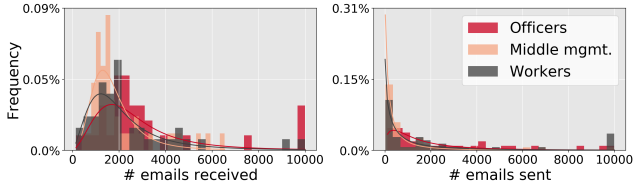
These well-established groupings delineate between *clearly distinguishable roles* (e.g., salesperson versus CEO) while avoiding *arbitrary distinctions* (e.g., project manager versus senior project manager), which differ between organizations and change over time. Inferring specific roles such as "engineer" and "sales agents" in the companies is left for future work.

To categorize each employee into a hierarchical role, we match each professional role to a set of manually curated keywords. We manually validate the categorizations due to the complexity of real-world job descriptions: for example, a "front office executive" is likely a "worker", not "officer". We categorized all employees in both the Trove and Enron datasets. If an employee's role changed during the period of time that is captured in the email network representation, we use her latest role as ground-truth. We give the distribution of professional roles per dataset in Table 2. The large size of the management class may be due to job title inflation an a high degree of delegation of authority in the workplace.

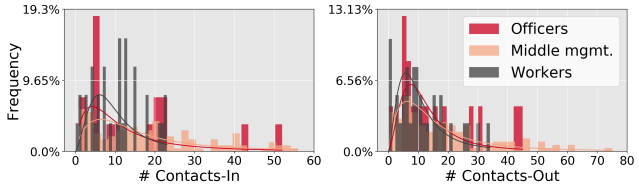
3.2.2 Preliminary analysis. We now present some preliminary analysis of email patterns for different professional roles, which motivated the design of our method, EMBER. Since we follow privacy standards of third-party email applications and do not use any textual or other information beyond the email network structure, we focus on *features that are encoded in the structure of the email network*: the number of emails that each employee sends and receives (i.e., edge weight), and the number of their contacts or the people that they exchange emails with (i.e., number of unweighted edges). Figures 2a-b correspond to the distributions of emails received and sent across different roles in Trove-183 and Enron. For Trove-183



(a) Trove-183: Emails received and sent by professional role.



(b) Enron: Emails received and sent by professional role.



(c) Trove-183: Contacts by professional role.

Figure 2: The distributions of communication volume and contacts by professional role demonstrate differences in email behaviors.

Table 3: Major symbols and their definitions.

Symbol	Definition
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Graph with nodes $\mathcal{V}$ and edges $\mathcal{E}$, where edge (u, v) has weight w_{uv}
$\mathcal{U} \subseteq \mathcal{V}$	Set of users to embed and learn professional roles for
$\mathcal{N}_u^{k+}$, $\mathcal{N}_u^{k-}$	k-step in-/out-neighborhoods of employee u, resp.
$\mathcal{P}_{u \to v}^{k+}$	k-step directed path from u to v (i.e., ordered edge set)
δ	Discount factor for step distance
K	Maximum step distance to consider
$\mathbf{b}_u = [\mathbf{b}_u^+ \ \mathbf{b}_u^-]$	Concatenated ingoing and outgoing structural behavior histograms for employee u
p	Embedding dimensionality, or the # of landmarks
$\tilde{\mathbf{Y}}$	Embedding of users in $\mathcal{G}$

we also give the distribution of the number of contacts (i.e., people who communicate with an employee) in Fig. 2c. The distributions for other companies are similar.

We observe that the distributions of email activity and contacts exhibit *some* differences per professional role. For example, officers (CEOs, CTOs, etc.) tend to receive more emails than lower-ranking employees, and tend to also have more contacts that they reach out to. Nevertheless, based on these email behaviors alone, there are no *clearly* distinguishable patterns that can be effectively leveraged on their own for role inference. Therefore, we will design a nuanced method that also captures *more complex relations* in email activity over the network in order to accurately predict professional roles.

4 METHODOLOGY

Our proposed method, EMBER, is motivated by our observation that the distributions of emails received and sent by different professional roles exhibit some observable patterns (Figure 2), although not enough to solve the role inference task alone (§ 5.2). Since the volume of email exchanges is captured by the weighted in- and out-degree in the email network $\mathcal{G}$, these will be central in the design of EMBER, the steps of which are:

- **S1** Capturing local network structure, such as volume of sent and received emails, around each employee (§ 4.2),
- **S2** Learning embeddings that preserve employee similarity based on this local structure (§ 4.3), and
- **S3** Role inference via multi-class classification (§ 4.4).

In this section, we describe each step in detail, and conclude with the asymptotic complexity of EMBER in § 4.5.

4.1 Preliminaries

Here we briefly overview important notation, also given in Table 3 for reference. Our email network is a weighted, directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the graph's nodes $\mathcal{V}$ represent employees or, more generally, users of the email client in question, and the edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ corresponds to directed communications between employees. An edge weight w_{uv} captures the number of emails employee u has sent employee v, and vice versa for w_{vu}. Let $\mathcal{U} \subseteq \mathcal{V}$ be the subset of nodes (employees) in $\mathcal{V}$ for whom we want to infer roles (those for whom we do not have ground-truth labels).

Next, we define directed neighborhoods in the email network. Given a node u, let $\mathcal{N}_u^{k+}$ be u's k-step out-neighborhood, or the employees that can be reached in k directed steps of email communication from u. For example, u's out-neighborhood for $k = 1$ are all the employees whom u has emailed directly. Likewise, let $\mathcal{N}_u^{k-}$ be u's k-step in-neighborhood, or the employees from which u is reachable by a directed path of k edges. We give an illustrative example of directed neighborhoods in Figure 3, where employee u's 2-step out-neighborhood $\mathcal{N}_u^{2+}$ consists of employee v.

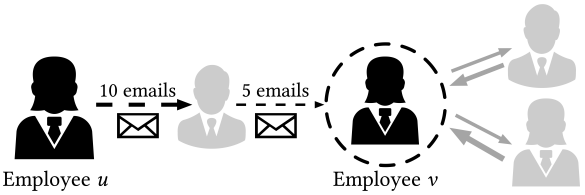


Figure 3: Illustrative example of structure in email networks. Employee u's 2-step out-neighborhood $\mathcal{N}_u^{2+}$ consists of employee v, and the weight of the path (§ 4.2.1) from u to v is $10 * 5 = 50$.

Finally, let $\mathcal{P}_{u \to v}^{k+}$ be a directed k-step shortest path from node u to $v \in \mathcal{N}_u^{k+}$. In Figure 3, the path $\mathcal{P}_{u \to v}^{2+}$ consists of two edges: one from u to the intermediary gray employee, and one from the intermediary employee to v. Incoming paths are similarly defined.

4.2 Structural behavior in email networks

As shown in Figure 2, an employee's local structure in an email network—how many emails she sends and receives, how many people she contacts—is relatively indicative of her professional role, although capturing these statistics alone is not sufficient for role inference (§ 5.2). Our goal in this section is to mathematically capture this local structure around each employee (step **S1**), with the ultimate goal of later obtaining embeddings that preserve the similarity between employees with similar local structure.

Intuitively, a model of an employee's "structural behavior" in an email network should capture "what the local neighborhood around each employee looks like", since employees at similar levels in the organizational hierarchy often have similar neighborhood structures: for example, a CEO is likely connected to many well-connected employees. Moreover, because our application focus is *email*, our model of "structural behavior" must account for the real-world differences between *sending* emails and *receiving* them, and for how *many* emails users send and receive. We will show in § 5 that these distinctions are important in professional role inference.

4.2.1 Capturing active communication. Intuitively, an important part of characterizing the neighborhood of each employee u in an email network is identifying the employees with whom u actively communicates (versus "less-important" employees with whom u has only communicated a few times, for example). Given employee u's k-step in and out neighborhoods $\mathcal{N}_u^{k+}$ and $\mathcal{N}_u^{k-}$ (§ 4.1), we propose to capture this intuition by *weighting paths* between u and her (in/out) neighbors. These path weights will be used in our final definition of structural behavior, when we formulate a unified version of "what the neighborhood around u looks like" (§ 4.2.2).

We define the weight of an outgoing k-step path $\mathcal{P}_{u \to v}$ as the product of all edge weights in the path, i.e.,

$$\texttt{path\_weight}\left(\mathcal{P}_{u \to v}^{k+}\right) = \prod_{(i,j) \in \mathcal{P}_{u \to v}^{k+}} w_{ij}, \tag{1}$$

There are other ways to define path weights, for example with summations instead of products, but this is not essential to our work and we find empirically that products work well. Going back

to our simple example in Figure 3, the path weight from employee u to employee v is $10 * 5 = 50$. Note that it is not the *exact* value of the path weight, but rather the relative values of path weights as compared to each other, that will be important.

4.2.2 Structural behavior histograms. As a reminder, our ultimate goal is to define a mathematical notion of "structural behavior" that captures the local structure surrounding each employee in the email network, where local structure includes edge directionality (received/sent emails) and weights (volume of communication). We propose to do this by creating a weighted histogram (i.e., a vector of counts) per employee u that captures *what the neighborhood around u looks like*, using the previously defined path weights, as well as the degrees of u's neighbors, which themselves capture how well-connected those neighbors are.

Let $\mathbf{b}_u^{k+}$ ($\mathbf{b}_u^{k-}$) be employee u's outgoing (incoming) *structural behavior vector* in her k-step neighborhood. Each entry of this vector, or histogram, captures the employees in u's k-step neighborhood of a certain level of connectedness (i.e., of degree d), and also incorporates the weight of the path from u to each employee of that degree, which can be seen as the importance of those employees to u. Here, we use a logarithmic grouping scheme to group larger ranges of high-degree employees together, to reflect the skewed (power law) distribution of communication commonly observed in real-world social and information networks.

Let D_u^{k+} be the set of employees in u's k-step out-neighborhood with degree d. In other words, $D_u^{k+} = \{v \in \mathcal{N}_u^{k+} \mid \lfloor \log_2(\deg(v)) \rfloor = d\}$. Then, we define the d-th entry of u's outgoing structural behavior histogram at k steps as

$$b_{u,d}^{k+} = \sum_{v \in D_u^{k+}} \texttt{path\_weight}\left(\mathcal{P}_{u \to v}^{k+}\right), \tag{2}$$

with incoming structural behavior at k steps defined similarly.

4.2.3 Putting it all together. In our setting, practitioners may want to capture local structure for each employee in the email network *across* different distances k. Therefore, we propose a formulation to this end that captures the diminishing importance at higher step distances k (i.e., at employees not as closely connected to u in the email network). As such, given a maximum step distance K (limited by the diameter of the email network), we define the overall outgoing structural behavior $\mathbf{b}_u^+$—note the absence of the k superscript here, which distinguishes from the definitions in § 4.2.2—as a linear combination of k-step structural behaviors $\mathbf{b}_u^{k+}$:

$$\mathbf{b}_u^+ = \textstyle\sum_{k=0}^{K} \delta^k \, \mathbf{b}_u^{k+}, \tag{3}$$

where δ^k is a "discount factor" to capture the diminishing importance of higher step distances. As with all previously described equations, the incoming behavior histogram $\mathbf{b}_u^{k-}$ is constructed similarly. Finally, to unify the incoming and outgoing histograms, which will allow us to obtain embeddings as discussed in the next section, we simply concatenate the in- and out-histograms to obtain the final structural behavior vector for employee u as $\mathbf{b}_u = [\mathbf{b}_u^+, \mathbf{b}_u^-]$.

4.3 From structural behavior to embeddings

So far we have constructed per-employee structural behavior histograms $\mathbf{b}_u$ by following incoming and outgoing paths. Our next goal is to use these histograms to obtain latent features via *embeddings*, which we will show in § 5 are powerful tools for professional role inference. As it has been shown that many existing embedding methods implicitly or explicitly factorize a node-to-node similarity matrix $\mathbf{S}$, whose construction varies by method [26], we take advantage of this connection and turn to fast and theoretically-sound *implicit* matrix factorization for a scalable approach (step **S2**).

To distinguish the conceptual differences between explicit and implicit matrix factorization for node embedding, consider that in the *explicit* matrix factorization approach, we would need to construct and factorize an employee-to-employee similarity matrix $\mathbf{S}$ that captures the similarity between employees' structural behavior histograms $\mathbf{b}_u$. But instead of *exactly* constructing the full matrix $\mathbf{S}$, which is quadratic in the number of nodes to embed, and learning an *approximate* factorization of $\mathbf{S}$, we utilize a low-rank *approximation* of $\mathbf{S}$ that *never* has to be computed, because its decomposition has a known, *exact* factorization. Here, we adapt a technique used for embedding-based network alignment in [17] to our setting:

Theorem 4.1 (Adapted from [17]). *Given a network $\mathcal{G}$ with a $|\mathcal{V}| \times |\mathcal{V}|$ structural similarity matrix $\mathbf{S} \approx \mathbf{Y}\mathbf{Z}^T$, its node embedding matrix $\mathbf{Y}$ can be approximated as $\tilde{\mathbf{Y}} = \mathbf{C}\mathbf{U}\mathbf{\Sigma}^{1/2}$, where $\mathbf{C}$ is the matrix of similarities between the $|\mathcal{V}|$ nodes and p landmark nodes [8], and $\mathbf{M}^\dagger = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is the SVD of the pseudoinverse of the $p \times p$ landmark-to-landmark similarity matrix $\mathbf{M}$.*

The key takeaway is that we select a small number p of employees called *landmarks*, and compare the employees for whom we want to learn professional roles *against the landmarks*. Let us assume that we want to infer the roles for all the employees $\mathcal{V}$ in the email network. Therefore, to obtain structural embeddings via the technique above, we only need to perform a *small fraction of employee-to-employee comparisons* $|\mathcal{V}| \times p$ stored in $\mathbf{C}$ ($p \ll |\mathcal{V}|$), and a few "expensive"' computations on the *small* $p \times p$ (sub)matrix $\mathbf{M}$.

Now, it is only left for us to discuss: (1) how we compute structural similarity between two employees' structural behavior histograms $\mathbf{b}_u, \mathbf{b}_v$; (2) how we select the landmarks; and (3) how we embed only a set of employees of interest, which makes EMBER even *more* scalable than the technique described in Theorem 4.1.

4.3.1 Structural user similarity. We define the similarity between two employees u and v based on their structural email behaviors as $sim(u, v) = e^{-||\mathbf{b}_u - \mathbf{b}_v||}$, where $||\cdot||$ is a vector norm, for example Euclidean distance. Recall that our setting assumes no additional side information for employees beyond their behavior in the email network for privacy reasons. However, if any side information is available, the similarity thereof between two employees may also be incorporated into the similarity function [17].

4.3.2 Landmark selection. In EMBER the number of landmark employees p determines the dimensionality of the generated embeddings. The landmark employees, used for the construction of the "thin" $\mathbf{C}$ similarity matrix, can be sampled uniformly at random [35] or according to more sophisticated matrix-theoretic methods [22]. Domain-specific heuristics, such as sampling employees with probability proportional to their degrees, are fast to compute and lead to more competitive and stable classification accuracy than random selection (see supplementary material B). Intuitively, since the

Algorithm 1 EMBER: EMBedding Email-based Roles

Input: Email network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, employees of interest $\mathcal{U} \subseteq \mathcal{V}$, maximum step K, discount factor $\delta \in (0, 1]$
Output: Professional roles for the employees of interest $\mathcal{U}$

S1: Capture structural behavior in email network

Outgoing / incoming structural behavior histograms $\mathbf{b}_u^{\pm} \leftarrow 0$
for $k = 1 \ldots K$ **do**
  Construct k-step outgoing / incoming histograms $\mathbf{b}_u^{k\pm}$ ▷ Eq. 2
  Update outgoing / incoming histograms to $\mathbf{b}_u^{\pm} \leftarrow \mathbf{b}_u^{\pm} + \delta^k \mathbf{b}_u^{\pm}$
end for
Concatenate final histograms into $\mathbf{b}_u \leftarrow [\mathbf{b}_u^{+}, \mathbf{b}_u^{-}]$

S2: Embed employees in email network

Select set of p landmark employees ▷ § 4.3.2
Compute $\mathbf{C}$ as $|\mathcal{V}| \times p$ similarity matrix of behavior histograms $\mathbf{b}_u$, $\mathbf{b}_v$
Compute the SVD of the pseudoinverse of the small submatrix $\mathbf{M}$ of $\mathbf{C}$
Obtain embeddings $\tilde{\mathbf{Y}} \leftarrow \mathbf{C}\mathbf{U}\Sigma^{1/2}$ ▷ Theorem 4.1

S3: Professional role inference

Learn a classifier with embeddings $\tilde{\mathbf{Y}}$ and the known roles

embeddings preserve similarity with respect to the landmarks, to capture diverse structural behavior in the embeddings it is advantageous to ensure structural diversity in the landmarks.

4.3.3 User subset embedding. In our application, an email client might only need to infer the organizational roles of a small subset of employees of interest $\mathcal{U} \subseteq \mathcal{V}$, where $|\mathcal{U}| \ll |\mathcal{V}|$. As the embedding computations in Theorem. 4.1 involve direct comparison only to landmarks, we can embed *any* subset of employees, as opposed to the entire network, which makes EMBER unique among representation learning techniques. Specifically, $\mathbf{C}$ can easily be adapted to be a $|\mathcal{U}| \times p$ matrix that holds the user-to-landmark similarities only for *employees of interest*.

4.4 Professional role classification

Given the embeddings from § 4.3, we infer organizational roles via multi-class classification (step **S3**). We assume that the organizational roles of some employees are known, and predict the roles of the remaining employees using supervised machine learning techniques on their embeddings. We give more details on the task setup in Sec. 5.2. The overview of EMBER is given in Algorithm 1. A more detailed version is given in Appendix A.

4.5 Complexity of EMBER

Here, we analyze the complexity of EMBER steps **S1** and **S2**, since **S3** can be implemented with well-studied supervised machine learning methods. Recall that scalability is an important requirement of our approach, since our task is motivated by the prevalence of third-party email applications that handle large amounts of data.

Assuming that we are obtaining embeddings for $|\mathcal{U}|$ employees, step **S1** of EMBER is $O(|\mathcal{U}|Kd_{avg}^2 + |\mathcal{U}|p \log_2 D_{max})$. Here, d_{avg} is the maximum between the average user in-degree and average user out-degree in the email network. In the second term, the factor of $\log_2 D_{max}$ in the second term comes from logarithmic binning (§ 4.2.2), with D_{max} is the maximum total degree in the graph and p being the number of landmarks (§ 4.3). Step **S2** requires $O(p^3)$ time to compute the pseudoinverse of the $p \times p$ similarity matrix $\mathbf{M}$, and then $O(|\mathcal{U}|p^2)$ time to left multiply it by $\mathbf{C}$. Since $p << |\mathcal{U}|$, the total time complexity for this step is $O(|\mathcal{U}|p^2)$. For large-scale problems, p, d_{avg}, and K are all asymptotically much smaller than $|\mathcal{U}|$, meaning that EMBER runs in time subquadratic to $|\mathcal{U}|$.

5 ANALYSIS AND INSIGHTS

In this section we present analysis and insights by putting EMBER into practice. Our main research questions are:

- **Q1** How does EMBER compare to the state-of-the-art in professional role inference?
- **Q2** How efficiently can EMBER infer professional roles?
- **Q3** Do roles across organizations of different sizes and sectors compare? What insights can we gain from role correspondences across organizations?

We ran all our analyses on a machine with a 6-core 3.50GHz Intel Xeon CPU and 256GB memory. For reproducibility, the source code is available at https://github.com/GemsLab/EMBER.

5.1 Background and setup

Here we briefly describe how we set up our experiments, including variants of EMBER we studied, baselines to which we compared EMBER, and choices of parameters for all methods compared.

5.1.1 EMBER variants. One of the main hypotheses of this work is that capturing email-specific behavior via sent/received emails and the volume of communication in the network is important in professional role inference. To test this hypothesis, we conduct our role inference experiments with three variants of EMBER beyond the one proposed in § 4: **EMBER-U** operates on unweighted, undirected graphs; **EMBER-D** only uses edge directions; and **EMBER-W** only considers edge weights. We run all variants of EMBER with maximum step distance $K = 2$ and discount parameter $\delta = 0.1$. We select the p landmark nodes with probability proportional to their degrees, following our analysis in Appendix B.

5.1.2 Baselines. Professional role inference can be approached with a variety of techniques. In our evaluation we consider *nine* baselines spanning well-known social network analysis, unsupervised and semi-supervised learning, and network embedding techniques. From the *non-embedding* literature, we compare to:

(1) SNA or Social Network Analysis [3] classifies roles based on graph statistics including degree, clustering coefficient, PageRank, HITS, and betweenness. To make the computation on the two largest networks (Enron and Trove) feasible, we estimate the betweenness centrality by sampling 1 000 users.
(2) RolX [18] is an *unsupervised* method that automatically infers structural roles via non-negative matrix factorization. We use the default settings provided in the paper.
(3) LinBP [11] is a belief propagation approach that leverages both the input labels and the network structure for classification. As input it requires a matrix of potentials $\mathbf{H}$, which defines the *homophily* between the different professional roles. We set it to [.45 .35 .2; .25 .5 .25; .25 .3 .35] based on the frequency of interactions between officers, middle managers, and workers in Trove-2K.

The *embedding* methods that we compare to are:

Table 4: Performance (AUC) of role inference across datasets and methods. "—" means that the method failed to finish within our time limit (12 hrs). EMBER and its variants prove strong in the role inference task. Moreover, EMBER outperforms its unweighted/undirected variants, demonstrating the importance of accounting for the volume and reciprocity of email exchanges in role inference. The asterisk, *, denotes statistically significant improvement over the best baseline at $p < 0.05$ in a two-sided t-test.

	SNA	RolX	LinBP	LINE	DeepWalk	node2vec	struc2vec	DNGR	Graphwave	EMBER-U	EMBER-D	EMBER-W	**EMBER**
Trove-318	.7605	.5670	.6908	.6618	.7602	.7648	.7799	.7131	.7685	.7749	.7563	.7625	**.8045***
Trove-183	.7648	.5787	.7718	.5657	.8071	.8223	**.8264**	4925	.6391	.7986	.7838	.8186	.8241
Trove-141	.6738	.5591	.7409	.7102	.7191	.7474	.7391	.6235	.7112	.7291	.7309	.6971	**.7568***
Trove-98	.6676	.5177	.6323	.6872	.5587	.6198	.6498	.5329	**.7177***	.6040	.5857	.6333	.6911
Trove-19	.5429	.6981	.6248	.7184	.5531	.5959	.6102	.6089	.7157	.6837	.7204	.6939	**.7337***
Trove-2K	.6305	.5212	.6622	.6771	.6769	.6780	**.6802**	.6527	.6594	.6689	.6345	.6677	.6745
Trove	.6633	.5280	5454	—	.6866	.6951	—	—	—	.6905	.7141	.7122	**.7162***
Enron	.6205	.5197	.5000	.6931	.7201	.7389	—	.5709	—	**.7393**	.7347	.7305	.7305

(4) LINE [13] We use 2nd-LINE to incorporate 2-order proximity and set other parameters to the provided defaults.
(5) DeepWalk [25] is a proximity-based embedding method that obtains node context via random walks.
(6) Node2vec [13] is a generalization of DeepWalk that strikes a balance between homophily and structural equivalence. We set $p = 1$ and $q = 100$ to put more emphasis on structural equivalence, as other settings resulted in worse performance.
(7) DNGR [5] uses a deep neural network on the positive pointwise mutual information matrix to embed weighted graphs. We use a 3-layer neural network model and set the random surfing probability $\alpha = 0.98$, as recommended in the paper.
(8) Struc2vec [27] is an embedding method that preserves structural similarity, unlike the previous approaches. It is the most related to EMBER and RolX. We keep the default settings stated by the authors with all 3 optimizations.
(9) GraphWave [7] computes structural embeddings based on heat wavelet diffusion. To evaluate the characteristic functions we use $\tau = d$ timepoints (equal to the dimensionality), and the default values for all the other parameters.

For all the embedding methods, including ours, we follow the literature by setting the dimension $d = 128$ for the email networks with more than 128 employees. Note that in the case of EMBER, the number of landmarks p corresponds to the embedding dimensionality d. For the smaller networks `Trove-98` and `Trove-19`, we set dimension $d = 64$ and $d = 16$, respectively.

5.2 Predicting professional roles with EMBER

In this section, we address question **Q1**, the key application and driver of our work: How accurately can EMBER infer employees' professional roles from email network data?

5.2.1 Methodology. As discussed in § 3.2, we cast the professional role inference problem as a multi-class classification task with three roles: *officers*, *middle management*, and *workers*. We evaluate all methods using the ground truth organizational roles per dataset (Table 2). For all the *supervised methods*, we feed the generated node representations (hand-crafted features for SNA, and embeddings learned from the rest) as inputs to the classifier. Our classification model is a one-vs-all SVM with linear kernel (penalty C=1, 10^6 iterations, and 10^{-6} tolerance); other models yielded similar results.

We perform 5-fold cross-validation across methods and datasets, and report the average (across folds) micro-AUC over all classes. For LinBP, which is *semi-supervised*, to imitate the 5-fold CV setting for the supervised methods, we select 80% of employees with ground truth to construct the explicit beliefs matrix $\mathbf{E}$—i.e., the known employee roles. LinBP then directly assigns a class to each user based on her maximum final belief. For RolX, which is an *unsupervised method*, we report the accuracy of the best match between the identified (structural) roles and the ground truth classes. Table 4 presents the micro-AUC results.

5.2.2 Findings. We immediately observe from Table 4 that while professional role inference is challenging, EMBER is clearly well-suited to the task, justifying our email-centric embedding approach over more generic techniques. Indeed, the email-centric design of EMBER leads to a statistically significant improvement over other methods on most datasets, by an average of 2-20%. In the cases where EMBER is not the highest performer, it comes in a close second, not even by a statistically significant margin.The good performance of EMBER is expected, as it is tailored to email networks and captures rich structural information therein. It should be noted that DNGR and GraphWave failed to finish within our time limit (12hrs) on `Trove` and `Enron` (Table 5). LINE and struc2vec failed to finish on `Trove`.

Importantly, we find that for all networks other than `Enron`, EMBER performs best *when using both edge connection strengths and directionality*. This confirms our initial hypotheses that the volume and reciprocity of email activity both characterize behaviors, which in turn distinguish professional roles, and justifies our use of such characteristics in the design of EMBER. That said, the `Enron` dataset is an exception. Here, both edge weights and directionality lead to marginal (< 1%) *decreases* in EMBER's accuracy. We hypothesize that this may be due to diverse, erratic email exchange behavior during the company's fraud crisis, which has been well-documented in the media and literature [36].

5.3 Efficiency of inference with EMBER

We now turn to question **Q2**: How fast is EMBER? Recall that our initial problem is motivated by the prevalence of third-party email applications that can benefit from role inference over email networks. Therefore, here we investigate whether EMBER is scalable enough to be practical in real-world scenarios.

Table 5: Average runtime in seconds, capped at 12h. While RoLX is faster for the smaller datasets, EMBER proves uniquely scalable on the Trove and Enron networks, which have up to *millions* of edges.

	Trove-318	Trove-2K	Trove	Enron
SNA	6.32	16.45	3193.26	333.33
RolX	**0.14**	**0.16**	2150.53	205.92
LinBP	0.54	2.88	14607.44	1038.09
LINE	171.95	153.12	>12h	267.48
DeepWalk	3.12	21.59	2464.13	255.84
node2vec	2.85	24.55	3484.05	254.60
struc2vec	17.48	188.65	>12h	29286.38
DNGR	21.05	72.83	>12h	>12h
Graphwave	2.73	5.66	>12h	>12h
EMBER	2.50	16.87	**830.80**	**84.98**

5.3.1 Methodology. We measured the time required to obtain the roles of employees in email networks of different size in the previously discussed role inference task. In Table 5 we report the average runtime in seconds across the 5 folds, and the average across 5 runs of the unsupervised RolX method.

5.3.2 Findings. We find that EMBER proves uniquely scalable for the large-scale Trove and Enron datasets, being 2.5 – 344× faster than all other methods that complete. This is especially true for the representation learning approaches that are most competitive with EMBER. Indeed, EMBER is over 4× faster than node2vec, and 508× faster than DNGR and GraphWave, based on their (incomplete) runtime of over 12 hours. This is not surprising given that EMBER relies on *implicit* factorization and can embed a given subset of nodes (§ 4.3). As a representative example, on the Trove network, which has over *40 million edges*, EMBER needs less than *14 minutes* to infer professional roles. EMBER is thus highly scalable, making it a practical candidate for real-world analysis of organizational communication, and for third-party email clients that recommend contacts and help prioritize emails.

5.4 Comparing professional roles with EMBER

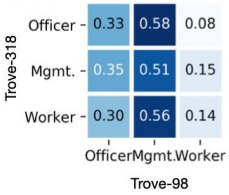
Finally, we address question **Q3**. Given the embeddings we learn with EMBER, we turn to a more qualitative study than our core role inference task. Here, we investigate whether we can use the EMBER embeddings to compare professional roles across organizations. This task is motivated by the unique nature of our Trove dataset, which comprises emails from many organizations of different sizes and sectors. We provide further results of this study in Appendix C.

5.4.1 Methodology. For the questions we asked in this study specifically, we used both the Trove-2K dataset studied in the previous sections as well as an *academia-specific* dataset collected from a university that collaborates with some of the companies in the Trove dataset. For reference, the academic email network consists of 3 078 users and 231 470 email exchanges.

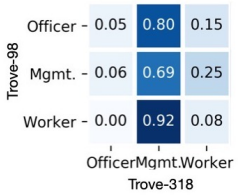
Our first step is to use EMBER to infer the roles for all employees in the Trove-2K network and the academic-specific network. Then, for all pairs of employees, we compute the ℓ_2 norm of the differences between the respective embeddings. We say that employee u at organization A "maps" to employee v at organization B if the ℓ_2 distance between u and v is minimal for all employees compared to u in B: $v = \arg\min_{j \in B} ||\mathbf{y}_u - \mathbf{y}_j||_2$, where $\mathbf{y}_u$ and $\mathbf{y}_j$ correspond to EMBER embeddings of employees u and j, respectively. In Figures 4a-4b, we show mappings of *officers*, *middle managers*, and *workers* across Trove-318 and Trove-98. The darker the color in the heatmaps, the more frequent is the corresponding employee mapping between the companies.

5.4.2 Findings. Interestingly, most employees at the bigger company (Trove-318) map to *high-ranking* positions at the smaller company (Trove-98), whereas most employees at Trove-98 map to *lower*-ranking positions at Trove-318. One potential explanation is that employees in larger companies may be more well-connected, in and outside of their own companies, and thus appear "higher-ranking" as compared to less well-connected employees at smaller companies. We also observe that middle management roles are similar to *all other* roles across companies, which may be because managers take on many fluid roles in the workplace, from core leadership to more basic day-to-day activities. We see similar patterns across all pairs of companies in the Trove dataset.

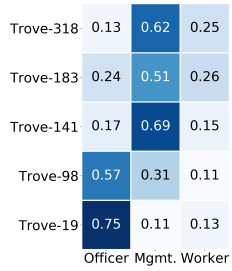
Using the academia email network, we also evaluate the similarity between academic roles and industry roles. Here we compare "professors" and "graduate students" to officers, middle management, and workers across the five companies in Trove. We find that professors are indeed similar to CEOs of smaller companies (Trove-98 and Trove-19), and more like managers in bigger companies (Trove-318 through Trove-141). We find this result fairly intuitive, given the day-to-day roles of university professors,



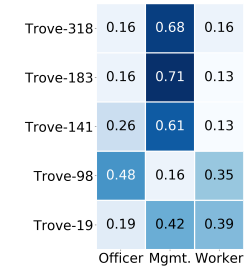
(a) Mapping similar employee behaviors from Trove-318 to Trove-98.



(b) Mapping similar employee behaviors from Trove-98 to Trove-318.



(c) Mapping professors to industry roles.



(d) Mapping graduate students to industry roles.

Figure 4: Mapping roles across companies and sectors. (a) and (b) indicate that employees in the bigger company Trove-318 are similar to positions at and above "management" in the smaller company Trove-98, and employees in Trove-98 are similar to positions at and below "management" in Trove-318. (c) and (d) show how similar "Professors" and "Graduate Students" are to job titles in different-sized companies: professors become more "important" in smaller companies (mapping to officers), while students are more similar to the management (or other positions) across companies.

who usually manage a (relatively small) group of students and staff, similar to higher-ranking employees in small companies and middle-ranking employees in large companies. Likewise, we find that graduate students are more like lower-level employees in small companies, suggesting that academic roles have some hierarchical equivalence with industry roles, and especially so in startups.

Our analysis has shown that the email-based behaviors of employees are indeed related to the size of the organizations for which they work. Therefore, changes in these role correspondences may inform company dynamics. For example, they may imply ongoing structural shifts which need to be addressed via reorganization [9].

6 CONCLUSION

Motivated by the prevalence of email in the workplace and the myriad of third-party email applications that could benefit from inferring characteristics about users, in this paper we study professional role inference in email networks. We introduce EMBER, which infers roles by leveraging embeddings learned from the *structural behavior* of employees in the network. We also study a new dataset with both *intra-* and *inter-*organization email exchanges, which enables our unique and extensive experiments and analyses.

There are many possibilities for future directions based on the results of this study. For one, email networks are inherently dynamic, and employees assume different professional roles or transition to new organizations. Therefore, a promising future direction involves extending EMBER's strengths to time-evolving networks. Moreover, although email is the most prevalent form of communication in the workplace, incorporating other sources of communication (Slack, Microsoft Teams) may lead to new insights, although it is challenging to obtain such data and combine different sources due to privacy reasons. Overall, as email networks contain a wealth of data, we believe that future analyses thereof may prove extremely useful to email clients, organizations, and employees alike.

ACKNOWLEDGEMENTS

This material is based upon work supported by the Trove AI, National Science Foundation under Grant No. IIS 1845491, an Adobe Digital Experience research faculty award, and an Amazon faculty award. 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] Trove AI. 2019. Your Personal Assistant for Networking. https://trove.com/.
[2] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph Based Anomaly Detection and Description: A Survey. *Data Min. Knowl. Discov.* 29, 3 (2015), 626–688.
[3] Abtin Zohrabi Aliabadi, Fatemeh Razzaghi, Seyed Pooria Madani Kochak, and Ali Akbar Ghorbani. 2013. Classifying organizational roles using email social networks. In *Canadian AI*. Springer, 301–307.
[4] Avrim Blum and Shuchi Chawla. 2001. Learning from Labeled and Unlabeled Data Using Graph Mincuts. In *ICML*. 19–26.
[5] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2016. Deep Neural Networks for Learning Graph Representations. In *AAAI*. 1145–1152.
[6] Michael S Cole and Heike Bruch. 2006. Organizational identity strength, identification, and commitment and their relationships to turnover intention: Does organizational hierarchy matter? *J. Occup. Organ. Psychol* 27, 5 (2006), 585–605.
[7] Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. 2018. Learning Structural Node Embeddings via Diffusion Wavelets. In *KDD, ACM*. 1320–1329.
[8] Petros Drineas and Michael W Mahoney. 2005. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *JMLR* 6 (2005), 2153–2175.
[9] Russell R. Dynes and B.E. Aguirre. 1979. Organizational Adaptation To Crises: Mechanisms Of Coordination And Structural Change. *Disasters* 3, 1 (1979), 71–74.
[10] Enron [n. d.]. Enron dataset. "http://www.cs.cmu.edu/ enron".
[11] Wolfgang Gatterbauer, Stephan Günnemann, Danai Koutra, and Christos Faloutsos. 2015. Linearized and Single-Pass Belief Propagation. *PVLDB* 8, 5 (2015).
[12] Palash Goyal and Emilio Ferrara. 2018. Graph embedding techniques, applications, and performance: A survey. *Knowl.-Based Syst.* 151 (2018), 78–94.
[13] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*. 855–864.
[14] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation Learning on Graphs: Methods and Applications. *IEEE Data Eng. Bull.* 40, 3 (2017), 52–74.
[15] Mark S. Handcock, David Hunter, Carter T. Butts, Steven M. Goodreau, and Martina Morris. 2003. statnet: An R package for the Statistical Modeling of Social Networks. http://www.csde.washington.edu/statnet.
[16] James R Harris. 1990. Ethical values of individuals at different levels in the organizational hierarchy of a single firm. *J Bus Ethics* 9, 9 (1990), 741–750.
[17] Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. 2018. REGAL: Representation Learning-based Graph Alignment. In *CIKM*.
[18] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. 2012. Rolx: structural role extraction & mining in large graphs. In *KDD*. ACM, 1231–1239.
[19] Xia Hu and Huan Liu. 2012. Social status and role analysis of Palin's email network. In *WWW (companion)*. ACM, 531–532.
[20] Danai Koutra, Tai-You Ke, U Kang, Duen Horng Chau, Hsing-Kuo Kenneth Pao, and Christos Faloutsos. 2011. Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms. In *ECML PKDD*. 245–260.
[21] Danai Koutra, Joshua Vogelstein, and Christos Faloutsos. 2013. DeltaCon: A Principled Massive-Graph Similarity Function. In *SDM*. 162–170.
[22] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. 2012. Sampling methods for the Nyström method. *JMLR* 13, Apr, 981–1006.
[23] Matthew Lynley. 2018. Slack hits 8 million daily active users with 3 million paid users. https://goo.gl/rSDQAZ (Timecrunch.com). Online; accessed 20-Jan-2018.
[24] Byung-Won On, Ee-Peng Lim, Jing Jiang, and Loo-Nin Teow. 2013. Engagingness and responsiveness behavior models on the enron email network and its application to email reply order prediction. In *The influence of technology on social network analysis and mining*. Springer, 227–253.
[25] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*.
[26] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network Embedding As Matrix Factorization: Unifying DeepWalk, LINE, PTE, and Node2Vec. In *WSDM*. ACM, 459–467.
[27] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *KDD*. ACM, 385–394.
[28] Ryan Rowe, German Creamer, Shlomo Hershkop, and Salvatore J Stolfo. 2007. Automated social hierarchy detection through email network analysis. In *WebKDD / SNA-KDD*. ACM, 109–117.
[29] Jitesh Shetty and Jafar Adibi. 2006. *The Enron Email Dataset Database Schema and Brief Statistical Report.* Technical Report. Online; accessed 3-June-2018.
[30] Salvatore J Stolfo, Shlomo Hershkop, Chia-Wei Hu, Wei-Jen Li, Olivier Nimeskern, and Ke Wang. 2006. Behavior-based modeling and its application to email analysis. *TOIT* 6, 2 (2006), 187–221.
[31] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale information network embedding. In *WWW*.
[32] Inc. The Radicati Group. 2018. Email Statistics Report. https://bit.ly/2MBL1wA (radicati.com). Online; accessed 20-January-2018.
[33] Chi-Yao Tseng, Jen-Wei Huang, and Ming-Syan Chen. 2007. ProMail: Using Progressive Email Social Network for Spam Detection. In *Adv. Knowl. Disc. and Data Min.* Springer, 833–840.
[34] Qinna Wang. 2014. Link prediction and threads in email networks. In *DSAA*. 470–476.
[35] Christopher KI Williams and Matthias Seeger. 2001. Using the Nyström method to speed up kernel machines. In *NIPS*. 682–688.
[36] Garnett Wilson and Wolfgang Banzhaf. 2009. Discovery of email communication networks from the enron corpus with a genetic algorithm using social network analysis. In *CEC*. IEEE, 3256–3263.
[37] Liu Yang, Susan T Dumais, Paul N Bennett, and Ahmed Hassan Awadallah. 2017. Characterizing and Predicting Enterprise Email Reply Behavior. In *SIGIR*. ACM, 235–244.
[38] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. 2003. Understanding Belief Propagation and its Generalizations. In *Exploring Artificial Intelligence in the New Millennium*. 239–269.
[39] Tian Zhu, Bin Wu, and Bai Wang. 2009. Social influence and role analysis based on community structure in social network. In *ADMA*. Springer, 788–795.
[40] Xiaojin Zhu. 2006. Semi-Supervised Learning Literature Survey.

Supplementary Material on Reproducibility

A EMBER: DETAILED ALGORITHM

In Section 4 we presented EMBER, and gave high-level pseudocode in Algorithm 1. Here we give a more detailed version of EMBER for replication purposes.

Algorithm 2 EMBER: EMBedding Email-based Roles

Input: Email network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, employees of interest $\mathcal{U} \subseteq \mathcal{V}$, maximum step K, discount factor $\delta \in (0, 1]$
Output: Professional roles for the employees of interest $\mathcal{U}$

`S1: Capture structural behavior in email network`

for user u in U **do** ▷ k-step in- and out- neighbors of u
  for step k up to K **do** ▷ $1 \leq K \leq$ graph diameter
    $\mathbf{b}_u^{k+}$ = Outdegree_Distribution($\mathcal{N}_u^+$) ▷ Eq. (2)
    $\mathbf{b}_u^{k-}$ = Indegree_Distribution($\mathcal{N}_u^-$)
  end for
  $\mathbf{b}_u^+ = \sum_{k=0}^{K} \delta^k \, \mathbf{b}_u^{k+}$ ▷ Out-neighborhood behavior – Eq. (3)
  $\mathbf{b}_u^- = \sum_{k=0}^{K} \delta^k \, \mathbf{b}_u^{k-}$ ▷ In-neighborhood behavior – Eq. (3)
  $\mathbf{b}_u = [\mathbf{b}_u^+ \; \mathbf{b}_u^-]$ ▷ Concatenation of in- and out-behaviors
end for

`S2: Embed employees in email network`

$\mathcal{L}$ = LandmarkSelection($\mathcal{U}$,p) ▷ $|\mathcal{L}| = p$ users selected from $\mathcal{U}$
for user u in $\mathcal{U}$ **do**
  for user v in $\mathcal{L}$ **do**
    $c_{uv} = e^{-||\mathbf{b}_u - \mathbf{b}_v||_2^2}$ ▷ $sim(u, v)$ based on § 4.3.1
  end for
end for
$\mathbf{M} = \mathbf{C}[\mathcal{L}, \mathcal{L}]$ ▷ Submatrix of $\mathbf{C}$ induced on the landmark users $\mathcal{L}$
$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{SVD}(\mathbf{M}^\dagger)$ ▷ $\mathbf{M}^\dagger$: pseudoinverse of $\mathbf{M}$
$\mathbf{Y} = \mathbf{C}\mathbf{U}\mathbf{\Sigma}^{-\frac{1}{2}}$ ▷ *Implicit* factorization of similarity graph
$\mathbf{Y} = \text{Normalization}(\mathbf{Y})$ ▷ Normalize embeddings to have magnitude 1

`S3: Professional role inference`

$[C_1, C_2, \ldots, C_*] = \text{classification}(\mathbf{Y}, \mathcal{T})$ ▷ Multi-class (C_k) classification

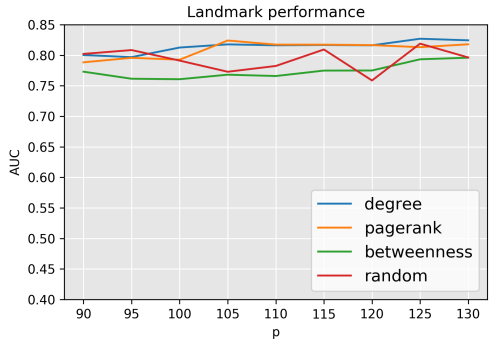
B LANDMARK SELECTION

How does the process of selecting landmarks impact the performance of EMBER? How many landmarks should one select, and based on what criterion? To answer these questions we perform role inference on our email networks, while varying the number of landmarks, and sampling them with probability proportional to their: (1) degree; (2) PageRank; (3) betweenness; or (4) randomly.

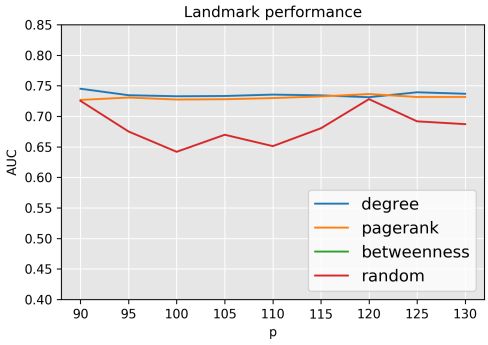
For brevity, we show the results for `Trove-318` and `Enron` in Fig. 5, since we observe similar patterns for other datasets. The degree-based landmark selection consistently outperforms other sampling methods in accuracy and robustness. The random approach unsurprisingly exhbits the most variation.

C ADDITIONAL ANALYSES

In Section 5.4 we addressed question **Q3**, where we investigated whether we can use the EMBER embeddings to compare professional roles across organizations. In Fig. 6, we present the full position correspondence matrix for all the companies that we studied in the `Trove-2K` email network. Our additional empirical results support our main observations in Section 5.4 about role differences across organizations of different sizes.



(a) `Trove-318`



(b) `Enron`

Figure 5: Performance of 4 landmark selection approaches on the `Trove-318` and `Enron` networks. X-axis: Number of landmarks; Y-axis: AUC. Random selection gives the most unstable performance, while degree-based selection gives stable and in most cases, the best performance in AUC. Note that for `Enron`, betweenness was too slow to run in a reasonable amount of time (3hrs).

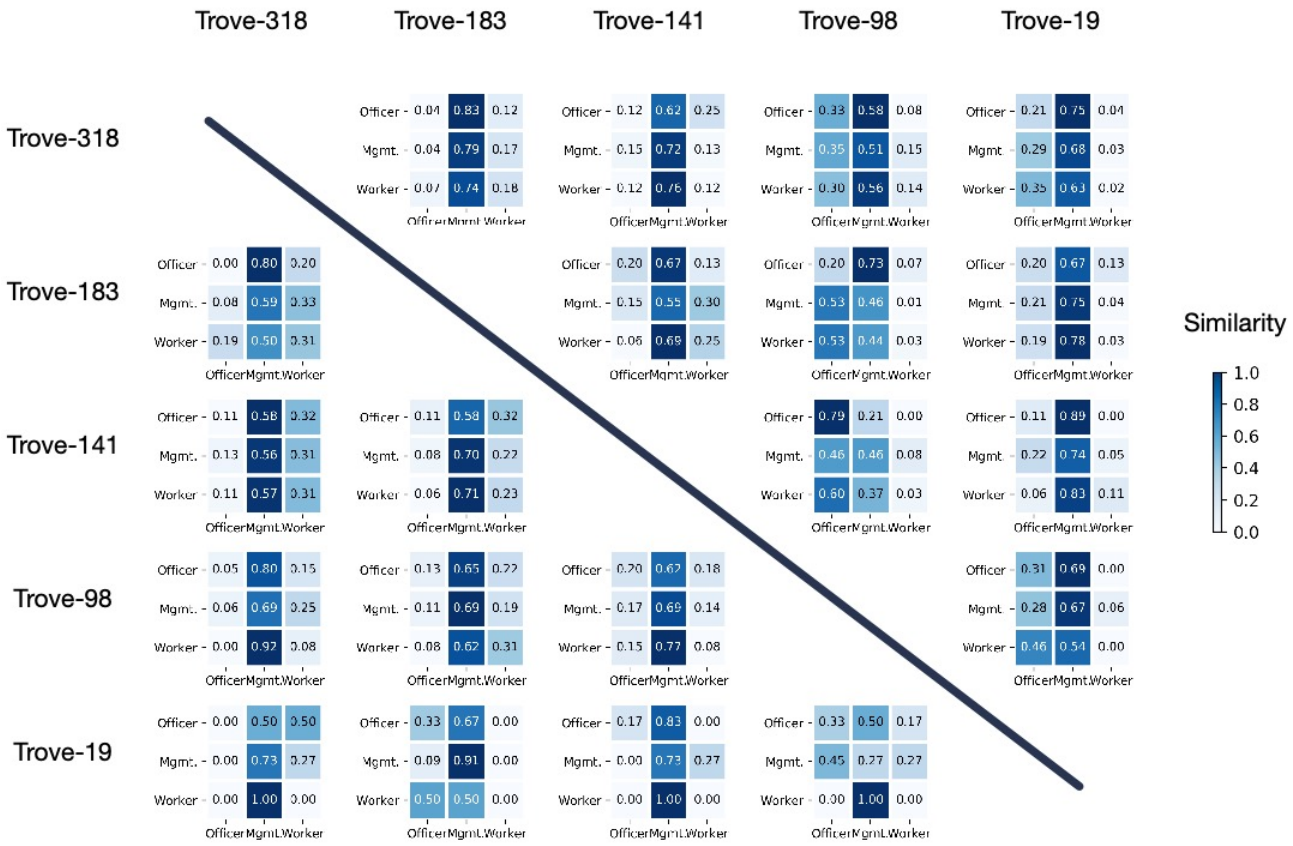


Figure 6: The position correspondence matrix between companies of different sizes. Each entry is normalized per row, higher values indicate higher similarity. The middle management employees across all companies share high behavioral similarity. Comparisons above the border indicate "mapping" positions from big companies to smaller ones where high similarity occur at the left-half of the matrix. This indicates that positions in any hierarchy from large companies tend to behave like "upper"-class in smaller companies (e.g., CEO, VP). Plots below the border illustrate the opposite pattern.