Transfer Learning for Latent Variable Network Models

Akhil Jalan
Department of Computer Science
UT Austin
akhiljalan@utexas.edu

Arya Mazumdar
Halıcıoğlu Data Science Institute & Dept of CSE
UC San Diego
arya@ucsd.edu

Soumendu Sundar Mukherjee
Statistics and Mathematics Unit (SMU)
Indian Statistical Institute, Kolkata
ssmukherjee@isical.ac.in

Purnamrita Sarkar
Department of Statistics and Data Sciences
UT Austin
purna.sarkar@austin.utexas.edu

Abstract

We study transfer learning for estimation in latent variable network models. In our setting, the conditional edge probability matrices given the latent variables are represented by P for the source and Q for the target. We wish to estimate Q given two kinds of data: (1) edge data from a subgraph induced by an $o(1)$ fraction of the nodes of Q, and (2) edge data from all of P. If the source P has no relation to the target Q, the estimation error must be $\Omega(1)$. However, we show that if the latent variables are shared, then vanishing error is possible. We give an efficient algorithm that utilizes the ordering of a suitably defined graph distance. Our algorithm achieves $o(1)$ error and does not assume a parametric form on the source or target networks. Next, for the specific case of Stochastic Block Models we prove a minimax lower bound and show that a simple algorithm achieves this rate. Finally, we empirically demonstrate our algorithm's use on real-world and simulated network estimation problems.

1 Introduction

Within machine learning and statistics, the paradigm of *transfer learning* describes a setup where data from a source distribution P is exploited to improve estimation of a target distribution Q for which a small amount of data is available. Transfer learning is quite well-studied in learning theory, starting with works such as Ben-David et al. (2006); Cortes et al. (2008); Crammer et al. (2008), and at the same time has found applications in areas such as computer vision (Tzeng et al., 2017a) and speech recognition (Huang et al., 2013). A fairly large body of work in transfer learning considers different types of relations that may exist between P and Q, for example, Mansour et al. (2009); Hanneke and Kpotufe (2019, 2022), with emphasis on model selection, multitask learning and domain adaptation. On the other hand, optimal nonparametric rates for transfer learning have very recently been studied, both for regression and classification problems (Cai and Wei, 2021; Cai and Pu, 2024).

In this paper, we study transfer learning in the context of *random network/graph models*. In our setting, we observe Bernoulli samples from the full $n \times n$ edge probability matrix for the source P and only a $n_Q \times n_Q$ submatrix of Q for $n_Q \ll n$. We would like to estimate the full $n \times n$ probability matrix Q, using the full source data and limited target data, i.e., we are interested in the task of estimating Q in the partially observed target network, utilizing information from the fully observed source network. This is a natural extension of the transfer learning problem in classification/regression

to a network context. However, it is to be noted that network transfer is a genuinely different problem owing to the presence of edge correlations.

While transfer learning in graphs seems to be a fundamental enough problem to warrant attention by itself, we are also motivated by potential applications. For example, metabolic networks model the chemical interactions related to the release and utilization of energy within an organism (Christensen and Nielsen, 2000). Existing algorithms for metabolic network estimation (Sen et al., 2018; Baranwal et al., 2020) and biological network estimation more broadly (Fan et al., 2019; Li et al., 2022) typically assume that some edges are observed for every node in the target network. One exception is Kshirsagar et al. (2013), who leverage side information for host-pathogen protein interaction networks. For the case of metabolic networks, determining interactions *in vivo*[1] requires metabolite balancing and labeling experiments, so only the edges whose endpoints are *both* incident to the experimentally chosen metabolites are observed (Christensen and Nielsen, 2000). For a non-model organism, the experimentally tested metabolites may be a small fraction of all metabolites believed to affect metabolism. However, data for a larger set of metabolites might be available for a model organism.

To study transfer learning on networks, one needs to fix a general enough class of networks that is appropriate for the applications (such as the biological networks mentioned above) and also suitable to capture the transfer phenomenon. The latent variable models defined below appear to be a natural candidate for that.

Latent Variable Models. Latent variable network models consist of a large class of models whose edge probabilities are governed by the latent positions of nodes. This includes latent distance models, stochastic block models, random dot product graphs and mixed membership block models (Hoff et al., 2002; Hoff, 2007; Handcock et al., 2007; Holland et al., 1983; Rubin-Delanchy et al., 2022; Airoldi et al., 2008). They can also be unified under graph limits or graphons (Lovász, 2012; Bickel and Chen, 2009), which provide a natural representation of vertex exchangeable graphs (Aldous, 1981; Hoover, 1979). In addition to their theoretical breadth and usefulness, latent variable models are relevant and applicable to real-world settings such as neuroscience Ren et al. (2023), ecology Trifonova et al. (2015), international relations Cao and Ward (2014), political pscyhology Barberá et al. (2015), and education research Sweet et al. (2013).

For unseen latent variables $x_1, \ldots, x_n \in \mathcal{X} \subset \mathbb{R}^d$ and unknown function $f_Q : \mathcal{X} \times \mathcal{X} \to [0, 1]$ where $\mathcal{X}$ is a compact set and d an arbitrary fixed dimension, the edge probabilities are:

$$Q_{ij} = f_Q(x_i, x_j). \tag{1}$$

Typically, in network estimation, one observes adjacency matrix $\{A_{ij}\}$ distributed as $\{\text{Bernoulli}(Q_{ij})\}$, and either has to learn x_i or directly estimate f_Q. There has been much work in the statistics community on estimating x_i for specific models (usually up to rotation). For stochastic block models, see the excellent survey in Abbe (2017).

Estimating f_Q can be done with some additional assumptions (Chatterjee, 2015). When f_Q has appropriate smoothness properties, one can estimate it by a histogram approximation (Olhede and Wolfe, 2014; Chan and Airoldi, 2014). This setting has also been compared to nonparametric regression with an unknown design (Gao et al., 2015). Methods for network estimation include Universal Singular Value Thresholding (Chatterjee, 2015; Xu, 2018), combinatorial optimization (Gao et al., 2015; Klopp et al., 2017), and neighborhood smoothing (Zhang et al., 2017; Mukherjee and Chakrabarti, 2019).

Transfer Learning on Networks. We wish to estimate the target network Q. However, we only observe f_Q on $\binom{n_Q}{2}$ pairs of nodes, for a uniformly random subset of variables $S \subset \{1, 2, \ldots, n\}$. We assume S is vanishingly small, so $n_Q := |S| = o(n)$.

Absent additional information, we cannot hope to achieve $o(1)$ mean-squared error. To see this, suppose f_Q is a stochastic block model with 2 communities of equal size. For a node $i \notin S$, no edges incident to i are observed, so its community cannot be learned. Since $n_Q \ll n$, we will attain $\Omega(1)$ error overall. To attain error $o(1)$, we hope to leverage transfer learning from a source P if available. In fact, we give an efficient algorithm to achieve $o(1)$ error, formally stated in Section 2.

Theorem 1.1 (Theorem 2.3, Informal)**.** *There exists an efficient algorithm such that, if given source data $A_P \in \{0, 1\}^{n \times n}$ and target data $A_Q \in \{0, 1\}^{n_Q \times n_Q}$ coming from an* appropriate *pair (f_P, f_Q)*

---

[1] In the organism, as opposed to *in vitro* (in the lab).

of latent variable models, outputs $\widehat{Q} \in \mathbb{R}^{n \times n}$ such that

$$\mathbb{P}\left[\frac{1}{n^2}\|Q - \widehat{Q}\|_F^2 \le o(1)\right] \ge 1 - o(1).$$

There must be a relationship between P and Q for them to be an *appropriate* pair for transfer learning. We formalize this relationship below.

Relationship Between Source and Target. It is natural to consider pairs (f_P, f_Q) such that for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$, the difference $(f_P(\boldsymbol{x}, \boldsymbol{y}) - f_Q(\boldsymbol{x}, \boldsymbol{y}))$ is small. For example, Cai and Pu (2024) study transfer learning for nonparametric regression when $f_P - f_Q$ is close to a polynomial in $\boldsymbol{x}, \boldsymbol{y}$. But, requiring $f_P - f_Q$ to be pointwise small does not capture a broad class of pairs in the network setting. For example, if $f_P = \alpha f_Q$. Then $f_P - f_Q = (\alpha - 1)f_Q$ can be far from all polynomials if f_Q is, e.g. a Hölder-smooth graphon.[2] However, under the network model, this means A_P and A_Q are stochastically identical modulo one being α times denser than the other.

We will therefore consider pairs (f_P, f_Q) that are close in some measure of local graph structure. With this in mind, we use a graph distance introduced in Mao et al. (2021) for a different inference problem.

Definition 1.2 (Graph Distance). *Let $P \in [0,1]^{n \times n}$ be the probability matrix of a graph. For $i, j \in [n], i \neq j$, we define the graph distance between them as follows:*

$$d_P(i, j) := \|(\boldsymbol{e}_i - \boldsymbol{e}_j)^T P^2 (I - \boldsymbol{e}_i \boldsymbol{e}_i^T - \boldsymbol{e}_j \boldsymbol{e}_j^T)\|_2^2,$$

where $\boldsymbol{e}_i, \boldsymbol{e}_j \in \mathbb{R}^n$ are standard basis vectors.

Intuitively, this first computes the matrix P^2 of common neighbors, and then computes the distance between two rows of the same (ignoring the diagonal elements). We will require that f_P, f_Q satisfy a local similarity condition on the relative rankings of nodes with respect to this graph distance. Since we only estimate the probability matrix of Q, the condition is on the latent variables $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ of interest. The hope is that the proximity in graph distance reflects the proximity in latent positions.

Definition 1.3 (Rankings Assumption at Quantile h_n). *Let (P, Q) be a pair of graphs evaluated on n latent positions. We say (P, Q) satisfy the rankings assumption at quantile $h_n \le 1$ if there exists constant $C > 0$ such that for all $i \in [n]$ and all $j \neq i$, if j belongs to the bottom h_n-quantile of $d_P(i, \cdot)$, then j belongs to the bottom Ch_n-quantile of $d_Q(i, \cdot)$.*

To further motivate Definition 1.3, recall our motivating example of biological network estimation. Previous works require some form of similarity between networks to enable transfer Sen et al. (2018); Fan et al. (2019); Baranwal et al. (2020). For example, Kshirsagar et al. (2013) require a *commonality hypothesis*: if pathogens A, B target the same neighborhoods in a protein interaction network, one can transfer from A to B. Our rankings assumption similarly posits that to transfer knowledge from A to B, A and B have similar 2-hop neighborhood structures.

Note that Definition 1.3 involves quantiles of graph distances; therefore it is a *relative* condition, because it depends on a rank-ordering within both graphs P, Q before comparison. On the other hand, an *absolute* condition would require that for nodes $i, j \in [n]$, if e.g. $d_P(i, j) < 100$ then $d_Q(i, j) < C \cdot 100$. Our condition is more flexible and will hold for a larger set of graph pairs (P, Q), such as pairs where one graph is much more dense than the other.

Finally, to illustrate Definition 1.3, consider stochastic block models f_P, f_Q with $k_P \ge k_Q$ communities respectively. If nodes i, j are in the same communities then $P\boldsymbol{e}_i = P\boldsymbol{e}_j$, so $d_P(i, j) = 0$. We require that j minimizes $d_Q(i, \cdot)$. This occurs if and only if $d_Q(i, j) = 0$. Hence if i, j belong to the same community in P, they are in the same community in Q. Note that the converse is not necessary; we could have Q with 1 community and P with arbitrarily many communities.

With the relationship between the source and target defined by the rankings assumption, our contributions are as follows.

(1) Algorithm for Latent Variable Models. We provide an efficient Algorithm 1 for latent variable models with Hölder-smooth f_P, f_Q. The benefit of this algorithm is that it does not assume a parametric form of f_P and f_Q. We prove a guarantee on its error in Theorem 2.3.

---

[2]In fact, Cai and Pu (2024) highlight this exact setting as a direction for future work.

(2) Minimax Rates. We prove a minimax lower bound for Stochastic Block Models (SBMs) in Theorem 3.2. Moreover, we provide a simple Algorithm 2 that attains the minimax rate for this class (Proposition 3.4).

(3) Experimental Results on Real-World Data. We test both of our algorithms on real-world metabolic networks and dynamic email networks, as well as synthetic data (Section 4).

All proofs are deferred to the Appendix.

1.1 Other Related work

Transfer learning has recently drawn a lot of interest both in applied and theoretical communities. The notion of transferring knowledge from one domain with a lot of data to another with less available data has seen applications in epidemiology Apostolopoulos and Bessiana (2020), computer vision Long et al. (2015); Tzeng et al. (2017b); Huh et al. (2016); Donahue et al. (2014); Neyshabur et al. (2020), natural language processing Daumé (2007), etc. For a comprehensive survey see Zhuang et al. (2019); Weiss et al. (2016); Kim et al. (2022). Recently, there have also been advances in the theory of transfer learning Yang et al. (2013); Tripuraneni et al. (2020); Agarwal et al. (2023); Cai and Wei (2021); Cai and Pu (2024); Cody and Beling (2023).

In the context of networks, transfer learning is particularly useful since labeled data is typically hard to obtain. Tang et al. (2016) develop an algorithmic framework to transfer knowledge obtained using available labeled connections from a source network to do link prediction in a target network. Lee et al. (2017) proposes a deep learning framework for graph-structured data that incorporates transfer learning. They transfer geometric information from the source domain to enhance performance on related tasks in a target domain without the need for extensive new data or model training. The SGDA method Qiao et al. (2023) introduce adaptive shift parameters to mitigate domain shifts and propose pseudo-labeling of unlabeled nodes to alleviate label scarcity. Zou et al. (2021) proposes to transfer features from the previous network to the next one in the dynamic community detection problem. Simchowitz et al. (2023) work on combinatorial distribution shift for matrix completion, where only some rows and columns are given. A similar setting is used for link prediction in egocentrically sampled networks in Wu et al. (2018). Zhu et al. (2021) train a graph neural network for transfer based on an ego-graph-based loss function. Learning from observations of the full network and additional information from a game played on the network Leng et al. (2020); Rossi et al. (2022). Wu et al. (2024) study graph transfer learning for node regression in the Gaussian process setting, where the source and target networks are fully observed.

Levin et al. (2022) proposes an inference method from multiple networks all with the same mean but different variances. While our work is related, we do not assume $\mathbb{E}[P_{ij}] = \mathbb{E}[Q_{ij}]$. Cao et al. (2010) do joint link prediction on a collection of networks with the same link function but different parameters.

Another line of related but different work deals with multiplex networks (Lee et al., 2014, 2015; Iacovacci and Bianconi, 2016; Cozzo et al., 2018) and dynamic networks Sarkar and Moore (2005); Kim et al. (2018); Sewell and Chen (2015); Sarkar et al. (2012); Chang et al. (2024); Wang et al. (2023). One can think of transfer learning in clustering as clustering with side information. Prior works consider stochastic block models with noisy label information (Mossel and Xu, 2016; Mazumdar and Saha, 2017a) or oracle access to the latent structure (Mazumdar and Saha, 2017b).

Notation. We use lowercase letters a, b, c to denote (real) scalars, boldface $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ to denote vectors, and uppercase A, B, C to denote matrices. Let $a \vee b := \max\{a, b\}$ and $a \wedge b := \min\{a, b\}$. For integer $n > 0$, let $[n] := \{1, 2, \ldots, n\}$. For a subset $S \subset [n]$ and $A \in \mathbb{R}^{n \times n}$, let $A[S, S] \in \mathbb{R}^{|S| \times |S|}$ be the principal submatrix with row and column indices in S. We denote the ℓ_2 vector norm as $\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2$, dot product as $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$, and Frobenius norm as $\|A\| = \|A\|_F$. For functions $f, g : \mathbb{N} \to \mathbb{R}$ we let $f \lesssim g$ denote $f = O(g)$ and $f \gtrsim g$ denote $f = \Omega(g)$. All asymptotics $O(\cdot), o(\cdot), \Omega(\cdot), \omega(\cdot)$ are with respect to n_Q unless specified otherwise.

2 Estimating Latent Variable Models with Rankings

In this section, we present a computationally efficient transfer learning algorithm for latent variable models. Algorithm 1 learns the local structure of P based on graph distances (Definition 1.2). For

---

Algorithm 1 $\widehat{Q}$-Estimation for Latent Variable Models

---

1: **Input:** $A_P \in \{0,1\}^{n \times n}, A_Q \in \{0,1\}^{n_Q \times n_Q}, S \subset [n]$ s.t. $|S| = n_Q$.
2: Initialize $\widehat{Q} \in \mathbb{R}^{n \times n}$ to be all zeroes.
3: For all i, all $j \neq i$, compute graph distances:

$$d_{A_P}(i,j) := \|(e_i - e_j)^T (A_P)^2 (I - e_i e_i^T - e_j e_j^T)\|_2^2.$$

4: Fix a bandwidth $h \in (0,1)$ based on n, n_Q.
5: **for** $i = 1$ to n **do**
6:      Let $T_i^{A_P}(h) \subset S$ be bottom h-quantile of S with respect to $d_{A_P}(i, \cdot)$.
7:      **if** $i \in S$ **then**
8:          Update $T_i^{A_P}(h) \leftarrow T_i^{A_P}(h) \cup \{i\}$.
9:      **end if**
10: **end for**
11: **for** $i = 2$ to n **do**
12:      **for** $1 \leq j < i$ **do**
13:          Compute $\widehat{Q}_{ij} = \widehat{Q}_{ji}$ by averaging:

$$\widehat{Q}_{ij} := \frac{1}{|T_i^{A_P}(h)||T_j^{A_P}(h)|} \sum_{r \in T_i^{A_P}(h)} \sum_{s \in T_j^{A_P}(h)} A_{Q;rs}.$$

14:      **end for**
15: **end for**
16: Return $\widehat{Q}$.

---

each node i of P, it ranks the nodes in S with respect to the graph distance $d_P(i, \cdot)$. For most nodes $i, j \in [n]$, none of the edges incident to i or j are observed in Q. Therefore, we estimate $\widehat{Q}_{ij}$ by using the edge information about nodes $r, s \in S$ such that $d_P(i,r)$ and $d_P(j,s)$ are small.

Formally, we consider a model as in Eq. (1) with a compact latent space $\mathcal{X} \subset \mathbb{R}^d$ and latent variables sampled i.i.d. from the normalized Lebesgue measure on $\mathcal{X}$. We set $\mathcal{X} = [0,1]^d$ without loss of generality and assume that functions $f : \mathcal{X} \times \mathcal{X} \to [0,1]$ are α-Hölder-smooth.

Definition 2.1. *Let $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $\alpha > 0$. We say f is α-Hölder-smooth if there exists $C_\alpha > 0$ such that for all $\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{y} \in \mathcal{X}$,*

$$\sum_{\kappa \in \mathbb{N}^d : \sum_i \kappa_i = \lfloor \alpha \rfloor} \left| \frac{\partial^{\sum_i \kappa_i} f}{\partial x_1^{\kappa_1} \cdots \partial x_d^{\kappa_d}}(\boldsymbol{x}, \boldsymbol{y}) - \frac{\partial^{\sum_i \kappa_i} f}{\partial x_1^{\kappa_1} \cdots \partial x_d^{\kappa_d}}(\boldsymbol{x}', \boldsymbol{y}) \right| \leq C_\alpha \|\boldsymbol{x} - \boldsymbol{x}'\|_2^{\alpha \wedge 1}.$$

To exclude degenerate cases where a node may not have enough neighbors in latent space, we require the following assumption.

Assumption 2.2 (Assumption 3.2 of Mao et al. (2021)). *Let G be a graph on $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. There exist $c_2 > c_1 > 0$ and $\Delta_n = o(1)$ such that for all $\boldsymbol{x}_i, \boldsymbol{x}_j$,*

$$c_1 \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^{\alpha \wedge 1} - \Delta_n \leq \frac{1}{n^3} d_G(i,j) \leq c_2 \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^{\alpha \wedge 1}.$$

The second inequality follows directly from Hölder-smoothness, and the first is shown to hold for e.g. Generalized Random Dot Product Graphs, among others (Mao et al., 2021).

We establish the rate of estimation for Algorithm 1 below.

Theorem 2.3. *Let $\widehat{Q}$ be as in Algorithm 1. Let f_P be α-Hölder-smooth and f_Q be β-Hölder-smooth for $\beta \geq \alpha > 0$, and let c be an absolute constant. Suppose (P, Q) satisfy Definition 1.3 at $h_n = c\sqrt{\frac{\log n_Q}{n_Q}}$ and P satisfies Assumption 2.2 with $\Delta_n = O((\frac{\log n}{n_Q})^{\frac{1}{2} \vee \frac{\alpha \wedge 1}{d}})$. Then there exists an absolute constant $C > 0$ such that*

$$\mathbb{P}\left[\frac{1}{n^2} \|\widehat{Q} - Q\|_F^2 \lesssim \left(\frac{d}{2}\right)^{\frac{\beta \wedge 1}{2}} \left(\frac{\log n}{n_Q}\right)^{\frac{\beta \wedge 1}{2d}} \right] \geq 1 - n_Q^{-C}.$$

5

To parse Theorem 2.3, consider the effect of various parameter choices. First, observe that our upper bound scales quite slowly with n. Even if n is superpolynomial in n_Q, e.g. $n = n_Q^{\log n_Q}$, then $\log n = O((\log n_Q)^2) = n_Q^{o(1)}$, so the overall effect on the error is dominated by the n_Q term.

Second, the bound is worse in large dimensions, and scales exponentially in $\frac{1}{d}$. This kind of scaling also occurs in minimax lower bounds for nonparametric regression (Tsybakov, 2009), and upper bounds for smooth graphon estimation (Xu, 2018). However, we caution that nonparametric regression can be quite different from network estimation; it would be very interesting to know the dependence of dimension on minimax lower bounds for network estimation, but to the best of our knowledge this is an open problem. Finally notice that a greater smoothness β results in a smaller error, up to $\beta = 1$, exactly as in (Gao et al., 2015; Klopp et al., 2017; Xu, 2018).

3 Minimax Rates for Stochastic Block Models

In this section, we will show matching lower and upper bounds for a very structured class of latent variable models, namely, Stochastic Block Models (SBMs).

Definition 3.1 (SBM). *Let $P \in [0,1]^{n \times n}$. We say P is an (n, k)-SBM if there exist $B \in [0,1]^{k \times k}$ and $z : [n] \to [k]$ such that for all i, j, $P_{ij} = B_{z(i)z(j)}$. We refer to $z^{-1}(\{j\})$ as community $j \in [k]$.*

We first state a minimax lower bound, proved via Fano's method.

Theorem 3.2 (Minimax Lower Bound for SBMs). *Let $k_P \geq k_Q \geq 1$ with k_Q dividing k_P. Let $\mathcal{F}$ be the family of pairs (P, Q) where P is an (n, k_P)-SBM, Q is an (n, k_Q)-SBM, and (P, Q) satisfy Definition 1.3 at $h_n = 1/k_P$. Moreover, suppose $S \subset [n]$ is restricted to contain an equal number of nodes from communities $1, 2, \ldots, k_P$ of P. Then the minimax rate of estimation is:*

$$\inf_{\widehat{Q} \in [0,1]^{n \times n}} \sup_{(P,Q) \in \mathcal{F}} \mathbb{E}\left[\frac{1}{n^2}\|\widehat{Q} - Q\|_F^2\right] \gtrsim \frac{k_Q^2}{n_Q^2}.$$

Note that Definition 1.3 at $h_n = 1/k_P$ implies that the true community structure of Q coarsens that of P. The condition that k_Q divides k_P is merely a technical one that we assume for simplicity.

We remark that minimax lower bounds for smooth graphon estimation are established by first showing lower bounds for SBMs, and then constructing a graphon with the same block structure using smooth mollifiers (Gao et al., 2015). Therefore, we expect that Theorem 3.2 can also be extended to the graphon setting, using the same techniques. However, sharp lower bounds for other classes such as Random Dot Product Graphs will likely require different techniques (Xie and Xu, 2020; Yan and Levin, 2023).

Remark 3.3 (Clustering Regime). *In Appendix A.4 we also prove a minimax lower bound of $\frac{\log k_Q}{n_Q}$ in the regime where the error of recovering the true clustering z dominates. This matches the rate of Gao et al. (2015), but for estimating all n^2 entries of Q, rather than just the n_Q^2 observed entries.*

Theorem 3.2 suggests that a very simple algorithm might achieve the minimax rate. Namely, use both A_P, A_Q to learn communities, and then use only A_Q to learn inter-community edge probabilities. If (P, Q) are in the nonparametric regime where regression error dominates clustering error (called the *weak consistency* or *almost exact recovery* regime), then the overall error will hopefully match the minimax rate.

We formalize this approach in Algorithm 2, and prove that it does achieve the minimax error rate in the weak consistency regime. To this end, we define the signal-to-noise ratio of an SBM with parameter $B \in [0,1]^{k \times k}$ as follows:

$$s := \frac{p - q}{\sqrt{p(1 - q)}},$$

where $p = \min_i B_{ii}, q = \max_{i \neq j} B_{ij}$.

Proposition 3.4 (Error Rate of Algorithm 2). *Suppose $P, Q \in [0,1]^{n \times n}$ are $(n, k_P), (n, k_Q)$-SBMs with minimum community sizes $n_{\min}^{(P)}, n_{\min}^{(Q)}$ respectively. Suppose also that (P, Q) satisfy*

Algorithm 2 $\widehat{Q}$-Estimation for Stochastic Block Models
___

1: **Input:** $A_P \in \{0,1\}^{n \times n}, A_Q \in \{0,1\}^{n_Q \times n_Q}, S \subset [n]$ s.t. $|S| = n_Q$.
2: Estimate clusterings $\widehat{Z}_P \in \{0,1\}^{n \times k_P}, \widehat{Z}_Q \in \{0,1\}^{n_Q \times k_Q}$ using Chen et al. (2014) on A_P, A_Q respectively.
3: Let $\widehat{W}_Q \in \mathbb{R}^{k_Q \times k_Q}$ be diagonal with

$$\widehat{W}_{Q;ii} = (\mathbf{1}^T \widehat{Z}_Q \boldsymbol{e}_i)^{-1}.$$

4: Initialize $\widehat{\Pi} \in \{0,1\}^{k_P \times k_Q}$ to be all zeroes.
5: **for** $i \in S$ **do**
6:     Let $j_P \in [k_P], j_Q \in [k_Q]$ be the unique column indices at which row i of $\widehat{Z}_P, \widehat{Z}_Q$ respectively are nonzero.
7:     Let $\widehat{\Pi}_{j_P,j_Q} = 1$.
8: **end for**
9: Let $\widehat{B}_Q \in [0,1]^{k_Q \times k_Q}$ be the block-average:

$$\widehat{B}_Q = \widehat{W}_Q \widehat{Z}_Q^T A_Q \widehat{Z}_Q \widehat{W}_Q.$$

10: **return** $\widehat{Q} := \widehat{Z}_P \widehat{\Pi} \widehat{B}_Q \widehat{\Pi}^T \widehat{Z}_P^T$.
___

Definition 1.3 at $h_n = n_{\min}^{(P)}/n$. Then if the signal-to-noise ratios are such that: $s_P \geq C(\frac{\sqrt{n}}{n_{\min}^{(P)}} \vee \frac{\log^2(n)}{\sqrt{n_{\min}^{(P)}}})$ and $s_Q \geq C(\frac{\sqrt{n_Q}}{n_{\min(Q)}} \vee \frac{\log^2(n_Q)}{\sqrt{n_{\min}^{(Q)}}})$ for large enough constant $C > 0$, Algorithm 2 returns $\widehat{Q}$ such that

$$\mathbb{P}\left[\frac{1}{n^2}\|\widehat{Q} - Q\|_F^2 \lesssim \frac{k_Q^2 \log(n_{\min}^{(Q)})}{n_Q^2}\right] \geq 1 - O\left(\frac{1}{n_Q}\right).$$

4 Experiments

In this section, we test Algorithm 1 against several classes of simulated and real-world networks. We use quantile cutoff of $h_n = \sqrt{\frac{\log n_Q}{n_Q}}$ for Algorithm 1 in all experiments.

Baselines. To the best of our knowledge, our exact transfer formulation has not been considered before in the literature. Therefore, we implement two algorithms as alternatives to Algorithm 1.

(1) Algorithm 2. Given $A_P \in \{0,1\}^{n \times n}, A_Q \in \{0,1\}^{n_Q \times n_Q}$, let $k_P = \lceil \sqrt{n} \rceil, k_Q = \lceil \sqrt{n_Q} \rceil$. Compute spectral clusterings $\widehat{Z}_P, \widehat{Z}_Q$ with k_P, k_Q clusters respectively. Let $J_S \in \{0,1\}^{n_Q \times n}$ is such that $J_{S;ij} = 1$ if and only if $i = j$ and $i \in S$. The projection $\widehat{\Pi} \in \mathbb{R}^{k_P \times k_Q}$ solves the least-squares problem $\min_{\Pi \in \mathbb{R}^{k_P \times k_Q}} \|J_S \widehat{Z}_P \Pi - \widehat{Z}_Q\|_F^2$. We compute the $\widehat{\Pi}$ differently from steps 4-7 in Algorithm 2 to account for cases where Q is not a true coarsening of P. When Q is a true coarsening of P, this reduces to the procedure in steps 4-7. Given $\widehat{Z}_P, \widehat{\Pi}$ we return $\widehat{Q}$ as in Algorithm 2.

(2) Oracle. Suppose that an oracle can access data for Q on *all* $n \gg n_Q$ nodes as follows. Fix an error probability $p \in (0,1)$. The oracle is given symmetric $A'_Q \in \{0,1\}^{n \times n}$ with independent entries following a mixture distribution. For all $i, j \in [n]$ with $i < j$ let $X_{ij} \sim$ Bernoulli(p) and $Y_{ij} \sim$ Bernoulli$(Q(\boldsymbol{x}_i, \boldsymbol{x}_j))$. Then:

$$A'_{Q;ij} = \mathbb{1}_{i \in S, j \in S} Y_{ij} + (1 - \mathbb{1}_{i \in S, j \in S})((1 - X_{ij})Y_{ij} + X_{ij}(1 - Y_{ij})).$$

Given A'_Q, the oracle returns the estimate from Universal Singular Value Thresholding on A'_Q Chatterjee (2015). As $p \to 0$, the error will approach $O(n^{\frac{-2\beta}{2\beta+d}})$ for a β-smooth network on on d-dimensional latent variables (Xu, 2018), so the oracle will outperform any transfer algorithm.

Source	Target	Alg. 1	Alg. 2	Oracle $(p=0.1)$	Oracle $(p=0.3)$	Oracle $(p=0.5)$
Noisy-MMSB $(0.7, 0.3, 0.01)$	Noisy-MMSB $(0.9, 0.1, 0.01)$	**0.7473**± **0.0648**	1.3761 ± 1.1586	*0.9556* ± *0.0633*	2.2568 ± 0.3107	4.2212 ± 0.2825
0.1-Smooth Graphon	0.5-Smooth Graphon	*1.7656* ± *0.7494*	4.5033 ± 1.5613	**0.5016** ± **0.0562**	2.4423 ± 0.4574	5.7774 ± 0.7126
$\mathbb{R}^{10}$ Latent(2.5)	$\mathbb{R}^{10}$ Latent(1.0)	**0.5744** ± **0.1086**	1.1773 ± 1.0481	*0.7715* ± *0.0456*	2.1822 ± 0.2741	4.3335 ± 0.3476

Table 1: Comparison of different algorithms on simulated networks. Each cell reports $\widehat{\mu} \pm 2\widehat{\sigma}$ of the mean-squared error over 50 independent trials. Error numbers are all scaled by $1e2$ for ease of reading. Bold: Best algorithm. Emphasis: Second-best algorithm.

Simulations. We first test on several classes of simulated networks. For $n_Q = 50, n = 200$, we run 50 independent trials for each setting. We report results for each setting in Table 1, and visualize estimates for stylized examples in Figure 1.

At a glance, Figure 1 shows that Algorithms 1 and 2 both work well on Stochastic Block Models (first row), that only Algorithm 1 works well on graphons (second and third rows), and that the Oracle performs well in all cases.

Smooth Graphons. The latent space is $\mathcal{X} = [0, 1]$. We consider graphons of the form $f_\gamma(x, y) = \frac{x^\gamma + y^\gamma}{2}$ where P, Q have different γ. We denote this the γ-Smooth Graphon.

Mixed-Membership Stochastic Block Model. Set $k_P = \lfloor\sqrt{n}\rfloor, k_Q = \lfloor\sqrt{n_Q}\rfloor$. The latent space $\mathcal{X}$ is the probability simplex $\mathcal{X} = \Delta_{k_P} := \{x \in [0, 1]^{k_P} : \sum_i x_i = 1\} \subset \mathbb{R}^{k_P}$. The latent variables $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are iid-Dirichlet distributed with equal weights $\frac{1}{k_P}, \ldots, \frac{1}{k_P}$. Then $P_{ij} = \boldsymbol{x}_i^T B_P \boldsymbol{x}_j$ and $Q_{ij} = \Pi(\boldsymbol{x}_i)^T B_Q \Pi(\boldsymbol{x}_j)$, for connectivity matrices $B_P \in [0, 1]^{k_P \times k_P}, B_Q \in [0, 1]^{k_Q \times k_Q}$, and projection $\Pi : \Delta_{k_P} \to \Delta_{k_Q}$ for a fixed subset of $[k_P]$. For parameters $a, b, \epsilon \in [0, 1]$ we generate $B \in [0, 1]^{k \times k}$ by sampling $E \in \text{Uniform}(-\epsilon, \epsilon)^{k \times k}$ and set $B = \text{clip}((a - b)I + b\boldsymbol{1}\boldsymbol{1}^T + E, 0, 1)$. We call this Noisy-MMSB(a, b, ϵ).

Latent Distance Model. The latent space is the unit sphere $\mathcal{X} = \mathbb{S}^{d-1} \subset \mathbb{R}^d$. For scale parameter $s > 0$, we call $f_s(\boldsymbol{x}, \boldsymbol{y}) = \exp(-s\|\boldsymbol{x} - \boldsymbol{y}\|_2)$ the $\mathbb{R}^d$-Latent(s) model.

Discussion. When the latent dimension is larger than 1 (the Noisy MMSB and Latent Variable Models), our Algorithm 1 is better than both Algorithm 2 and the Oracle with $p = 0.1$. Note that Algorithms 1 and 2 use $\frac{n_Q^2}{n^2} \approx 0.06$ unbiased edge observations from Q, while the Oracle with $p = 0.1$ observes $(1 - p)\frac{n^2 - n_Q^2}{n^2} \approx 0.9$ unbiased edge observations in expectation.

Real-World Data. Next, we test on two classes of real-world networks. We summarize our dataset characteristics in Table 2. See Appendix C for further details.

Table 2: Dataset Characteristics

Name	n	Median Degree	Type
BiGG Model iWFL1372	251	15.00	Source
BiGG Model iPC815	251	12.00	Source
BiGG Model iJN1463	251	14.00	Target
EMAIL-EU Days 1-80	1005	6.92	Source
EMAIL-EU Days 81-160	1005	7.35	Target
EMAIL-EU Days 561-640	1005	7.66	Target

Transfer Across Species in Metabolic Networks. For a fixed organism, a metabolic network has a node for each metabolite, and an edge exists if and only if two metabolites co-occur in a metabolic
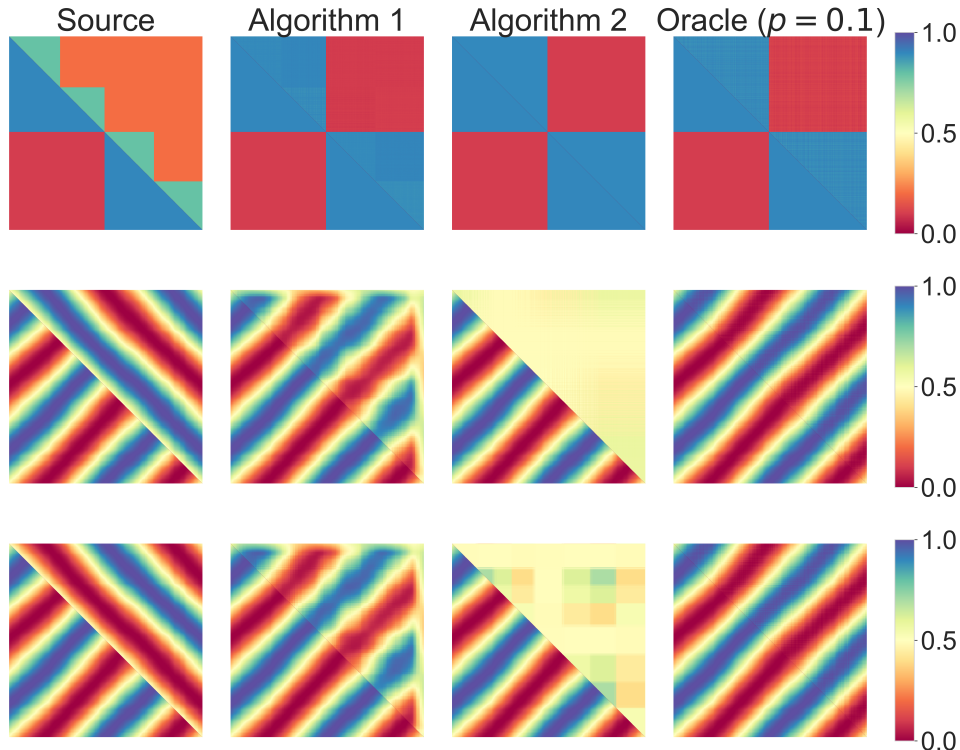
Figure 1: Comparison of algorithms on three source-target pairs ($n = 2000, n_Q = 500$). Each row corresponds to a different source/target pair (P, Q). For a fixed row, the upper triangular part on columns 2, 3, 4 corresponds a $\widehat{Q}$ for a different algorithm. The upper triangular part of column 1 shows the true P. The lower triangular part of columns 1, 2, 3, and 4 is identical for a fixed row, and shows the true Q. In each heatmap, the lower triangle is the target Q. Algorithm 2 performs best when (P, Q) are SBMs (top), while Algorithm 1 is better for smooth graphons (2nd and 3rd rows).

reaction in that organism. We obtain the unweighted metabolic networks for multiple gram-negative bacteria from the BiGG genome-scale metabolic model dataset (King et al., 2016; Norsigian et al., 2020). In the left half of Figure 2, we compare two choices of source organism in estimating the network for BiGG model iJN1463 (*Pseudomonas putida*). For a good choice of source, Algorithm 1 is competitive with the Oracle at $p = 0.1$.

Transfer Across Time in the Email Interaction Networks. We use the EMAIL-EU interaction network between $n = 1005$ members of a European research institution across 803 days Leskovec and Krevl (2014); Paranjape et al. (2017). The source graph A_P is the network from day 1 to ≈ 80 ([1, 80]). In Figure 2 we simulate transfer with targets [81, 160] (left) and [561, 640] (right). We visualize results for arbitrary target periods; similar results hold for other targets. Unlike metabolic networks, Algorithm 2 has comparable performance to both our Algorithm 1 and the oracle algorithm with $p \in \{0.01, 0.05\}$. Compared to the metabolic networks, this indicates that the email interaction networks are relatively well-approximated by SBMs, although Algorithm 1 is still the best.

Additional Experiments and Baseline. In Appendix B.1, we present additional ablation experiments that test the dependence of Algorithms 1 and 2 on all relevant parameters. We compare their performance to the Oracle baseline with $p = 0.0$ (the non-transfer setting), and an additional baseline adapted from Levin et al. (2022). We find that our Algorithms outperform this new baseline but are worse than the Oracle with $p = 0.0$, as expected. Further, in Appendix B.2, we test our Algorithms and original baselines on a link prediction task in the setting of Figure 2. We find that the relative accuracy of the methods for link prediction is qualitatively similar to that of Figure 2, and the Oracle performs even better with sparsity tuning.
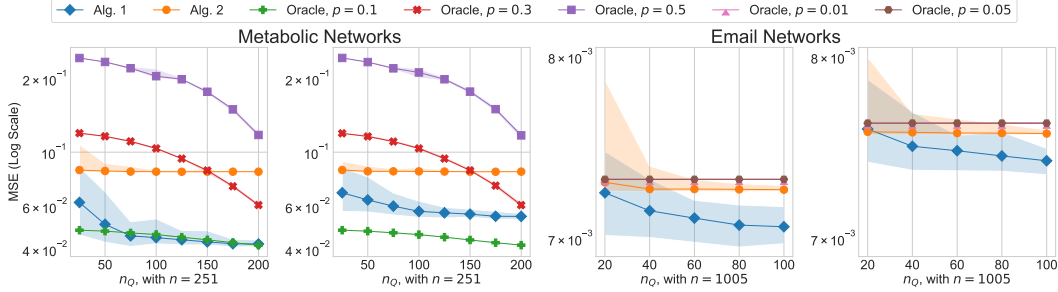
Figure 2: Results of network estimation on real-world data. Shaded regions denote $[1, 99]$ percentile outcomes from 50 trials.
Left half: Estimating metabolic network of iJN1463 (*Pseudomonas putida*) with source iWFL1372 (*Escherichia coli W*) leftmost, and source iPC815 (*Yersinia pestis*) second-left.
Right half: Using source data from days $1 - 80$ of EMAIL-EU to estimate target days $81 - 160$ (third-left) and target days $561 - 640$ (rightmost). Note that we use smaller values of p for the Oracle in EMAIL-EU.

5   Conclusion

In this paper, we study transfer learning for network estimation in latent variable models. We show that there exists an efficient Algorithm 1 that achieves vanishing error even when $n \geq n_Q^{\omega(1)}$, and a simpler Algorithm 2 for SBMs that achieves the minimax rate.

There are several interesting directions for future work.

First, we believe that Algorithm 1 works for moderately sparse networks with population edge density $\Omega(\frac{1}{\sqrt{n}})$. This is because the concentration of empirical graph distance (Algorithm 1 line 3) requires expected edge density $\widetilde{\Omega}(n^{-1/2})$ Mao et al. (2021). It would be interesting to see if a similar approach can work for edge density $\Omega(\frac{\log n}{n})$. For example, in the aforementioned paper it is shown that a variation of the graph distance of Definition 1.2 concentrates at expected edge density $\widetilde{\Omega}(n^{-2/3})$. While is this still far from the $\Omega(\frac{\log n}{n})$ regime, it suggests that variations on the graph distance might ensure our Algorithm 1 works for sparser graphs.

Second, the case of multiple sources is also interesting. We have focused on the case of one source distribution, as in Cai and Wei (2021); Cai and Pu (2024), but expect that our algorithms can be extended to multiple sources as long as they satisfy Definition 1.3.

Acknowledgments and Disclosure of Funding

References

Abbe, E. (2017). Community detection and stochastic block models. *arXiv preprint arXiv:1703.10146*.

Agarwal, A., Song, Y., Sun, W., Wang, K., Wang, M., and Zhang, X. (2023). Provable benefits of representational transfer in reinforcement learning. In Neu, G. and Rosasco, L., editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 2114–2187. PMLR.

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014.

Aldous, D. J. (1981). Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11:581–598.

Apostolopoulos, I. D. and Bessiana, T. (2020). Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*, 43:635 – 640.

Baranwal, M., Magner, A., Elvati, P., Saldinger, J., Violi, A., and Hero, A. O. (2020). A deep learning architecture for metabolic pathway prediction. *Bioinformatics*, 36(8):2547–2553.

Barberá, P., Jost, J. T., Nagler, J., Tucker, J. A., and Bonneau, R. (2015). Tweeting from left to right: Is online political communication more than an echo chamber? *Psychological science*, 26(10):1531–1542.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. (2006). Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19.

Bickel, P. J. and Chen, A. (2009). A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073.

Cai, T. T. and Pu, H. (2024). Transfer learning for nonparametric regression: Non-asymptotic minimax analysis and adaptive procedure. *arXiv preprint arXiv:2401.12272*.

Cai, T. T. and Wei, H. (2021). Transfer learning for nonparametric classification: Minimax rate and adaptive classifier. *The Annals of Statistics*, 49(1).

Cao, B., Liu, N. N., and Yang, Q. (2010). Transfer learning for collective link prediction in multiple heterogenous domains. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 159–166. Citeseer.

Cao, X. and Ward, M. D. (2014). Do democracies attract portfolio investment? transnational portfolio investments modeled as dynamic network. *International Interactions*, 40(2):216–245.

Chan, S. and Airoldi, E. (2014). A consistent histogram estimator for exchangeable graph models. In *International Conference on Machine Learning*, pages 208–216. PMLR.

Chang, S., Koehler, F., Qu, Z., Leskovec, J., and Ugander, J. (2024). Inferring dynamic networks from marginals with iterative proportional fitting. *arXiv preprint arXiv:2402.18697*.

Chatterjee, S. (2015). Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, pages 177–214.

Chen, Y., Sanghavi, S., and Xu, H. (2014). Improved graph clustering. *IEEE Transactions on Information Theory*, 60(10):6440–6455.

Christensen, B. and Nielsen, J. (2000). *Metabolic Network Analysis*, pages 209–231. Springer Berlin Heidelberg, Berlin, Heidelberg.

Cody, T. and Beling, P. A. (2023). A systems theory of transfer learning. *IEEE Systems Journal*, 17(1):26–37.

Cortes, C., Mohri, M., Riley, M., and Rostamizadeh, A. (2008). Sample selection bias correction theory. In *International conference on algorithmic learning theory*, pages 38–53. Springer.

Cozzo, E., De Arruda, G. F., Rodrigues, F. A., and Moreno, Y. (2018). *Multiplex networks: basic formalism and structural properties*, volume 10. Springer.

Crammer, K., Kearns, M., and Wortman, J. (2008). Learning from multiple sources. *Journal of Machine Learning Research*, 9(8).

Daumé, H. (2007). Frustratingly easy domain adaptation. *ArXiv*, abs/0907.1815.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 647–655, Bejing, China. PMLR.

Fan, J., Cannistra, A., Fried, I., Lim, T., Schaffner, T., Crovella, M., Hescott, B., and Leiserson, M. D. (2019). Functional protein representations from biological networks enable diverse cross-species inference. *Nucleic acids research*, 47(9):e51–e51.

Gao, C., Lu, Y., and Zhou, H. H. (2015). Rate-optimal graphon estimation. *The Annals of Statistics*, pages 2624–2652.

Guruswami, V., Rudra, A., and Sudan, M. (2019). Essential coding theory.

Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354.

Hanneke, S. and Kpotufe, S. (2019). On the value of target data in transfer learning. *Advances in Neural Information Processing Systems*, 32.

Hanneke, S. and Kpotufe, S. (2022). A no-free-lunch theorem for multitask learning. *The Annals of Statistics*, 50(6):3119–3143.

Hoeffding, W. (1994). Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pages 409–426.

Hoff, P. (2007). Modeling homophily and stochastic equivalence in symmetric relational data. *Advances in neural information processing systems*, 20.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent Space Approaches to Social Network Analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137.

Hoover, D. N. (1979). *Relations on probability spaces arrays of random variables*. Institute for Advanced Study,RI.

Huang, J.-T., Li, J., Yu, D., Deng, L., and Gong, Y. (2013). Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 7304–7308. IEEE.

Huh, M., Agrawal, P., and Efros, A. A. (2016). What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*.

Iacovacci, J. and Bianconi, G. (2016). Extracting information from multiplex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(6).

Kim, B., Lee, K. H., Xue, L., and Niu, X. (2018). A review of dynamic network models with latent variables. *Statistics Surveys*, 12(none):105 – 135.

Kim, H., Cosa-Linan, A., Santhanam, N., and et al. (2022). Transfer learning for medical image classification: a literature review. *BMC Medical Imaging*, 22(1):69.

King, Z. A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J. A., Ebrahim, A., Palsson, B. O., and Lewis, N. E. (2016). Bigg models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic acids research*, 44(D1):D515–D522.

Klopp, O., Tsybakov, A. B., and Verzelen, N. (2017). Oracle inequalities for network models and sparse graphon estimation. *Annals of Statistics*, 45(1):316–354.

Kshirsagar, M., Carbonell, J., and Klein-Seetharaman, J. (2013). Multitask learning for host–pathogen protein interactions. *Bioinformatics*, 29(13):i217–i226.

Lee, J., Kim, H., Lee, J., and Yoon, S. (2017). Transfer learning for deep learning on graph-structured data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Lee, K.-M., Kim, J. Y., Lee, S., and Goh, K.-I. (2014). Multiplex networks. *Networks of networks: The last frontier of complexity*, pages 53–72.

Lee, K.-M., Min, B., and Goh, K.-I. (2015). Towards real-world complexity: an introduction to multiplex networks. *The European Physical Journal B*, 88:1–20.

Leng, Y., Dong, X., Wu, J., and Pentland, A. (2020). Learning quadratic games on networks. In *International Conference on Machine Learning*, pages 5820–5830. PMLR.

Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

Levin, K., Lodhia, A., and Levina, E. (2022). Recovering shared structure from multiple networks with unknown edge distributions. *Journal of machine learning research*, 23(3):1–48.

Li, L., Dannenfelser, R., Zhu, Y., Hejduk, N., Segarra, S., and Yao, V. (2022). Joint embedding of biological networks for cross-species functional alignment. *bioRxiv*, pages 2022–01.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Lovász, L. (2012). *Large Networks and Graph Limits.*, volume 60 of *Colloquium Publications*. American Mathematical Society.

Mansour, Y., Mohri, M., and Rostamizadeh, A. (2009). Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*.

Mao, X., Chakrabarti, D., and Sarkar, P. (2021). Consistent nonparametric methods for network assisted covariate estimation. In *International Conference on Machine Learning*, pages 7435–7446. PMLR.

Mazumdar, A. and Saha, B. (2017a). Clustering with noisy queries. *Advances in Neural Information Processing Systems*, 30.

Mazumdar, A. and Saha, B. (2017b). Query complexity of clustering with side information. *Advances in Neural Information Processing Systems*, 30.

Mossel, E. and Xu, J. (2016). Local algorithms for block models with side information. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 71–80.

Mukherjee, S. S. and Chakrabarti, S. (2019). Graphon estimation from partially observed network data. *arXiv preprint arXiv:1906.00494*.

Neyshabur, B., Sedghi, H., and Zhang, C. (2020). What is being transferred in transfer learning? In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 512–523. Curran Associates, Inc.

Norsigian, C. J., Pusarla, N., McConn, J. L., Yurkovich, J. T., Dräger, A., Palsson, B. O., and King, Z. (2020). Bigg models 2020: multi-strain genome-scale models and expansion across the phylogenetic tree. *Nucleic acids research*, 48(D1):D402–D406.

Olhede, S. C. and Wolfe, P. J. (2014). Network histograms and universality of blockmodel approximation. *Proceedings of the National Academy of Sciences*, 111(41):14722–14727.

Paranjape, A., Benson, A. R., and Leskovec, J. (2017). Motifs in temporal networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 601–610.

Qiao, Z., Luo, X., Xiao, M., Dong, H., Zhou, Y., and Xiong, H. (2023). Semi-supervised domain adaptation in graph transfer learning. *ArXiv*, abs/2309.10773.

Ren, M., Zhang, S., and Wang, J. (2023). Consistent estimation of the number of communities via regularized network embedding. *Biometrics*, 79(3):2404–2416.

Rossi, E., Monti, F., Leng, Y., Bronstein, M., and Dong, X. (2022). Learning to infer structures of network games. In *International Conference on Machine Learning*, pages 18809–18827. PMLR.

Rubin-Delanchy, P., Cape, J., Tang, M., and Priebe, C. E. (2022). A statistical interpretation of spectral embedding: the generalised random dot product graph. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(4):1446–1473.

Sarkar, P., Chakrabarti, D., and Jordan, M. I. (2012). Nonparametric link prediction in dynamic networks. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 1897–1904, Madison, WI, USA. Omnipress.

Sarkar, P. and Moore, A. (2005). Dynamic social network analysis using latent space models. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press.

Sen, R., Tagore, S., and De, R. K. (2018). Asapp: Architectural similarity-based automated pathway prediction system and its application in host-pathogen interactions. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(2):506–515.

Sewell, D. K. and Chen, Y. (2015). Latent space models for dynamic networks. *Journal of the american statistical association*, 110(512):1646–1657.

Simchowitz, M., Gupta, A., and Zhang, K. (2023). Tackling combinatorial distribution shift: A matrix completion perspective. In Neu, G. and Rosasco, L., editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 3356–3468. PMLR.

Sweet, T. M., Thomas, A. C., and Junker, B. W. (2013). Hierarchical network models for education research: Hierarchical latent space models. *Journal of Educational and Behavioral Statistics*, 38(3):295–318.

Tang, J., Lou, T., Kleinberg, J., and Wu, S. (2016). Transfer learning to infer social ties across heterogeneous networks. *ACM Trans. Inf. Syst.*, 34(2).

Trifonova, N., Kenny, A., Maxwell, D., Duplisea, D., Fernandes, J., and Tucker, A. (2015). Spatio-temporal bayesian network models with latent variables for revealing trophic dynamics and functional networks in fisheries ecology. *Ecological Informatics*, 30:142–158.

Tripuraneni, N., Jordan, M. I., and Jin, C. (2020). On the theory of transfer learning: the importance of task diversity. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. Springer series in statistics. Springer, Dordrecht.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017a). Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017b). Adversarial discriminative domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971.

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.

Wang, H., Mao, Y., Sun, J., Zhang, S., Fan, Y., and Zhou, D. (2023). Dynamic transfer learning across graphs. *arXiv preprint arXiv:2305.00664*.

Weiss, K. R., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, 3:1–40.

Wu, J., Ainsworth, L., Leakey, A., Wang, H., and He, J. (2024). Graph-structured gaussian processes for transferable graph learning. *Advances in Neural Information Processing Systems*, 36.

Wu, Y.-J., Levina, E., and Zhu, J. (2018). Link prediction for egocentrically sampled networks. *Journal of Computational and Graphical Statistics*, 32:1296 – 1319.

Xie, F. and Xu, Y. (2020). Optimal bayesian estimation for random dot product graphs. *Biometrika*, 107(4):875–889.

Xu, J. (2018). Rates of convergence of spectral methods for graphon estimation. In *International Conference on Machine Learning*, pages 5433–5442. PMLR.

Yan, H. and Levin, K. (2023). Minimax rates for latent position estimation in the generalized random dot product graph. *arXiv preprint arXiv:2307.01942*.

Yang, L., Hanneke, S., and Carbonell, J. (2013). A theory of transfer learning with applications to active learning. *Machine learning*, 90:161–189.

Yu, B. (1997). Assouad, fano, and le cam. In *Festschrift for Lucien Le Cam: research papers in probability and statistics*, pages 423–435. Springer.

Zhang, Y., Levina, E., and Zhu, J. (2017). Estimating network edge probabilities by neighbourhood smoothing. *Biometrika*, 104(4):771–783.

Zhu, Q., Yang, C., Xu, Y., Wang, H., Zhang, C., and Han, J. (2021). Transfer learning of graph neural networks with ego-graph information maximization. *Advances in Neural Information Processing Systems*, 34:1766–1779.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2019). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109:43–76.

Zou, J., Lin, F., Gao, S., Deng, G., Zeng, W., and Alterovitz, G. (2021). Transfer learning based multi-objective genetic algorithm for dynamic community detection. *arXiv preprint arXiv:2109.15136*.

A Proofs

A.1 Preliminaries

Recall Hoeffding's inequality.

Lemma A.1 (Hoeffding (1994))**.** *Let $X_1, \ldots, X_n$ be independent random variables such that $a_i \leq X_i \leq b_i$ almost surely for all $i \in [n]$. Then*

$$\mathbb{P}\left[\left|\sum_{i=1}^{n}(X_i - \mathbb{E}[X_i])\right| \geq t\right] \leq 2\exp\left(\frac{-2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

We also need Bernstein's inequality.

Lemma A.2 (Bernstein's Inequality)**.** *Let $X_1, \ldots, X_n$ be independent mean-zero random variables with $|X_i| \leq 1$ for all i and $n \geq 5$. Then*

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^{n}X_i\right| \geq t\right] \leq 2\exp\left(\frac{-nt^2}{2(1+\frac{t}{3})}\right) \leq 2\exp\left(-\frac{nt^2}{4}\right).$$

A.2 Proof of Theorem 2.3

Throughout this section, let $\mathcal{X} = [0,1]^d$ and $\mu : \mathcal{X} \to [0,1]$ be the normalized Lebesgue measure.

We require the following Lemmata.

Lemma A.3. *Let $\upsilon \in (0,1)$ and $\mu : \mathcal{X} \to [0,1]$ be the normalized Lebesgue measure. Then for all $\boldsymbol{x} \in \mathcal{X}$,*

$$\mu(\mathrm{Ball}(\boldsymbol{x}, 2\upsilon) \cap \mathcal{X}) \geq \mu(\mathrm{Ball}(\boldsymbol{0}, \upsilon) \cap \mathcal{X}).$$

Proof. Recall $\mathcal{X} = [0,1]^d$. Fix $\boldsymbol{x} \in \mathcal{X}, \upsilon > 0$. Note that $\mu(\mathrm{Ball}(\boldsymbol{x}, \upsilon) \cap \mathcal{X})$ is smallest when $\boldsymbol{x}$ is a vertex of the hypercube; therefore take $\boldsymbol{x} \in \{0,1\}^d$ without loss of generality. Then, note that for each $\boldsymbol{z} \in \mathrm{Ball}(\boldsymbol{x}, \upsilon) \cap \mathcal{X}$, we can find $(2^d - 1)$ other points $\boldsymbol{z}' \in \mathrm{Ball}(\boldsymbol{x}, \upsilon) \setminus \mathcal{X}$ by reflecting subsets of coordinates of $\boldsymbol{z}$ about $\boldsymbol{x}$. There are $2^d - 1$ such nonempty subsets of coordinates. This shows that $\mu(\mathrm{Ball}(\boldsymbol{x}, \upsilon) \cap \mathcal{X}) \geq \mu(\mathrm{Ball}(\boldsymbol{x}, \upsilon))/2^d$ for all $\boldsymbol{x}$. Since $\mu(\mathrm{Ball}(\boldsymbol{x}, \upsilon)) \asymp \upsilon^d$, the conclusion follows. $\square$

We will repeatedly make use of the concentration of latent positions.

Lemma A.4 (Latent Concentration)**.** *Let $\mathcal{X} = [0,1]^d$ and μ denote the normalized Lebesgue measure on $\mathcal{X}$. Suppose $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \sim \mathcal{X}$ are sampled iid and uniformly at random from μ. Fix some $T \subset \mathcal{X}$ such that $\mu(T) = \upsilon$. Then*

$$\mathbb{P}\left[\left|\upsilon n - |\{j \in [n] : \boldsymbol{x}_j \in T\}|\right| \geq 10\sqrt{\frac{\log n}{n}}\right] \leq n^{-10}.$$

Proof. Let X_i be an indicator variable that equals 1 if $\boldsymbol{x}_i \in T$ and zero otherwise. Notice the X_i are iid and bounded within $[0,1]$. Moreover, $\sum_i \mathbb{E}[X_i] = n\mu(T)$. Therefore by Hoeffding's inequality, for any $t > 0$,

$$\mathbb{P}[|\upsilon n - |\{j \in [n] : \boldsymbol{x}_j \in T\}|| \geq t] \leq 2\exp\left(\frac{-2t^2}{n}\right).$$

Setting $t = 10\sqrt{\frac{\log n}{n}}$ gives the result. $\square$

Corollary A.5. *Let $\epsilon > 0$. For $i \in [n]$ let $\epsilon_i' > 0$ be $\epsilon_i' := \sup\{v > 0 : \mu(\mathrm{Ball}(\boldsymbol{x}_i, v) \cap \mathcal{X}) \le \epsilon$. Let $T_i := \mathrm{Ball}(\boldsymbol{x}_i, \epsilon_i') \cap \mathcal{X}$. Let $u_i(S) := |\{j \in S : \boldsymbol{x}_j \in T_i\}|$ denote the number of members of S landing in T_i. Then*

$$\mathbb{P}\left[\forall i \in [n] : |u_i(S) - n_Q \epsilon| \le 10\sqrt{\frac{\log n}{n_Q}}\right] \ge 1 - n^{-8}.$$

Proof. Notice that each T_i has Lebesgue measure ϵ by definition. Therefore $\mathbb{E}[u_i(S)] = n_Q \epsilon$. Since S has n_Q members, setting $t = 10\sqrt{\frac{\log n}{n_Q}}$ in the statement of Lemma A.4 and taking a union bound over all $i \in [n]$ gives the conclusion. $\square$

We will decompose the error of Algorithm 1 into two parts.

Proposition A.6. *Let $\widehat{Q} \in [0, 1]^{n \times n}$ be the estimator from Algorithm 1. Then*

$$\frac{1}{n^2}\|Q - \widehat{Q}\|_F^2 \le \frac{2}{n^2}\sum_{i,j \in [n]}(J_S(i,j) + J_B(i,j)),$$

where J_S, J_B are the smoothing and Bernoulli errors respectively:

$$J_S(i,j) := \frac{1}{|T_i|^2 |T_j|^2}\left(\sum_{r \in T_i, s \in T_j} Q_{ij} - Q_{rs}\right)^2;$$

$$J_B(i,j) := \frac{1}{|T_i|^2 |T_j|^2}\left(\sum_{r \in T_i, s \in T_j} Q_{rs} - A_{Q;rs}\right)^2.$$

Controlling the Bernoulli errors is relatively straightforward.

Proposition A.7. *Let h be the bandwidth of Algorithm 1. The Bernoulli error is at most $O(\frac{\log n}{m})$ with probability $\ge 1 - n^{-8}$, where $m = h^2 n_Q^2$.*

Proof. Fix $i, j \in [n]$. We will bound the maximum Bernoulli error $J_S(i,j)$ over i, j, which suffices to bound the average. Let $m = |T_i|\,|T_j|$. We want to bound:

$$\left|\frac{1}{|T_i|\,|T_j|}\sum_{r \in T_i, s \in T_j}(Q_{rs} - A_{Q;rs})\right|^2.$$

Notice each summand is bounded within $\pm\frac{1}{m}$. Bernstein's inequality gives:

$$\mathbb{P}\left[\left(\frac{1}{|T_i|\,|T_j|}\sum_{r \in T_i, s \in T_j}Q_{rs} - A_{Q;rs}\right)^2 \ge t^2\right] \le 2\exp(-0.5t^2 m).$$

Setting $t = C\sqrt{\frac{\log n}{m}}$ for large enough $C = O(1)$, a union bound tells us that with probability $\ge 1 - n^{-8}$, the Bernoulli error is bounded by t^2. $\square$

Corollary A.8. *The Bernoulli error is at most $O(\sqrt{\frac{\log n_Q}{n_Q}})$ with probability $\ge 1 - n_Q^{-4}$.*

The rest of this section is devoted to bounded the smoothing errors $J_S(i,j)$.

A.2.1 Latent Distance to Graph Distance

We claim that if nodes are close in the latent space then they are close in graph distance.

Proposition A.9. *Suppose that $\|\boldsymbol{x}_i - \boldsymbol{x}_r\| \leq \epsilon$ and Q is β-smooth. Then $d_Q(i,r) \leq C_\beta^2 n^3 \epsilon^{2(\beta \wedge 1)}$.*

Proof. We the use smoothness of Q. By definition there exists $C_\beta > 0$ such that $Q_{ki} - Q_{kr} \leq C_\beta \|\boldsymbol{x}_i - \boldsymbol{x}_r\|^{\beta \wedge 1}$. Therefore,

$$
\begin{aligned}
d_Q(i,r) &= \sum_{\ell \neq i,r} \left| (Q^2)_{\ell i} - (Q^2)_{\ell r} \right|^2 \\
&= \sum_{\ell \neq i,r} \left(\sum_{k \in [n]} Q_{\ell k}(Q_{ki} - Q_{kr}) \right)^2 \\
&\leq \sum_{\ell \neq i,r} \sum_{k \in [n]} Q_{\ell k}^2 C_\beta^2 \epsilon^{2(\beta \wedge 1)} \\
&\leq n^3 C_\beta^2 \epsilon^{2(\beta \wedge 1)}.
\end{aligned}
$$
$\square$

We can now bound the minimum sizes of the neighborhoods using the concentration of latent positions and the smoothness of the graphon.

Lemma A.10 (Vershynin (2018)). *The volume of a ball of radius $r > 0$ in $\mathbb{R}^d$ is $\frac{\sqrt{\pi}^d}{\Gamma(d/2+1)} r^d$, where $\Gamma(\cdot)$ is the Γ function.*

Proposition A.11. *Let $C_d = (\Gamma(\frac{d}{2}+1))^{1/d}$. Let C_0, C' be constants. If $\upsilon_n \geq C \cdot C_d(\sqrt{\frac{\log n}{n_Q}})^{1/d}$ for large enough constant $C > 0$, and $g_n = C_0 C_\beta^2 n^2 (\upsilon_n)^{2(\beta \wedge 1)}$, then with probability $\geq 1 - n^{-6}$ for all $i \in [n]$ the neighborhood size is $|\{r : d_Q(i,r) \leq g_n\}| \geq C' n_Q \sqrt{\frac{\log n}{n_Q}}$.*

Proof. Fix $i \in [n]$ and $\upsilon_n > 0$. Let ϵ_i denote the Lebesgue measure of $\mathrm{Ball}(\boldsymbol{x}_i, \upsilon_n) \cap \mathcal{X}$. By Lemma A.3 and Lemma A.10, for all i, $\epsilon_i \geq (\frac{\sqrt{\pi}\upsilon_n}{2C_d})^d = (\frac{0.5\sqrt{\pi}\upsilon_n}{C_d})^d$. Let $\epsilon = \min_{i \in [n]} \epsilon_i$.

By Corollary A.5, with probability $\geq 1 - n^{-8}$, there are $n_Q \epsilon - C\sqrt{\frac{\log n}{n_Q}}$ members j of S such that $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq \upsilon_n$. A union bound over i gives the result simultaneously for all i with probability $\geq 1 - n^{-6}$.

From Proposition A.9, it follows that for all $i \in [n]$,

$$
\left| \{r \in S : d_Q(i,r) \leq C_\beta^2 n^2 (2\upsilon_n')^{2(\beta \wedge 1)}\} \right| \geq n_Q \epsilon - 10\sqrt{\frac{\log n}{n_Q}}.
$$

Choosing $\upsilon_n \geq C \cdot C_d(\frac{\log n}{n_Q})^{\frac{1}{2d}}$ for large enough $C > 0$ gives the conclusion. $\square$

A.2.2 Graph Distance Concentration

Next, we show that the empirical graph distance concentrates to the population distance.

Proposition A.12. *For any arbitrary symmetric $P \in [0,1]^{n \times n}$, we have, for all i, j simultaneously with probability at least $\geq 1 - O(n^{-8})$, that*

$$
|d_{A_P}(i,j) - d_P(i,j)| \leq O(n^2 \log n) + O(n^{2.5}\sqrt{\log n}).
$$

Proof. Fix i, j. Let $C_{ij} := (A_P^2)_{ij}$. By Mao et al. (2021) A.1, we have $C_{ij} = (P^2)_{ij} + t_{ij}$ for an error term t_{ij} such that $\mathbb{P}[\forall i, j : |t_{ij}| \leq 10\sqrt{n \log n}] \geq 1 - n^{-10}$. Then,

$$
\begin{aligned}
|d_{A_P}(i, j) - d_P(i, j)| &= \left| \sum_{\ell \neq i, j} \left((C_{i\ell} - C_{j\ell})^2 - ((P^2)_{i\ell} - (P^2)_{j\ell})^2 \right) \right| \\
&= \sum_{\ell \neq i, j} \left| (t_{i\ell} + t_{j\ell})^2 + 2(t_{i\ell} + t_{j\ell})((P^2)_{i\ell} - (P^2)_{j\ell}) \right| \\
&\leq O(n^2 \log n) + O\left(\sqrt{n \log n} \sum_{\ell \neq i, j} \left((P^2)_{i\ell} - (P^2)_{j\ell} \right) \right).
\end{aligned}
$$

Finally, notice that all entries of P^2 are of size $O(n)$, so the conclusion follows. $\qquad\square$

Finally, we will show that taking the restriction of the graph distance T_i^P to nodes in $S \subset [n]$ does not incur too much error.

Proposition A.13. *Suppose $n = n_Q^{O(1)}$. Then there exists a constant C such that if $h_0 \geq C\sqrt{\frac{\log n}{n_Q}} + \Delta_n$, then for all i, r simultaneously, $r \in T_i^{A_P}(h_0)$ implies $r \in T_i^P(h_2)$ for some $h_2 = O(h)$ with probability $\geq 1 - O(n^{-5})$.*

Proof. Let us introduce the notation $T_i^{P,S}(h)$ to denote the bottom h-quantile of $\{d_P(i, j) : j \in S\}$. In this notation, $T_i^{A_P}(h) := T_i^{A_P,S}(h)$ since we restrict the quantile to nodes in S. From Proposition A.12 and Assumption 2.2, we know that if $n \geq n_Q$ then for $h_0 \leq h_1 - 20\sqrt{\frac{\log n}{n}} - \Delta_n$ we have $T_i^{A_P}(h_0) \subseteq T_i^{P,S}(h_1)$ simultaneously for all $i \in [n]$ with probability $\geq 1 - O(n^{-8})$. It remains to compare $T_i^{P,S}(h_1)$ with $T_i^P(h_2)$ for some h_2.

We claim that if $h_2 \geq 30\sqrt{\frac{\log n_Q}{n_Q}}$ then $\mathbb{P}[\forall i \, |T_i^P \cap S| \geq h_2 n_Q - 3\sqrt{n_Q \log n_Q}] \geq 1 - O(n_Q^{-2})$. To see this, fix $i \in [n]$ and consider $T_i^P(h_2)$. For $j \in S$, let X_j be the indicator variable:

$$
X_j = \begin{cases} 1 & \text{if } j \in T_i^P(h_2), \\ 0 & \text{otherwise.} \end{cases}
$$

Notice that $\left|T_i^P(h_2) \cap S\right| = \sum_{j \in S} X_j$. By Hoeffding's inequality, since $\mathbb{E}[\sum_{j \in S} X_j] = h_2 n_Q$ and $|X_j - h_2| \leq 1$ for all j, we have

$$
\mathbb{P}\left[\left| |T_i^P(h_2) \cap S| - h_2 n_Q \right| \geq 3\sqrt{n_Q \log n} \right] \leq 2 \exp\left(-\frac{6 n_Q^2 \log n}{n_Q^2} \right) \leq 2n^{-6}.
$$

Taking a union bound over all $i \in [n]$ shows the claim holds with probability $\geq 1 - O(n^{-5})$. Therefore we set $h_1 \leq h_2 - 3.1\sqrt{\frac{\log n}{n_Q}}$ then $j \in T_i^{P,S}(h_1)$ implies $j \in T_i^P(h_2)$.

The conclusion follows with $C = 24\sqrt{\frac{\log n}{\log n_Q}} = O(1)$. $\qquad\square$

The ranking condition (Definition 1.3) then allows us to translate between graph distances in A_P and Q.

Corollary A.14. *Suppose that Definition 1.3 holds for (P, Q) at $h_n = c\sqrt{\frac{\log n_Q}{n_Q}} + \Delta_n$, for large enough constant $c > 0$. Suppose $n_Q \leq n \leq n_Q^{O(1)}$. Then for $h > h_n$ and $r \in T_i^{A_P}(h)$, it follows that $r \in T_i^Q(h_3)$ for some $h_3 = O(h)$. The statement holds simultaneously for all i, r with probability $\geq 1 - O(n^{-5})$.*

A.2.3 Control of Smoothing Error

We will decompose smoothing error into a sum of two terms called $E_{S,1}$ and $E_{S,2}$. The control of $E_{S,1}$ is relatively straightforward.

Lemma A.15. *The total smoothing error can be bounded with two terms:*

$$\frac{2}{n^2} \sum_{i,j \in [n]} J_S(i,j) \le E_{S,1} + E_{S,2},$$

where

$$E_{S,1} := \frac{C}{n} \max_{j \in [n], s \in T_j} \|Q(e_j - e_s)\|_2^2;$$

$$E_{S,2} := \frac{4}{n^2} \sum_{i \in [n]} \frac{1}{|T_i|} \mathbb{E}\left[\sum_{r \in T_i} \sum_{j \in [n]} \sum_{s \in T_j} (Q_{rj} - Q_{rs})^2. \right]$$

Proof. Note that

$$
\begin{aligned}
\frac{2}{n^2} \sum_{i,j \in [n]} J_S(i,j) &= \frac{2}{n^2} \sum_{i,j \in [n]} \frac{1}{|T_i|^2 \, |T_j|^2} \mathbb{E}\left[\left(\sum_{r \in T_i, s \in T_j} Q_{ij} - Q_{rs} \right)^2 \right] \\
&\le \frac{2}{n} \sum_{i \in [n]} \frac{1}{n\,|T_i|} \sum_{j \in [n]} \frac{2}{|T_j|} \mathbb{E}\left[\sum_{r \in T_i, s \in T_j} (Q_{ij} - Q_{rj})^2 + (Q_{rj} - Q_{rs})^2 \right] \\
&= \frac{4}{n} \sum_{i \in [n]} \frac{1}{n\,|T_i|} \mathbb{E}\left[\sum_j \frac{1}{|T_j|} \left(\sum_{r \in T_i} (Q_{ij} - Q_{rj})^2 + \sum_{r \in T_i} \sum_{s \in T_j} (Q_{rj} - Q_{rs})^2 \right) \right].
\end{aligned}
$$

The second inner summand is precise $E_{S,2}$. For $E_{S,1}$, notice that $|T_i| = |T_j| = h(n_Q - 1)$ by definition. Therefore

$$\sum_j \frac{1}{|T_j|} \sum_{r \in T_i} (Q_{ij} - Q_{rj})^2 = \frac{1}{h(n_Q - 1)} \sum_{r \in T_i} \sum_j (Q_{ij} - Q_{rj})^2 \le 2 \max_{r \in T_i} \|(e_i - e_r)^T Q\|_2^2. \quad \square$$

We can now bound $E_{S,1}$ in terms of graph distances.

Lemma A.16. *The smoothing error term $E_{S,1}$ can be bounded as follows:*

$$E_{S,1} \le \frac{2}{n} \max_{i \in [n], r \in T_i} \sqrt{d_Q(i,r)} + \frac{2c}{\sqrt{n}}$$

for some constant $c > 0$.

Proof. Fix $i \in [n]$ and $r \in T_i$. We have

$$
\begin{aligned}
\|Q(e_i - e_r)\|_2^2 &\le \|e_i - e_r\|_2 \|Q^T Q(e_i - e_r)\|_2 \\
&\le 2\|Q^2(e_i - e_r)\|_2.
\end{aligned}
$$

Now we will pass to graph distances. Let $e_{ab} := ((Q^2)_{aa} - (Q^2)_{ab})^2$ for $a, b \in [n]$. Notice that $\|Q^2(e_i - e_r)\|_2 = \sqrt{d_Q(i,r) + e_{ir} + e_{ri}}$. Moreover, $\sqrt{e_{ir} + e_{ri}} \le 2\sqrt{n}$ since the entries of Q^2 are individually bounded by $O(n)$. The conclusion follows. $\square$

Proposition A.17. *Suppose $\Delta_n = O(\sqrt{\frac{\log n}{n_Q}})$. Let C_d be the constant of Proposition A.11. Then if the bandwidth of Algorithm 1 is $h_n = C\sqrt{\frac{\log n}{n_Q}}$, for a constant $C = O(1)$, then the smoothing error $E_{S,1}$ is at most*

$$E_{S,1} \le C_2 C_d^{\beta \wedge 1} \left(\sqrt{\frac{\log n_Q}{n_Q}} \right)^{\frac{\beta \wedge 1}{d}}$$

for some $C_2 = O(1)$, with probability $\ge 1 - O(n^{-6})$.

Proof. Fix $i \in [n]$ and $r \in T_i^{A_P}(h_n)$. By Corollary A.14, if $h_n \geq C\sqrt{\frac{\log n}{n_Q}} + \Delta_n$ for a large enough constant $C > 0$, then there exists constant $C_2 > 0$ such that the following holds. With probability $\geq 1 - O(n^{-5})$, for all $i \in [n]$ and $r \in S$, $r \in T_i^Q(C_2 h_n)$,

Let $v_n = CC_d(\sqrt{\frac{\log n}{n_Q}})^{1/d}$ for C_d as in Proposition A.11 and $C > 0$ large enough constant. Then by Proposition A.11 the set of $s \in S$ such that $d_Q(i, r) \leq C_0 C_\beta^2 n^2 (v_n)^{2(\beta \wedge 1)}$ has size at least $C_2 n_Q \sqrt{\frac{\log n}{n_Q}}$. The statement holds for all i simultaneously with probability at least $1 - O(n^{-6})$. Therefore for all $i \in [n]$ and $r \in T_i^{A_P}(h_n)$, we have

$$d_Q(i, r) \leq C_0 C_\beta^2 n^2 (v_n)^{2(\beta \wedge 1)}$$

for some $C_0, C_\beta = O(1)$, with probability $\geq 1 - O(n^{-6})$. By Lemma A.16 we conclude that $E_{S,1}$ is bounded by $2v_n^{\beta \wedge 1} + \frac{2}{\sqrt{n}}$ with the same probability. $\square$

A.2.4 Control of the Second Smoothing Error

In this section, we show that the second smoothing error can be controlled in terms of $E_{S,1}$. We will need to track the following quantity.

Definition A.18 (Membership Count). *For $r \in S$ and bandwidth h, distance cutoff ϵ, the P-neighborhood count of r is $\psi_P(r) := \left|\{j \in [n] : r \in T_j^P(h, \epsilon)\}\right|$.*

In words, $\psi_P(r)$ counts the number of nodes $j \in [n]$ such that r lands in the neighborhood of j in our algorithm. While we know that $\left|T_j^P(h)\right| \leq h n_Q$ always, simply applying the pigeonhole principle gives too weak of a bound on membership counts. The base case is that there may be a "hub" node r lands in $T_j^P(h)$ for all j. We will show that there can be no such hub node.

Supposing that we can control of the empirical count ψ_{A_P}, we show that the smoothing error can be bounded.

Proposition A.19. *Let h_n be the bandwidth. Then*

$$E_{S,2} \leq O\left(\frac{E_{S,1}}{h_n n}\right) \cdot \max_{r \in [n]} (\psi_{A_P}(r)).$$

Proof. Rearranging terms, we have

$$E_{S,2} = \frac{1}{n^2 h^2 n_Q^2} \sum_{i,j \in [n], r \in T_i, s \in T_j} (Q_{rj} - Q_{rs})^2$$

$$= \frac{1}{n^2 h^2 n_Q^2} \sum_{r \in S} \psi_{A_P}(r) \sum_{j,s} (Q_{rj} - Q_{rs})^2$$

$$= \frac{n_Q}{n^2 h^2 n_Q^2} \mathbb{E}_{r \in S}\left[\psi_{A_P}(r) \sum_{j,s} (Q_{rj} - Q_{rs})^2\right]$$

$$= \frac{n_Q}{n^2 h^2 n_Q^2} \mathbb{E}_{r \in [n]}\left[\psi_{A_P}(r) \sum_{j,s} (Q_{rj} - Q_{rs})^2\right],$$

where the last step follows because j, s do not depend on i, r and because $S \subset [n]$ is chosen uniformly at random. Now, we will control the expectation by passing to a row sum, which is handled by $E_{S,1}$.

$$\mathbb{E}_{r \in [n]}\left[\psi_{A_P}(r) \sum_{j,s} (Q_{rj} - Q_{rs})^2\right] \leq \max_{r \in [n]} \left(\frac{\psi_{A_P}(r)}{n}\right) \cdot \sum_{j \in [n]} \sum_{s \in T_j} \|Q(e_j - e_s)\|_2^2.$$

Recall that $n^2 n_Q h_n E_{S,1} = \Omega\left(\sum_{j \in [n]} \sum_{s \in T_j} \|Q(e_j - e_s)\|_2^2\right)$. Hence we conclude that

$$E_{S,2} \leq O\left(\frac{E_{S,1}}{h_n n}\right) \cdot \max_{r \in [n]} (\psi_{A_P}(r)). \qquad \square$$

We therefore must show that $\max_{r \in S} \psi_{A_P}(r) \leq O(hn)$ with high probability.

Proposition A.20 (Population Version). *Suppose Assumption 2.2 holds for P with $c_1 < c_2$ and $\Delta_n = O((\frac{\log n}{n_Q})^{\frac{1}{2} \vee \frac{\alpha \wedge 1}{d}})$. Then if $h \leq C\sqrt{\frac{\log n}{n_Q}}$ for large enough constant $C > 0$, then we have $\max_{r \in S} \psi_P(r) \leq O(hn)$ with probability at least $1 - O(n_Q^{-8})$.*

Proof. Fix $r \in S$. Let C_d be as in Proposition A.11. Suppose that $\epsilon = C_d(C + 10)\sqrt{\frac{\log n_Q}{n_Q}}^{-1/d}$ and $h = C\sqrt{\frac{\log n_Q}{n_Q}}$. Now, we will claim that for large enough constant $c > 0$, that $\psi_P(r)$ is at most the size of $\text{Ball}(\boldsymbol{x}_r, c\epsilon) \cap \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$.

Suppose that $c > 0$ is a large enough constant. Now suppose that $\boldsymbol{x}_j$ is such that $\|\boldsymbol{x}_j - \boldsymbol{x}_r\| \geq c\epsilon$. We can lower bound the graph distance using Assumption 2.2, as:

$$d_P(r, j) := \|(e_r - e_j)^T P^2 (I - e_r e_r^T - e_j e_j^T)\|_2^2 \geq c_1 n^3 (c\epsilon)^{2(\alpha \wedge 1)} - n^3 \Delta_n.$$

On the other hand, suppose that $i \in S$ is such that $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq \epsilon$. Then $d_P(i, j) \leq C_\alpha^2 n^3 \epsilon^{2(\alpha \wedge 1)}$ by Proposition A.9. Therefore since $\epsilon = C_d(C + 10)\sqrt{\frac{\log n_Q}{n_Q}}^{-1/d}$ and $\Delta_n = O((\frac{\log n}{n_Q})^{\frac{1}{2} \vee \frac{\alpha \wedge 1}{d}})$, for large enough $c_1 > 0$ we have

$$d_P(r, j) := \|(e_r - e_j)^T P^2 (I - e_r e_r^T - e_j e_j^T)\|_2^2 \geq \frac{c_1}{2} n^3 (c\epsilon)^{2(\alpha \wedge 1)}.$$

Then, if we choose $c > 0$ such that $c^{2(\alpha \wedge 1)} > \frac{2 C_\alpha^2}{c_1}$, then $d_P(i, j) < d_P(r, j)$.

Next, from our choices of h, ϵ, by Corollary A.5, simultaneously for all $i \in [n]$ there are at least hn_Q nodes in S that have distance $\leq \epsilon$ in latent space from $\boldsymbol{x}_i$, with probablity $\geq 1 - O(n_Q^{-6})$.

Therefore, if $\boldsymbol{x}_r \notin \text{Ball}(\boldsymbol{x}_j, c\epsilon) \cap \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ then $r \notin T_j^P(h)$. This implies that $\psi_P(r) \leq |\{\text{Ball}(\boldsymbol{x}_r, 2c\epsilon) \cap \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}\}|$. We can bound the size of this ball with Lemma A.4. Notice the Lebesgue measure of $\text{Ball}(\boldsymbol{x}_r, 2c\epsilon) \cap [0, 1]$ is at most $(\frac{4c\epsilon}{C_d})^d$. Therefore, since $\boldsymbol{x}_i$ are chosen iid from the Lebesgue measure on $\mathcal{X}$, with probability at least $\geq 1 - O(n_Q^{-10})$, we have

$$\frac{1}{n} |\text{Ball}(\boldsymbol{x}_r, 2c\epsilon) \cap \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}| \leq 2c\epsilon + 10\sqrt{\frac{\log n}{n}}.$$

The right-hand side is bounded by $O(h)$ if $n \geq n_Q$. Taking a union bound over all $r \in S$ gives the conclusion. $\square$

We conclude with the desired upper bound.

Proposition A.21 (Bound on $\psi_{A_P}(r)$). *Suppose Assumption 2.2 holds for P with $c_1 < c_2$ and $\Delta_n = O((\frac{\log n}{n_Q})^{\frac{1}{2} \vee \frac{\alpha \wedge 1}{d}})$. Then if $h \leq C_0\sqrt{\frac{\log n_Q}{n_Q}}$ for small enough constant C_0, then we have $\max_{r \in S} \psi_{A_P}(r) \leq O(hn)$ with probability at least $1 - O(n_Q^{-8})$.*

Proof. By Proposition A.12, with probability at least $1 - O(n_Q^{-8})$, we have for all $r \in S, j \in [n]$ simultaneously that

$$d_{A_P}(r, j) \geq d_P(r, j) - O(n^{2.5}\sqrt{\log n})$$
$$\geq (1 - O(\frac{1}{\sqrt{n}})) d_P(r, j).$$

Similarly, $d_{A_P}(r, j) \leq (1 + O(\frac{1}{\sqrt{n}})) d_P(r, j)$. We conclude that $\psi_{A_P}(r) \leq 2\psi_P(r) = O(hn)$ with probability $\geq 1 - O(n_Q^{-8})$. $\square$

A.2.5 Overall Error

We can bound $C_d := \Gamma(\frac{d}{2} + 1)^{1/d}$ with the elementary inequality.

Lemma A.22. *Let $C_d := \Gamma(\frac{d}{2} + 1)^{1/d}$. Then $C_d \leq \sqrt{d/2}$.*

Proof of Theorem 2.3. By Proposition A.21 and Prop A.19, we have that $E_{S,1} \leq O(E_{S,1})$ with probability at least $1 - O(n_Q^{-8})$. Therefore by Proposition A.17,

$$\mathbb{P}\left[E_{S,1} + E_{S,2} \leq O\left(C_d^{\beta \wedge 1}\left(\frac{\log n}{n_Q}\right)^{\frac{\beta \wedge 1}{2d}}\right)\right] \geq 1 - O(n_Q^{-6}).$$

By Lemma A.22, $C_d \leq \sqrt{d/2}$. Finally, by Corollary A.8, the Bernoulli error is bounded by $O(\sqrt{\frac{\log n_Q}{n_Q}})$ with probability $\geq 1 - O(n_Q^{-4})$. Applying a union bound over the two kinds of error and Lemma A.15 gives the result. $\square$

A.3 Proof of Theorem 3.2

Recall the Gilbert-Varshamov code (Guruswami et al., 2019).

Theorem A.23 (Gilbert-Varshamov). *Let $q \geq 2$ be a prime power. For $0 < \epsilon < \frac{q-1}{q}$ there exists an ϵ-balanced code $C \subset \mathbb{F}_q^n$ with rate $\Omega(\epsilon^2 n)$.*

We will use the following version of Fano's inequality.

Theorem A.24 (Generalized Fano Method, Yu (1997)). *Let $\mathcal{P}$ be a family of probability measures, $(\mathcal{D}, d)$ a pseudo-metric space, and $\theta : \mathcal{P} \to \mathcal{D}$ a map that extracts the parameters of interest. For a distinguished $P \in \mathcal{P}$, let $X \sim P$ be the data and $\widehat{\theta} := \widehat{\theta}(X)$ be an estimator for $\theta(P)$.*

Let $r \geq 2$ and $\mathcal{P}_r \subset \mathcal{P}$ be a finite hypothesis class of size r. Let $\alpha_r, \beta_r > 0$ be such that for all $i \neq j$, and all $P_i, P_j \in \mathcal{P}_r$,

$$d(\theta(P_i), \theta(P_j)) \geq \alpha_r;$$
$$KL(P_i, P_j) \leq \beta_r.$$

Then

$$\max_{j \in [r]} \mathbb{E}_{P_j}[d(\widehat{\theta}(X), \theta(P_j))] \geq \frac{\alpha_r}{2}\left(1 - \frac{\beta_r + \log 2}{\log r}\right).$$

Definition A.25 (Relative Hamming Distance). *For $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^m$, we define their* relative Hamming distance *as follows:*

$$d_H(\boldsymbol{x}, \boldsymbol{y}) := \frac{1}{m}\left|\{i \in [m] : x_i \neq y_i\}\right|.$$

We will need the following construction of coupled codes.

Proposition A.26. *Let $m_P, m_Q \geq 2$ and m_Q divide m_P. There exists a code $C \subset \{0,1\}^{m_P}$ and a projection map $\Pi : \{0,1\}^{m_P} \to \{0,1\}^{m_Q}$ such that if $C' = \{\Pi(w) : w \in C\}$ then C' is a code with relative Hamming distance $\Omega(1)$. Moreover, $|C| = |C'| \geq 2^{0.1 m_Q}$*

Throughout the proof, we will identify the community assignment function $z : [n] \to [k]$ of an SBM (Definition 3.1) with the matrix $Z \in \{0,1\}^{n \times k}$ where $Z_{ij} = 1$ if and only if $z(i) = j$.

Proof. Begin with a Gilbert-Varshamov code $B \subset \{0,1\}^{m_Q}$ as in Theorem A.23. We can "lift" B to a code on $\{0,1\}^{m_P}$ simply by concatenation. If $w \in B$, then the corresponding $w' \in C$ is just $w' = (w, w, \ldots, w) \in \{0,1\}^{m_P}$. Let $\Pi : \{0,1\}^{m_P} \to \{0,1\}^{m_Q}$ simply select the first m_Q bits of a word. It is clear that $B = \{\Pi(w) : w \in C\}$, so we are done. $\square$

Now we are ready to prove Theorem 3.2.

Proof of Theorem 3.2. Let $m_P = \binom{n}{2}$, $m_Q = \binom{n_Q}{2}$, and $m = m_P$. Let $C \subset \{0,1\}^{m_P}$ be the code and $\Pi : \{0,1\}^{m_P} \to \{0,1\}^{m_Q}$ the projection map of Prop A.26. For each $w \in C$, we construct a pair of SBMs $P_w, Q_w \in \mathbb{R}^{n \times n}$ as follows.

Each P_w, Q_w is a stochastic block model with k_P, k_Q classes respectively. All the P_w share the same community structure, namely the lexicographic assignment where nodes $1, 2, \ldots, \frac{n}{k_P}$ are assigned to community 1, and so on. Similarly all the Q_w share the same lexicographic community structure with nodes $1, 2, \ldots, \frac{n}{k_Q}$ assigned to community 1, and so on. Therefore, there are fixed $Z_P \in \{0,1\}^{n \times k_P}, Z_Q \in \{0,1\}^{n \times k_Q}$, such that for all $w \in C$, there exist $A_w \in \mathbb{R}^{k_P \times k_P}, B_w \in \mathbb{R}^{k_Q \times k_Q}$ with

$$P_w = Z_P A_w Z_P^T,$$
$$Q_w = Z_Q B_w Z_Q^T.$$

The A_w, B_w are defined as follows. Let $i, j \in [k_P]$ and $i', j' \in [k_Q]$ be such that $i < j$ and $i' < j'$. Since $m_P = \binom{k_P}{2}$ and $m_Q = \binom{k_Q}{2}$, we can identify (i,j) and (i',j') with indices of $[m_P], [m_Q]$ respectively. Then for fixed $\delta_P, \delta_Q > 0$, the edge connectivity probabilities are

$$A_w(i,j) = A_w(j,i) := \begin{cases} 1/2 & \text{if } w_{ij} = 0, \\ 1/2 + \delta_P & \text{if } w_{ij} = 1; \end{cases}$$

$$B_w(i',j') = B_w(j',i') := \begin{cases} 1/2 & \text{if } \Pi(w)_{i'j'} = 0, \\ 1/2 + \delta_Q & \text{if } \Pi(w)_{i'j'} = 1. \end{cases}$$

We can set the diagonals of A_w, B_w to be $1/2$ as well.

Next, let $\mathcal{P}_r$ be a family of $r = |C|$ probability measures. For fixed $w \in C$, the corresponding measure is the distribution over data $(A_P, A_Q) \in \{0,1\}^{n \times n} \times \{0,1\}^{n_Q \times n_Q}$ sampled from $(P_w, Q_w[S,S])$. Note that we restrict S to be a fixed subset of $[n]$.

Next, let $\theta((P_w, Q_w)) := Q_w$, and let $d(\theta((P_w, Q_w)), \theta((P_{w'}, Q_{w'}))) := \frac{1}{n}\|Q_w - Q_{w'}\|_F$. We will show that for all $w, w' \in C$ with $w \neq w'$,

$$KL((P_w, Q_w), (P_{w'}, Q_{w'})) \leq KL(P_w, P_{w'}) + KL(Q_w, Q_{w'})$$
$$\leq O(n^2\delta_P^2 + n_Q^2\delta_Q^2)$$
$$=: \beta,$$
$$d((P_w, Q_w), (P_{w'}, Q_{w'})) := \frac{1}{n}\|Q_w - Q_{w'}\|_F$$
$$\geq \Omega(\delta_Q)$$
$$=: \alpha.$$

For the β claim, by Proposition 4.2 of Gao et al. (2015), if $\delta_P, \delta_Q \in (0, 1/4)$, we have

$$KL((P_w, Q_w), (P_{w'}, Q_{w'})) \leq KL(P_w, P_{w'}) + KL(Q_w, Q_{w'})$$
$$\lesssim \sum_{i,j\in[n]} (P_w(i,j) - P_{w'}(i,j))^2 + (Q_w(i,j) - Q_{w'}(i,j))^2.$$

Next, notice that $A_w(i,j) \neq A_{w'}(i,j)$ if and only if $w_{ij} \neq w'_{ij}$. Then for distinct $w, w' \in C$, we have $d_H(w, w') = \Omega(m_P)$, so

$$\sum_{i,j\in[n]} (P_w(i,j) - P_{w'}(i,j))^2 \lesssim \delta_P^2 \frac{n^2}{k_P^2} d_H(w, w') \binom{k_P}{2} \lesssim \delta_P^2 n^2.$$

The bound for Q_w is similar, so this verifies the β claim.

Similarly, for the α claim, notice that

$$\frac{1}{n}\|Q_w - Q_{w'}\|_F \gtrsim \frac{1}{k_Q}\sqrt{\delta_Q^2 d_H(\Pi(w), \Pi(w'))} \geq \frac{\delta_Q}{k_Q}\sqrt{d_H(\Pi(w), \Pi(w'))}.$$

By Prop A.26, $d_H(\Pi(w), \Pi(w')) = \Omega(m_Q) = \Omega(k_Q^2)$. Therefore $\alpha \leq \Omega(\delta_Q)$.

Next, by Prop A.27, the pair (P_w, Q_w) satisfies Definition 1.3 for all $w \in C$. Moreover, $\log |C| \geq 0.1 m_Q$ by Prop A.26.

Combining these results, by Theorem A.24 the overall lower bound is

$$\inf_{\widehat{Q}} \sup_w \frac{1}{n} \|\widehat{Q} - Q_w\|_F \gtrsim \alpha \left(1 - \frac{\beta + \log 2}{0.1\binom{k_Q}{2}}\right)$$

$$\geq \delta_Q \left(1 - \frac{30n^2 \delta_P^2}{k_Q^2} - \frac{30 n_Q^2 \delta_Q^2}{k_Q^2} - o(1)\right).$$

If we choose $\delta_P = 0.01(\frac{k_Q}{n})$ and $\delta_Q = 0.01 \frac{k_Q}{n_Q}$, then

$$\inf_{\widehat{Q}} \sup_w \frac{1}{n^2} \|\widehat{Q} - Q_w\|_F^2 \gtrsim \delta_Q^2$$

$$\gtrsim \frac{k_Q^2}{n_Q^2}.$$

Note that $k_Q \leq n_Q \leq n$, so $\delta_P, \delta_Q \in (0, 1/4)$ as desired. $\qquad\square$

Proposition A.27. *If $h_n = \min\{\frac{1}{k_P}, \frac{1}{k_Q}\}$ then for all $w \in C$, the pair (P_w, Q_w) satisfies Defn 1.3 at h_n.*

Proof. Consider $h = h_n$ and some node $i \in [n]$. Suppose that $j \neq i$ is in the same P_w-community as i, and that $\ell \neq i$ is in a different community. Then notice that $d_{P_w}(i, \ell) \geq d_{P_w}(i, j)$. Therefore $j \in T_i^{P_w}(h)$. Moreover, since $h \leq \frac{1}{k_P}$ and since the nodes of $S \subset [n]$ are equidistributed among the communities $1, 2, \ldots, k_P$, it follows that all members of $T_i^{P_w}(h)$ must belong to the same P_w-community as i.

Therefore, since the communities of Q_w are a coarsening of the communities of P_w, $j \in T_i^{Q_w}(\frac{1}{k_Q})$. Since $h \leq \frac{1}{k_Q}$, we are done. $\qquad\square$

A.4   SBM Clustering Error

In this section, we prove a minimax lower bound in the clustering regime for stochastic block models.

Theorem A.28. *Let Π denote the parameter space of pairs of SBMs (P, Q) on n nodes with k_P, k_Q communities respectively, such that the cluster structure of Q is a coarsening the cluster structure of P. Then*

$$\inf_{\widehat{Q}} \sup_{(P,Q) \in \Pi} \mathbb{E}[\frac{1}{n^2} \|\widehat{Q} - Q_i\|_F^2] \gtrsim \frac{\log k_Q}{n_Q}.$$

Proof. Let $H_m \in [0, 1]^{m \times m}$ be the Hadamard matrix of order m modified to replace all entries -1 with 0. If m is not a power of two, let H_m be defined as follows. Let $\ell = \lfloor \log_2 m \rfloor$ and let $H_{m'} \in \mathbb{R}^{m/2 \times m/2}$ contain $H_{2^{\ell-1}}$ on its top left block and zeroes elsewhere. Let

$$H_m = \begin{bmatrix} \mathbf{0}\mathbf{0}^T & H_{m'} \\ H_{m'}^T & \mathbf{0}\mathbf{0}^T \end{bmatrix}.$$

Notice that at most $\frac{7}{8}$ fraction of the entries of H_m are zero-padded, for any m. Now, let $B_P = \frac{1}{2}\mathbf{1}\mathbf{1}^T + \delta_P H_{k_P}$ and $B_Q = \frac{1}{2}\mathbf{1}\mathbf{1}^T + \delta_Q H_{k_Q}$ for some $\delta_P, \delta_Q \in (0, 1/4)$ to be chosen later.

We will define two families of matrices indexed by a finite set T. For $i \in T$, there are some $Z_i \in \{0, 1\}^{n \times k_P}$ and $Y_i \in \{0, 1\}^{n \times k_Q}$ to be specified later. Then

$$P_i = Z_i B_P Z_i^T,$$
$$Q_i = Y_i B_Q Y_i^T.$$

25

Now, we define Y_i as follows. Let Z_{n,k_Q} denote the set of balanced clusterings $z : [n] \to [k_Q]$ such that for all $i, j \in [k_Q]$, $\left| z^{-1}(\{i\}) \right| = \left| z^{-1}(\{j\}) \right|$. Let $Z \subset Z_{n,k_Q}$ select the z such that for all $j \leq k_Q/2$, $z^{-1}(j) = \{ \left\lfloor \frac{n(j-1)}{k_Q} \right\rfloor, \ldots, \left\lfloor \frac{nj}{k_Q} \right\rfloor \}$. Define a distance on Z as follows. For $y, y' \in Z$ let $Y, Y' \in \{0,1\}^{n \times k_Q}$ be the corresponding cluster matrices and let $d(y, y') := \frac{1}{n} \| Y B_Q Y^T - Y' B_Q (Y')^T \|_F$. By Theorem 2.2 of Gao et al. (2015), there exists a packing $T_0 \subset Z$ with respect to d such that for all $y, y' \in T_0$, we have $|\{ j : y'(j) \neq y(j) \}| \geq n/6$. Moreover, $\log |T_0| \geq \frac{1}{12} n \log k_Q$. Set $T = T_0$. For any $y_i \in T_0$, let $Y_i \in \{0,1\}^{n \times k_Q}$ be the corresponding cluster matrix and then $Q_i = Y_i B_Q Y_i^T$.

Now, to define Z_i, take $a \in [k_Q]$ and partition $y_i^{-1}(\{a\}) \subset [n]$ into k_P/k_Q equally sized communities in a uniformly random way. Number these $1, \ldots, \frac{k_P}{k_Q}$. In this way, we split community 1 of y_i into communities $1, \ldots, \frac{k_P}{k_Q}$ of z_i, and so on. Define Z_i to be the matrix corresponding to z_i. Notice that Z_i, Y_i are both balanced clusterings and that the clustering Y_i coarsens that of Z_i. Therefore (P_i, Q_i) are a pair of heterogeneous symmetric SBMs satisfying Definition 1.3 at $h = 1/k_Q$.

Next, we apply Fano's Inequality (Theorem A.24). Recall $\log |T| \geq \frac{1}{12} n \log k_Q$. Now, for $i, j \in T$ distinct, Prop 4.2 of Gao et al. (2015) gives

$$D_{KL}((P_i, Q_i), (P_j, Q_j)) \leq D_{KL}(P_i, P_j) + D_{KL}(Q_i, Q_j) \leq O(n^2 \delta_P^2 + n_Q^2 \delta_Q^2) =: \gamma_1.$$

Finally, we can bound:

$$\frac{1}{n^2} \| Q_i - Q_{i'} \|_F^2 \geq \frac{1}{n^2} \sum_{n/2 < j \leq n} \frac{n}{k_Q} \| (e_{y_i(j)} - e_{y_i'(j)}) B_Q \|^2$$

$$\geq c_0 \delta_Q^2 =: \gamma_2^2,$$

where $c_0 > 1$ is some constant. This follows because there are a constant fraction of $j > n/2$ such that $y_i(j) \neq y_i'(j)$, and any two rows of the Hadamard matrix differ on half their entries.

Now, set $\delta_Q^2 = \frac{n_Q \log k_Q}{10 n_Q^2}$ and $\delta_P^2 = \frac{\log k_Q}{10 n^2}$. Since $n \geq n_Q$, we conclude that

$$\inf_{\widehat{Q}} \sup_{i \in T} \mathbb{E} \left[\frac{1}{n^2} \| \widehat{Q} - Q_i \|_F^2 \right] \gtrsim \gamma_2^2 \left(1 - \frac{\gamma_1 + \log 2}{(1/12) n \log k_Q} \right)$$

$$\gtrsim \frac{\log k_Q}{n_Q}. \qquad \square$$

A.5 Proof of Proposition 3.4

We first argue that Algorithm 2 perfectly recovers Z_P, Z_Q with high probability.

Theorem A.29 (Implicit in Chen et al. (2014)). *Let $M = ZBZ^T$ be an (n, n_{min}, s)-HSBM. Then there exists absolute constant $C > 0$ such that the Algorithm of Chen et al. (2014) can recover Z, up to permutation, with zero error with probability $\geq 1 - O(n^{-8})$ if*

$$s \geq C \left(\frac{\sqrt{n}}{n_{\min}} \vee \frac{\log^2(n)}{\sqrt{n_{\min}}} \right).$$

Proof. The algorithm of Chen et al. (2014) returns a matrix $Y \in \{0,1\}^{n \times n}$ such that $Y_{ij} = 1$ if and only if i, j are in the same community, with probability $\geq 1 - O(n^{-8})$. Therefore, to construct a clustering from Y, simply assign the cluster of node 1 to all $j \in [n]$ such that $Y_{1j} = 1$, and so on. This returns the true $Z \in \{0,1\}^{n \times k}$ up to permutation with probability $\geq 1 - O(n^{-8})$. Note that k is correctly chosen because Y is equal to a block-diagonal matrix of ones up to permutation, with k blocks. $\qquad \square$

Theorem A.29 implies the following.

Proposition A.30. *Let $\widehat{Z}_P, \widehat{Z}_Q$ be as in Algorithm 2. Let s_P, s_Q be the signal to noise ratios of P, Q respectively. If s_P, s_Q satisfy the conditions of Theorem A.29 with respect to $(n, n_{\min}^{(P)}$ and $(n_Q, n_{\min}^{(Q)})$ respectively, then then with probability $\geq 1 - O(n_Q^{-8})$, there are permutation matrices $U_P \in \{0,1\}^{k_P \times k_P}, U_Q \in \{0,1\}^{k_Q \times k_Q}$ such that $\widehat{Z}_P = Z_P U_P$ and $\widehat{Z}_Q = Z_Q U_Q$.*

Next, we want to recover the clustering of Q on all n nodes, not just the n_Q nodes that we observe in A_Q. This is given by the following.

Proposition A.31. *1. If $h_n = 1/k_P$ and $k_Q \leq k_P$ then there exists a unique $\Pi \in \{0,1\}^{k_P \times k_Q}$ such that $Z_P \Pi$ contains the Q-clustering of all nodes in $[n]$. Let $\widetilde{Z}_Q := Z_P \Pi$.*

2. Let $\widehat{\Pi}$ be as in Algorithm 2 and U_P, U_Q be as in Proposition A.30. Then with probability $1 - O(\frac{1}{n_Q})$, $Z_P U_P \widehat{\Pi} = \widetilde{Z}_Q U_Q$.

Proof. Part (1) follows immediately from the SBM structure of P, Q and definition of Definition 1.3.

For Part (2), first notice that by Proposition A.30, with probability at least $1 - O(\frac{1}{n_Q})$, Algorithm 2 returns the true clusterings $\widehat{Z}_P = Z_P \in \{0,1\}^{n \times k_P}$ and $\widehat{Z}_Q = Z_Q \in \{0,1\}^{n_Q \times k_Q}$, up to permutation.

Now, Algorithm 2 simply takes unions of the clusters of Z_P to learn $\widehat{\Pi}$. Therefore, let $V : \mathbb{R}^n \to \mathbb{R}^{n_Q}$ project onto coordinates in S. Then $V \widehat{Z}_P \widehat{\Pi} = \widehat{Z}_Q$. Moreover, by Proposition A.30, $\widehat{Z}_P = Z_P U_P$ and $\widehat{Z}_Q = Z_Q U_Q$. Hence $V Z_P U_P \widehat{\Pi} = V \widetilde{Z}_Q U_Q$. To remove dependence on V, we need to argue that each Q-cluster has a reprensentatve in S.

Let E be the event that at least one Q-cluster has no representative in S. For a fixed $j \in [k_Q]$, cluster j has no representative in S with probability $\leq \left(1 - \frac{n_{\min}^{(Q)}}{n_Q}\right)^{n_Q}$. A union bound implies that

$$\mathbb{P}[E] \leq k_Q \left(1 - \frac{n_{\min}^{(Q)}}{n_Q}\right)^{n_Q} \leq k_Q \exp(-n_{\min}^{(Q)}) \leq O(n_Q^{-1}).$$

The last inequality holds because the condition of Theorem A.29 implies that $n_{\min}^{(Q)} \geq \Omega(\sqrt{n_Q})$ and $k_Q \leq \frac{n_Q}{n_{\min}^{(Q)}}$.

Finally, we proceed by conditioning on $\neg E$. Since $\widehat{Z}_P = Z_P U_P$, we know that for all $i \in S$, the unique $j_P \in [k_P]$ such that row i, column j_P of Z_P is nonzero contains its true P-community up to U_P. Similarly since $\widehat{Z}_Q = Z_Q U_Q$, the the unique $j_Q \in [k_Q]$ such that row i, column j_Q of Z_P is nonzero contains its true Q-community up to U_Q. Therefore the nodes in community j_P in P are in community j_Q in Q. So, up to permutations U_P and U_Q, we have $\Pi_{j_P, j_Q} = 1$. Since we condition on $\neg E$, each cluster of Q has at least one representative in S, so each columns of Π is nonzero. We conclude that $Z_P U_P \widehat{\Pi} = \widetilde{Z}_Q U_Q$ with probability at least $1 - O(n_Q^{-1})$. □

We are ready to give the overall error of Proposition 2.

Proposition A.32. *Suppose that $\widehat{Z}_P = Z_P, \widehat{\Pi} = \Pi$ in Algorithm 2. Then with probability $\geq 1 - O(\frac{1}{n_Q})$, Algorithm 2 returns a $\widehat{Q} \in [0,1]^{n \times n}$ such that*

$$\frac{1}{n^2} \|\widehat{Q} - Q\|_F^2 \lesssim \frac{k_Q^2 \log(n_{\min}^{(Q)})}{n_Q^2}.$$

Proof. By Proposition A.31, with probability $\geq 1 - O(\frac{1}{n_Q})$, we have $\widehat{Z}_P = Z_P U_P$, $\widehat{Z}_Q = Z_Q U_Q$, and $\widetilde{Z}_Q U_Q = Z_P U_P \widehat{\Pi}$. We proceed by conditioning on these events.

Next, let $W_Q \in \mathbb{R}^{k_Q \times k_Q}$ be the population version of $\widehat{W}_Q$ with $W_{Q;ii} = (\mathbf{1}^T Z_Q e_i)^{-1}$. Then since $\widehat{Z}_Q = Z_Q U_Q$ we have $\widehat{W}_Q = U_Q^T W_Q U_Q$. Hence

$$\widehat{Q} = (Z_P U_P \widehat{\Pi})(U_Q^T W_Q U_Q^T)(Z_Q U_Q)^T A_Q (Z_Q U_Q)(U_Q^T W_Q U_Q)(Z_P U_P \widehat{\Pi})^T$$
$$= \widetilde{Z}_Q (W_Q Z_Q^T A_Q Z_Q W_Q) \widetilde{Z}_Q^T.$$

Next, let $z_Q : [n] \to [k_Q]$ be the ground truth clustering map given by $\widetilde{Z}_Q \in \{0,1\}^{n \times k_Q}$. Let B_Q be defined analogously to $\widehat{B}_Q$ in Algorithm 2, but using $W_Q, Z_Q, \mathbb{E}[A_Q]$ in place of $\widehat{W}_Q, \widehat{Z}_Q, A_Q$. Let $m_i := W_{Q;ii}^{-1}$ be the the number of nodes in S belong to community i, and let n_i be the the number of nodes in $[n]$ belonging to community i of Q. Then the error of Algorithm 2 is then

$$\frac{1}{n^2}\|\widetilde{Z}_Q(\widehat{B}_Q - B_Q)\widetilde{Z}_Q^T\|_F^2 = \frac{1}{n^2}\left(\sum_{i,j\in[k_Q]} n_i n_j \left(\sum_{\substack{r\in z_Q^{-1}(\{i\})\cap S \\ s\in z_Q^{-1}(\{j\})\cap S}} \frac{B_{Q;ij} - A_{Q;rs}}{m_i m_j} \right)^2 \right)$$
$$= \frac{1}{n^2} \sum_{i,j\in[k_Q]} \frac{n_i n_j}{m_i^2 m_j^2} \left(\sum_{\substack{r\in z_Q^{-1}(\{i\})\cap S \\ s\in z_Q^{-1}(\{j\})\cap S}} B_{Q;ij} - A_{Q;rs} \right)^2.$$

Next, fix $i,j \in [k_Q]$ and let

$$X_{ij} = \sum_{\substack{r\in z_Q^{-1}(\{i\})\cap S \\ s\in z_Q^{-1}(\{j\})\cap S}} B_{Q;ij} - A_{Q;rs}.$$

If we condition on the clusterings of P, Q being correct then $\mathbb{E}[B_{Q;ij} - A_{Q;rs}] = 0$. Therefore by Hoeffding's inequality,

$$\mathbb{P}(X_{ij} \geq t^2) \leq 2\exp\left(-\frac{2t^2}{m_i m_j} \right).$$

Setting $t^2 = 10\log(m_i m_j)m_i m_j$ implies that with probability at least $1 - k_Q^2 \min_i(m_i)^{-20}$, that the overall error is

$$\frac{1}{n^2}\|\widehat{Q} - Q\|_F^2 \leq \frac{1}{n^2} \sum_{i,j\in[k_Q]} \frac{10\log(m_i m_j)n_i n_j}{m_i m_j}.$$

Finally, note that there exists a constant $c_0 > 0$ such that for all $i \in [k_Q]$, $m_i \geq c_0\sqrt{n_Q}$ and $n_i \geq c_0\sqrt{n}$, by assumption. Note that each m_i is a random quantity depending on the choice of $S \subset [n]$ such that $\mathbb{E}[m_i] = \frac{n_Q}{n}n_i$. Hoeffding's inequality and a union bound over all $i \in [k_Q]$ imply that that with probability at least $\geq 1 - O(n_Q^{-8})$ that $m_i \geq \mathbb{E}[m_i] - 10\sqrt{\log n_Q} \geq \Omega(\mathbb{E}[m_i])$. We conclude that

$$\frac{1}{n^2}\|\widehat{Q} - Q\|_F^2 \leq O\left(\frac{1}{n_Q^2} \sum_{i,j\in[k_Q]} 10\log(m_i m_j) \right)$$
$$\leq O\left(\frac{k_Q^2 \log(n_{\min}^{(Q)})}{n_Q} \right). \qquad \square$$

B   Additional Experiments

B.1   Ablation Experiments

In this section, we discuss additional experiments that quantify the dependence of our algorithms on all relevant parameters. Our experiments also include a new baseline adapted from the estimator of Levin et al. (2022).

Description of New Baseline. Levin et al. (2022) assumes that full edge data from both P and Q are observed, and $P = Q$. Since this is not true for us, we instead compute the following modified MLE based on their estimator from Section 3.3 of Levin et al. (2022).

$$\widetilde{Q}_{ij} = \begin{cases} \frac{w_P}{w_P + w_Q} A_{P;ij} + \frac{w_Q}{w_P + w_Q} A_{Q;ij} & \text{if } i, j \in S, \\ A_{P;ij} & \text{otherwise.} \end{cases}$$

Here w_P, w_Q are computed as in their paper, based on estimated sub-gamma parameters of the noise for A_P, A_Q. Akin to their adjacency spectral embedding, which assumes known rank of Q, we use Universal Singular Value Thresholding to obtain $\widehat{Q}$ from $\widetilde{Q}$ Chatterjee (2015).

Oracle with $p = 0.0$. In addition to testing the new baseline from Levin et al. (2022), we also test the Oracle baseline with $p = 0.0$. As noted in Section 4, this corresponds to the non-transfer setting where all edges from the target graph Q are observed. Note that in this case, the value of n_Q does not matter because edges incident to nodes outside of S never get flipped. The Oracle error for β-smooth graphons on d-dimensional latent variables will therefore be $O(n^{-\frac{2\beta}{2\beta+d}})$ Xu (2018), which is less than the error bound of Theorem 2.3. Indeed, we will find that the Oracle our transfer algorithms in the regimes where its theoretical upper bound is better than our theoretical upper bounds.

Next, we describe the experimental results.

Figure 3 tests Algorithm 1 for general latent variable models. The error (Theorem 2.3) depends on the smoothness β of the target graph, the number of observed target nodes n_Q, and the dimension of the latent variables d.

Figure 4 tests Algorithm 2 for Stochastic Block Models. The error (Proposition 3.4) depends on the number of communities k_Q in the target graph, and the number of observed target nodes n_Q. Note that Proposition 3.4 also depends logarithmically on the minimum community size of Q, but this is less significant.
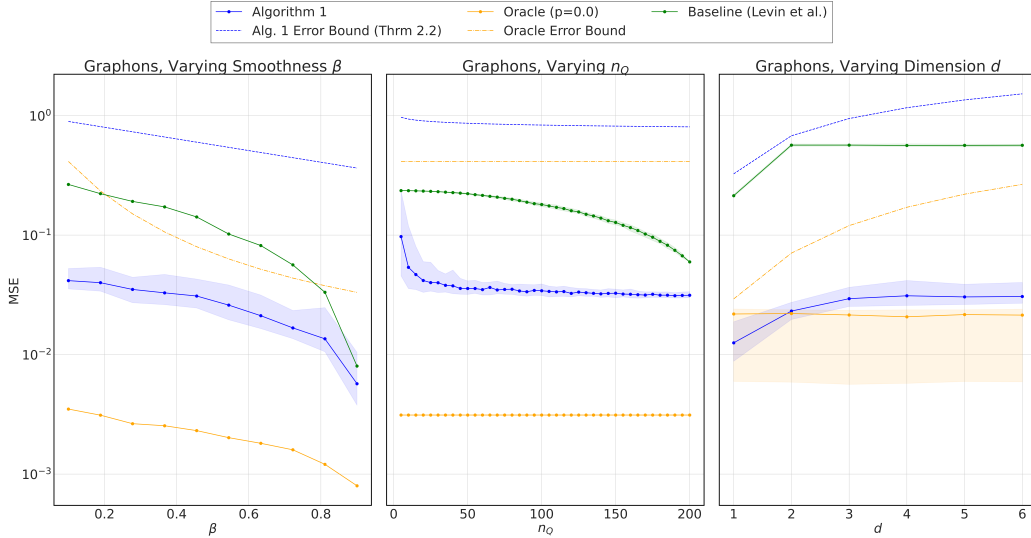
Figure 3: Testing parameters of Algorithm 1 (Transfer for Latent Variable Models). For most parameter settings, our method is better than the baseline and worse than the Oracle.
Left: Testing Hölder-smoothness of f_Q with $n = 200, n_Q = 25, d = 1$. All methods improve as $\beta \to 1$. Here $f_P(x, y) = \frac{x^\alpha + y^\alpha}{2}, f_Q(x, y) = \frac{x^\beta + y^\beta}{2}$ with $\alpha = 0.01$ and β varying.
Middle: Testing number of observed target nodes n_Q with $n = 200, d = 1$. Here $f_P(x, y) = \frac{x^\alpha + y^\alpha}{2}, f_Q(x, y) = \frac{x^\beta + y^\beta}{2}$ with $\alpha = 0.01, \beta = 0.1$. Note that the oracle does not depend on n_Q because it observes the full adjacency matrix $A_Q \in \{0, 1\}^{n \times n}$.
Right: Testing dimension d of latent positions $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in [0, 1]^d$ (i.i.d. Lebesgue) with $n = 200, n_Q = 25$. Here $f_P(\boldsymbol{x}, \boldsymbol{y}) = \exp(-6\|\boldsymbol{x} - \boldsymbol{y}\|_2)$ and $f_Q(\boldsymbol{x}, \boldsymbol{y}) = \exp(-|x_1 - y_1|)$.
Points are the median MSE across 50 trials, with with $[5, 95]$ percentile outcomes shaded.
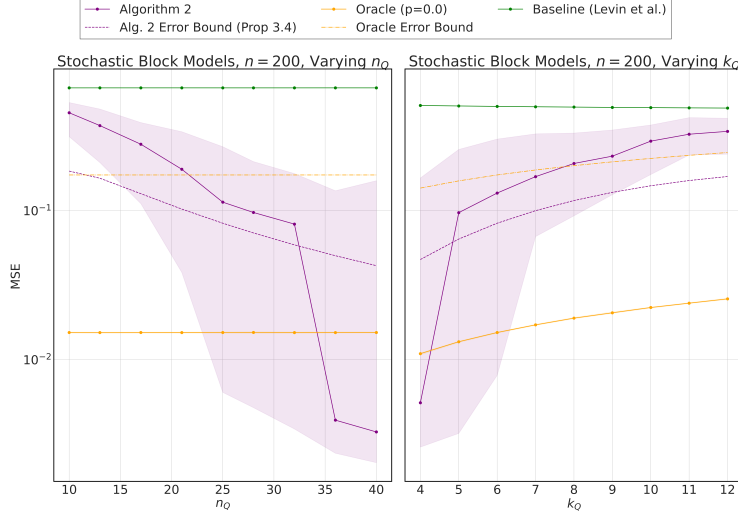
Figure 4: Testing parameters of Algorithm 2 (Transfer for SBMs). For most parameter settings, our method is better than the baseline and worse than the Oracle.
Left: $n = 200, k_P = 12, k_Q = 6$. Note that the oracle does not depend on n_Q because it observes the full adjacency matrix $A_Q \in \{0, 1\}^{n \times n}$.
Right: $n = 200, n_Q = 25, k_P = 2k_Q$.
For both experiments, the intra-community edge probabilities are $0.2, 0.9$ for P, Q respectively, while the inter-community edge probabilities are $0.1, 0.8$ respectively. Points are the median MSE across 50 trials, with with $[5, 95]$ percentile outcomes shaded.

Note that while we can plot theoretical guarantees for the mean squared error $\frac{1}{n^2} ||\widehat{Q} - Q||_F^2$ of both our algorithms' $\widehat{Q}$ and the oracle's $\widehat{Q}$, Levin et al. (2022) only give theoretical guarantees on the spectral norm $||\widehat{Q} - Q||_2$ for their estimator $\widehat{Q}$. Analyzing the stronger metric of mean-squared error would require different techniques than their paper.

B.2   Link Prediction Experiments

In this section, we present additional link prediction experiments on the EMAIL-EU and BIGG MODELS datasets. Unlike Section 4, we tune the sparsity estimate $\widehat{\rho} \in (0, 1)$ used in the Universal Singular Value Thresholding step of the Oracle baseline. In particular, we set $\widehat{\rho} \in (0, 1)$ to be the mean of the entries of the ground truth target matrix $Q \in [0, 1]^{n \times n}$. Note that this value is inaccessible to other algorithms since it requires knowing all the edges of Q.

Figures 7 and 8 show the performance of our Algorithms on the EMAIL-EU dataset, and Figures 5 and 6 for the BIGG MODELS dataset. As in the mean-squared error setting (Figure 2), we find that Algorithm 1 outperforms Algorithm 2, and that the Oracle baseline outperforms both for small p. Moreover, we find that the choice of source & target affects the performance of both of our algorithms. Hence Figure 7 shows better performance than Figure 8 for the same source but different targets, and Figure 5 shows better performance than Figure 6 for the same target but different sources.

C   Experimental Details

In this section, we give further details on the experiments of Section 4.

Compute Environment. We run all experiments on a personal Linux machine with 378GB of CPU/RAM. The total compute time across all results in the paper was less than 2 hours.

Functions for Figure 1. For the top row, the source is an $(n, 4)$-SBM with 0.8 on the diagonal and 0.2 on the off-diagonal of $B \in \mathbb{R}^{4 \times 4}$. The target is an $(n, 2)$-SBM with 0.9 on the diagonal and 0.1 on the off-diagonal of $B \in \mathbb{R}^{2 \times 2}$.
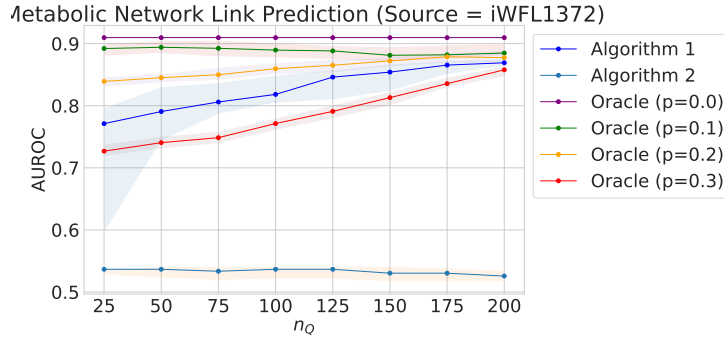
Figure 5: Link prediction results with the metabolic network of BiGG model iWFL1372 (*Escherichia coli W*) as the source and iJN1463 (*Pseudomonas putida*) the target. Shaded regions denote $[5, 95]$ percentile outcomes from 50 independent trials.
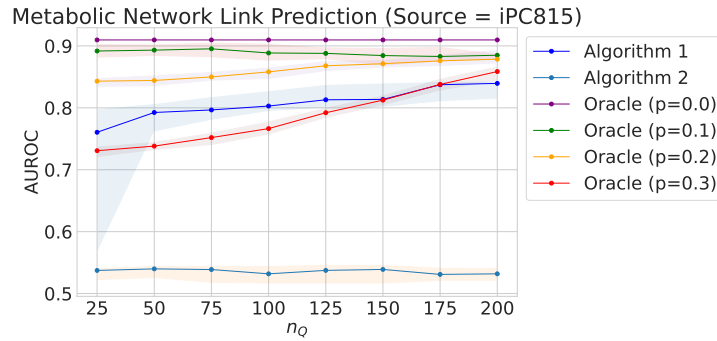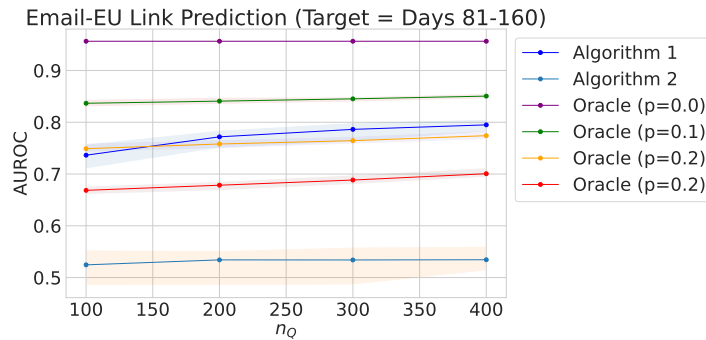

Figure 6: Link prediction results with the metabolic network of BiGG model iPC815 (*Yersinia pestis*) as the source and iJN1463 (*Pseudomonas putida*) the target. Shaded regions denote $[5, 95]$ percentile outcomes from 50 independent trials.

Figure 7: Link prediction results with Days 1-80 of EMAIL-EU as the source, and Days 81-160 as target. Shaded regions denote $[5, 95]$ percentile outcomes from 50 independent trials.

For the second and third rows, the source function is $Q(x, y) = \frac{1 + \sin(\pi(1 + 3(x+y-1)))}{2}$ (modified from Zhang et al. (2017)). The sources are $P(x, y) = 1 - Q(x, y)$ and $P(x, y) = Q(\phi(x), y)$, where $\phi(x) = 0.5 + |x - 0.5|$ if $x < 0.5$, and $0.5 - |x - 0.5|$ otherwise.

Metabolic Networks. We access metabolic models from King et al. (2016) at `http://bigg.ucsd.edu`. To construct a reasonable set of shared metabolites across the networks, we take the intersection of the node sets for the following BiGG models: iCHOv1, IJN1463, iMM1415, iPC815, iRC1080,
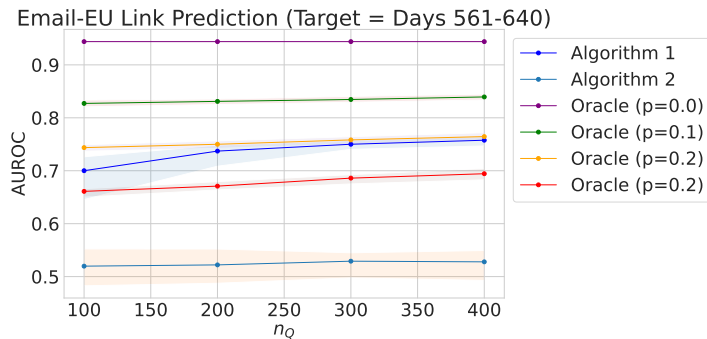
Figure 8: Link prediction results with Days 1-80 of EMAIL-EU as the source, and Days 561-640 as target. Shaded regions denote $[5, 95]$ percentile outcomes from 50 independent trials.

iSDY1059, iSFxv1172, iYL1228, iYS1720, and Recon3D. We obtain a set of $n = 251$ metabolites that are present in all of the listed models.

The resulting networks are undirected, unweighted graphs on 251 nodes. We construct the matrix $A_P \in \{0, 1\}^{n \times n}$ for species P by setting $A_{P;uv} = 1$ if and only if u and v co-occur in a metabolic reaction in the BiGG model for P.

EMAIL-EU. We use the "email-EU-core-temporal" dataset at `https://snap.stanford.edu/data/email-Eu-core-temporal.html`, as introduced in Paranjape et al. (2017). Note that we do not perform any node preprocessing, so we use all $n = 1005$ nodes present in the data, as opposed to Leskovec and Krevl (2014); Paranjape et al. (2017) who use only 986 nodes.

Data consist of triples (u, v, t) where u, v are anonymized individuals and $t > 0$ is a timestamp. We split the data into 10 bins based on equally spaced timestamp percentiles. For simplicity we refer to these time periods as consisting of 80 days each in Section 4, but technically there are 803 days total. The network at time period ℓ consists of an unweighted undirected graph with adjacency matrix entry $A_{uv} = 1$ if and only if (u, v, t) or (v, t, u) occurred in the data for an appropriate timestamp t.

Hyperparameters. We do not tune any hyperparameters. For Algorithm 1 we use the quantile cutoff of $h_n = \sqrt{\frac{\log n_Q}{n_Q}}$ in all experiments.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We discuss our main contributions in the abstract and introduction, and then develop them in the main body of the paper. Specifically, we point out Algorithm 1 and Algorithm 2, the upper bounds from Theorem 2.3 and Proposition 3.4, and the lower bound Theorem 3.2. Finally, Section 4 shows experimental results.

   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss the limitation of lacking a minimax lower bound for network estimation in d-dimensions after Theorem 2.3, the need for a different graph distance for graphs with sparsity $O(\frac{\log n}{n})$ in the Conclusion, and adapting to multiple source distributions in the Conclusion.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We give self-contained theoretical statements with definitions and assumptions in the main body, and full proofs for all claims in Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We give experimental details in Section 4, and give further details on our exact data preparation methods in Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We submit our code as a supplementary zip file in accordance with the NeurIPS code and data submission guidelines.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We describe our choices of source and target network, the network sizes and degrees, and the single hyperparameter (quantile cutoff) for Algorithm 1 in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: For each experiment we run 50 independent trials. We report ± 2 standard deviations for the mean-squared error in Table 1, and $[1, 99]$ percentile errors in Figure 2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: As described in Appendix C, we run all experiments on a personal Linux machine with 378GB of CPU/RAM. The total compute time across all results in the paper was less than 2 hours.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: We conform with the NeurIPS Code of Ethics with regards to potential harms caused by the research process, societal impact, and impact mitigation measures. Moreover, we have anonymized our code and manuscript.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

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

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not release data or models that have a high risk for misuse. We use only publicly available data from the academic literature.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We credit the authors of the BiGG metabolic models dataset (King et al., 2016; Norsigian et al., 2020) and the EMAIL-EU dataset (Leskovec and Krevl, 2014; Paranjape et al., 2017). Both datasets are publicly released for academic research.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We do not release new assets. See Appendix C for the exact details regarding our use of existing datasets and assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not perform any crowdsourcing or research on human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not perform any research on living subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.