Appendix to "Diffusion Source Identification on Networks with Statistical Confidence"

Quinlan Dawkins, Tianxi Li and Haifeng Xu

A Proofs

A.1 Proof of Proposition 1

Both the two claims are straightforward to show by using the definition. First, we have

$$\mathbb{E}_s \ell_{rc}(y, z) = 1 - \mathbb{E}\mathbb{I}(y = \zeta(z)) = 1 - \mathbb{P}_s(\zeta(Z) = y).$$

So minimizing the loss is equivalent to the MLE, which is equivalent to the rumor center in infinite regular trees.

Secondly, notice that that both y and $\zeta(Z)$ are n dimensional binary vectors. So

$$\|y - \zeta(Z)\|_2^2 = \sum_i I(y_i \neq \zeta(Z)_i)$$

which is the symmetric difference between the set $\{i : y_i = 1\}$ and $\{i : \zeta(Z)_i = 1\}$.

A.2 Proof of Theorem 1

Define $q^{\alpha}_{T,s^*} = \inf_t\{t : \mathbb{P}_{s^*}(T_{s^*}(\zeta(Z)) \geq t) \leq \alpha\}$. Notice that q^{α}_{T,s^*} can be seen as one generalized definition for the right quantile of the distribution of the random variable $T_{s^*}(\tilde{Y})$, where $\tilde{Y} := \zeta(Z)$ is a random infection status of the network generated by the diffusion process starting from s^*.

Now assume Y is a random infection status from the diffusion process from s^*. According to the definition of the p-value, we have

$$\begin{aligned}
\mathbb{P}_{s^*}\left(s^* \in S(Y)\right) &= \mathbb{P}_{s^*}\left(\psi_{s^*}(Y) > \alpha\right) \\
&= \mathbb{P}_{s^*}\left(\mathbb{P}_{s^*}\left(T_{s^*}(\tilde{Y}) \geq T_{s^*}(Y)\right) > \alpha\right) \\
&\geq \mathbb{P}_{s^*}\left(T_{s^*}(Y) < q^{\alpha}_{T,s^*}\right) \\
&= 1 - \mathbb{P}_{s^*}\left(T_{s^*}(Y) \geq q^{\alpha}_{T,s^*}\right).
\end{aligned}$$

Note that since s^* is the true source node, $T_{s^*}(Y)$ and $T_{s^*}(\tilde{Y})$ are following exactly the same distribution, thus

$$\mathbb{P}_{s^*}\left(s^* \in S(Y)\right) \geq 1 - \mathbb{P}_{s^*}\left(T_{s^*}(Y) \geq q^{\alpha}_{T,s^*}\right) \geq 1 - \alpha.$$

A.3 Proof of Theorem 2

Given a path generated by the diffusion process starting from v_0 **containing** u, denoted by

$$z = \{v_0, s_1, s_2, \cdots, s_{K-1}, u, s_{K+1}, \cdots, s_T\},$$

we match it to the path $f_u(Z)$ defined as

$$f_u(z) = \{u, v_0, s_1, \cdots, s_{K-1}, s_{K+1}, \cdots, s_T\}.$$

We start from the probability mass of z starting from v_0. By using the Markov property, we have

$$\begin{aligned}
p(z|v_0) =& \mathbb{P}(s_1|v_0)\mathbb{P}(s_2|v_0, s_1)\cdots\mathbb{P}(s_{K-1}|v_0, s_1, \cdots, s_{K-2}) \\
&\times \mathbb{P}(s_{K+1}|v_0, \cdots, u)\cdots\mathbb{P}(s_T|v_0, \cdots, s_{T-1}) \\
&\times \mathbb{P}(u|v_0, s_1, \cdots, s_{K-1})
\end{aligned} \tag{12}$$

In contrast, for the path $f_u(z)$, we have

$$\begin{aligned}\mathbb{P}(f_u(z)|u) =&\mathbb{P}(v_0|u)\mathbb{P}(s_1|u,v_0)\mathbb{P}(s_2|u,v_0,s_1)\cdots\mathbb{P}(s_{K-1}|u,v_0,s_1,\cdots,s_{K-2})\\ &\times\mathbb{P}(s_{K+1}|u,v_0,\cdots,s_{K-1})\cdots\mathbb{P}(s_T|u,v_0,\cdots,s_{T-1})\\ =&\mathbb{P}(s_1|u,v_0)\mathbb{P}(s_2|u,v_0,s_1)\cdots\mathbb{P}(s_{K-1}|u,v_0,s_1,\cdots,s_{K-2})\\ &\times\mathbb{P}(s_{K+1}|u,v_0,\cdots,s_{K-1})\cdots\mathbb{P}(s_T|u,v_0,\cdots,s_{T-1}).\end{aligned} \tag{13}$$

Notice that the conditional probability $\mathbb{P}(s_{k+1}|v_0,\cdots,u,s_{K+1},\cdots s_k), k>K$ only depends on the infection status before the kth infection and is invariant to the infection order. This property indicates that all terms after the $K+1$th (in the second rows) of (12) and (13) are equal.

Next, we compare the terms in the first line in each of (12) and (13). Notice that for each $k<K$, the term $\mathbb{P}(s_k|v_0,s_1,\cdots,s_{k-1})$ is identical for each available connections given the infected nodes $v_0,s_1,\cdots,s_{k-1}$ while the term $\mathbb{P}(s_k|u,v_0,s_1,\cdots,s_{k-1})$ is identical on all available edges given the infected nodes $u,v_0,s_1,\cdots,s_{k-1}$. The only difference in the two infected sets is on u. Since u has only one connection to v_0, at each point, the number of available infecting edges is one more in the former case. Therefore, we have

$$\mathbb{P}(s_k|u,v_0,s_1,\cdots,s_{k-1})=\frac{1}{1-\mathbb{P}(s_k|v_0,s_1,\cdots,s_{k-1})}\mathbb{P}(s_k|v_0,s_1,\cdots,s_{k-1}), k<K.$$

In addition, notice that in the third line of (12), there is one extra term that does not appear in (13). Combining the aforementioned three relations, we final obtain probability mass factor to be

$$\frac{\mathbb{P}(f_u(\pi)|S_0=u)}{\mathbb{P}(\pi|S_0=v_0)}=\frac{1}{(1-\mathbb{P}(s_1|v_0))(1-\mathbb{P}(s_2|v_0,s_1))\cdots(1-\mathbb{P}(s_{K-1}|v_0,s_1,\cdots,s_{K-2}))\mathbb{P}(u|v_0,s_1,\cdots,s_{K-1})} \tag{14}$$

Moreover, if z does not contain u, we set the ratio to be 0.

A.4 Proof of Theorem 3

Since Z_i's are a random sample from p_1, under the current assumption, by the strong law of large numbers, we have

$$\mathbb{P}\left(\hat{\eta}\to\mathbb{E}_1[\frac{g(\phi(Z))}{|\phi^{-1}(\phi(Z))|}\frac{p_2(\phi(Z))}{p_1(Z)}]\right)=1.$$

Notice that ϕ is a surjection. Therefore, the term $\mathbb{E}_1[\frac{g(\phi(Z))}{|\phi^{-1}(\phi(Z))|}\frac{p_2(\phi(Z))}{p_1(Z)}]$ can be rewritten as

$$\begin{aligned}
\mathbb{E}_1[\frac{g(\phi(Z))}{|\phi^{-1}(\phi(Z))|}\frac{p_2(\phi(Z))}{p_1(Z)}] &= \sum_{z\in\mathcal{C}_1}\frac{g(\phi(z))}{|\phi^{-1}(\phi(z))|}\frac{p_2(\phi(z))}{p_1(z)}p_1(z)\\
&= \sum_{z\in\mathcal{C}_1}\frac{g(\phi(z))}{|\phi^{-1}(\phi(z))|}p_2(\phi(z))\\
&= \sum_{z\in\mathcal{C}_1'}\frac{g(\phi(z))}{|\phi^{-1}(\phi(z))|}p_2(\phi(z)) + \sum_{z\in\mathcal{C}_1/\mathcal{C}_1'}\frac{g(\phi(z))}{|\phi^{-1}(\phi(z))|}p_2(\phi(z))\\
&= \sum_{z\in\mathcal{C}_1'}\frac{g(\phi(z))}{|\phi^{-1}(\phi(z))|}p_2(\phi(z))\\
&= \sum_{\tilde{z}\in\mathcal{C}_2}\sum_{z:\phi(z)=\tilde{z}}\frac{g(\phi(z))}{|\phi^{-1}(\phi(z))|}p_2(\phi(z))\\
&= \sum_{\tilde{z}\in\mathcal{C}_2}\sum_{z:\phi(z)=\tilde{z}}\frac{g(\tilde{z})}{|\phi^{-1}(\tilde{z})|}p_2(\tilde{z})\\
&= \sum_{\tilde{z}\in\mathcal{C}_2}\sum_{z\in\phi^{-1}(\tilde{z})}\frac{g(\tilde{z})}{|\phi^{-1}(\tilde{z})|}p_2(\tilde{z})\\
&= \sum_{\tilde{z}\in\mathcal{C}_2}|\phi^{-1}(\tilde{z})|\frac{g(\tilde{z})}{|\phi^{-1}(\tilde{z})|}p_2(\tilde{z})\\
&= \sum_{\tilde{z}\in\mathcal{C}_2}g(\tilde{z})p_2(\tilde{z})\\
&= \mathbb{E}_2[g(Z)].
\end{aligned}$$

A.5 Proof of Corollary 1

We will use $f_u(z)$ in place of ϕ to apply Theorem 3. The only remaining step is to find $|f_u(f_u^{-1}(z))|$. By the definition of f_u in Theorem 3, it is easy to see that the other $T-1$ nodes (except u and v_0) and their order uniquely determine to the mapped path. Therefore, $|f_u(f_u^{-1}(z))|$ would always be T in this situation.

A.6 Proof of Theorem 4

Notice that, given T, the probability mass function of a diffusion path only depends on the network A and the source node. Condition 1 of Definition 4 indicates that Z_π starts from v. Condition 2 and condition 3 of Definition 4 together indicate that Z_π is also a valid diffusion path. Condition 3, in particular, indicates that

$$p_u(Z) = p_v(Z_\pi).$$

Now define can define π^{-1} to be the inverse of π. For any $\tilde{Z}$ from v, for the same reason, $\tilde{Z}_{\pi^{-1}}$ is also a valid diffusion path starting from u. In particular, we have $Z = (Z_\pi)_{\pi^{-1}}$. Therefore, Z_π has the same sample space and probability mass function as the random diffusion path from v.

B Details for Isomorphism Identification in Section 4.2

Based on the properties in Proposition 2, Algorithm 2 finds all isomorphic pairs and the permutations in the network. Next, we provide a proof of Proposition 2.

Proof of Proposition 2. $d_u = d_v$ because π gives a 1-1 mapping from $N_1(u)$ to $N_1(v)$. Furthermore, we have $\pi(N_1(u)) = N_1(v)$. By condition 3 of Definition 4, we also have $D_1(u) = D_1(v)$.

For the last one, we can prove by contradiction. Suppose there exists a node w, such that $w \in N_2(u)$ but $w \notin N_2(v)$.

Algorithm 2 Identification of First-Order Isomorphic Pairs

Input: Graph $G = (V, E)$
Initialize $L = \emptyset$ to store the list of isomorphic node pairs
for every node $u \in V$ **do**
  Compute $N_{1-4}(u)$ as the set of all neighbors of u within 4 hops
    // $N_{1-4}(u)$ includes all nodes that are possible to be isomorphic to u
  for $v \in N_{1-4}(u)$ **do**
    if $d_v == d_u$ **then**
      Compute $D_1(u) = \{d_{u'} : u' \in N_1(u)\}$, the multi-set of degrees of nodes in $N_1(u)$
      Similarly, compute $D_1(v)$, the multi-set of degrees of all one-hop neighbors of v
      if $D_1(u) == D_1(v)$ **then**
        Compute $\tilde{N}_2(u) = N_2(u) - N_1(u) - N_1(v) - \{u, v\}$
          // $\tilde{N}_2(u)$ contains all (exactly) two-hop neighbors of u, but with all (exactly) one-hop neighbors of u, v removed
        Compute $\tilde{N}_2(v) = N_2(v) - N_1(u) - N_1(v) - \{u, v\}$
        if $\tilde{N}_2(u) == \tilde{N}_2(v)$ **then**
          Do exhaustive search to check whether u, v are isomorphic by enumerating all possible matchings of their neighbors under the constraints of $\tilde{N}_2(u)$ and the matching $D_1(u), D_1(v)$, and if so, add (u, v) to list L
            // Usually, not many pairs need to go through this step
        end if
      end if
    end if
  end for
end for

Since $\pi(N_1(u)) = N_1(v)$ while $\pi(w) = w$, so after applying the permutation to the network, we have $\pi(w)$ disconnected from $\pi(N_1(w))$. This contradicts condition 3 of Definition 4. □

C Loss Function Computation Acceleration for Surjective Importance Sampling

The calculation strategy for canonical discrepancy functions can also be further generalized to the weighted averaging scenario used for the single-degree nodes in Section 4.1. Specifically, there we need to calculate terms like

$$\begin{aligned}
\frac{1}{m}\sum_{i=m+1}^{2m}\ell\left(y, f_u(z_i)\right)\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T} &= -\frac{1}{m}\sum_{i=m+1}^{2m}\sum_{v:y_v=1}\mathbb{I}(v\in f_u(z_i))h(t_{f_u(z_i)}(v))\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T}\\
&= -\frac{1}{m}\sum_{v:y_v=1}\sum_{i=m+1}^{2m}\mathbb{I}(v\in f_u(z_i))h(t_{f_u(z_i)}(v))\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T}\\
&= -\frac{1}{m}\sum_{v:y_v=1}\sum_{i=m+1}^{2m}\mathbb{I}(v\in f_u(z_i))[\sum_{k=1}^{T}\mathbb{I}(t_{f_u(z_i)}(v)=k)h(k)]\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T}\\
&= -\frac{1}{m}\sum_{v:y_v=1}\sum_{i=m+1}^{2m}\sum_{k=1}^{T}\mathbb{I}(v\in f_u(z_i))\mathbb{I}(t_{f_u(z_i)}(v)=k)h(k)\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T}\\
&= -\frac{1}{m}\sum_{v:y_v=1}\sum_{k=1}^{T}h(k)[\sum_{i=m+1}^{2m}\mathbb{I}(t_{f_u(z_i)}(v)=k)\frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)}\frac{1}{T}].
\end{aligned}$$

Therefore, to use this strategy in Section 4.1, when general MC samples from v_0, in addition to caching M, we also want to cache the matrix adjusted by the factor

$$M_{v,k}^{(v_0 \to u)} = \sum_{i=m+1}^{2m} \mathbb{I}(t_{f_u(z_i)}(v) = k) \frac{\mathbb{P}\left(f_u(z_i)|u\right)}{\mathbb{P}\left(z_i|v_0\right)} \frac{1}{T}.$$

D Parallel Algorithm for Confidence Set Construction

As discussed in Section 3.3, our confidence set construction algorithm can be implemented in parallel, further boosting its speed. In the main paper, we only include the details and timing for the sequential version. The parallelized algorithm is described in Algorithm 3.

Algorithm 3 Parallel Confidence Set Construction

Input: MC sample number m, confidence level α, Network G, data y, discrepancy function ℓ
Compute $S = \{g_1, g_2, \cdots, g_M\}$, the isomorphic groups for infected nodes with degree at least 2.
for each $g \in S$ **do**
  Extend g by including all of its single-degree neighbor.
end for
for each infected isomorphic group $g \in S$ **in parallel do**
  Select any $s \in g$ with degree at least 2
  Generate $2m$ samples $z_i \in \mathcal{Z}, i = 1, \cdots, 2m$ from the T-step diffusion process from source s.
  Calculate the p-value for s following (6), (7), and (8)
  for each $v \in g$ that is isomorphic to s **do**
    Calculate $\hat{\psi}_v(y)$ according to Theorem 4.
  end for
  for each single-degree node $v \in g$ **do**
    Calculate the p-value $\hat{\psi}_v(y)$ according to the surjective importance sampling in Section 4.1.
  end for
end for
return the level $1 - \alpha$ confidence set:

$$\mathcal{C}_\alpha(y) = \{s \in V_I : \hat{\psi}_s(y) > \alpha\}.$$

As can be seen, Algorithm 3 needs MC sampling for only one node in each group, and the calculations for other nodes can be done using pooled MC methods. When additional cores are available, the for-loops in the algorithm can be further parallelized.

E Evaluation on 381 real-world networks

The data set from Ghasemian et al. (2020) contains 550 networks. We focus on 381 networks with more than 200 nodes for stable evaluation. The removed ones are either too small or have certain pathological structures to stable computation. The 381 networks are from six domains (71 biological, 110 economic, 9 informational, 105 social, 56 technological, and 30 transportation networks).

Though there are no real diffusion labels observed on these networks, we can generate a synthetic diffusion process based on our model. We generate a diffusion process with $T = \min(0.2N, 150)$. By doing this, we can evaluate the confidence set properties on these networks and the timing. The average coverage probability of the 90% confidence set and the relative size of $|\mathcal{C}|/T$ are shown in Table 5. The results match what we observed previously on the simulated networks, and the ADiT is more effective than the Euclidean loss, indicating the valid method on the real-world network. More importantly, we also evaluate the timing improvement based on the pooled MC methods. We calculate the improvement percentage of the pooled MC strategies (over the vanilla MC) on each network. The results are summarized in Figure 4. As can be seen, the pooled MC is very effective on economic networks and social networks, resulting in an average improvement of 40%. It is moderately effective on biological and informational networks with 10%-20% improvement. The technological networks

and transportation networks are suitable structures for the strategy. The economic, social, and biological networks are the three largest domains in the data set. These results demonstrate the pooled MC's potential as a general computational strategy.

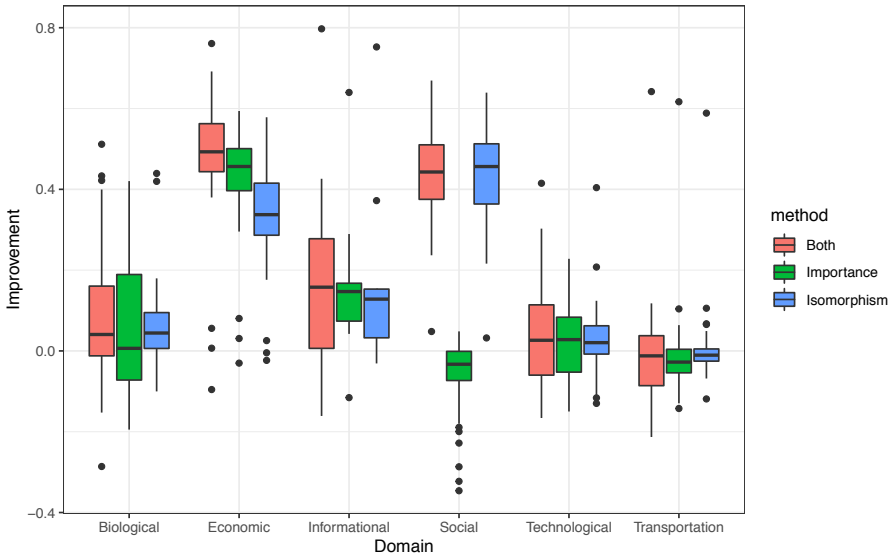

Figure 4: The timing improvement proportion by the pooled MC over the vanilla MC on 381 real-world networks.

Table 5: The average coverage rate of the 90% confidence sets and the relative size $|\mathcal{C}|/T$ on the 381 real-world networks.

	BIOLOGICAL	ECONOMIC	INFORMATIONAL	SOCIAL	TECHNOLOGICAL	TRANSPORTATION
EUCLIDEAN-90%	90.0%	90.1%	88.9%	89.8%	89.1%	90.9%
$\|\mathcal{C}\|/T$	0.60	0.41	0.66	0.44	0.57	0.38
ADiT-90%	90.2%	89.9%	87.6%	89.3%	89.3%	90.1%
$\|\mathcal{C}\|/T$	0.55	0.36	0.61	0.40	0.52	0.35