A DERIVATION OF EQUATIONS

A.1 MI Bound – Route Entropy Inequality

Here we derive the inequality in Eq. (4). Given node $\mathbf{w}_i$, assuming that there are $R_m^i = K!$ routes to the m-th combination of neighbors, i.e., $p(C_m^K|\mathbf{w}_i) = \sum_{r=1}^{R_m^i} p(C_m^{r,K}|\mathbf{w}_i)$. And we have following derivation:

$$
\begin{aligned}
&\sum_{m=1}^{|C^K|} p(C_m^K, \mathbf{w}_i) \log p(C_m^K|\mathbf{w}_i) \\
=&\sum_{m=1}^{|C^K|} p(\mathbf{w}_i) \left(\sum_{r=1}^{R_m^i} p(C_m^{r,K}|\mathbf{w}_i) \right) \log p(C_m^K|\mathbf{w}_i) \\
=&\sum_{m=1}^{|C^K|} p(\mathbf{w}_i) \left(\sum_{r=1}^{R_m^i} p(C_m^{r,K}|\mathbf{w}_i) \log p(C_m^K|\mathbf{w}_i) \right) \\
=&\sum_{m=1}^{|C^K|} p(\mathbf{w}_i) \left(\sum_{r=1}^{R_m^i} p(C_m^{r,K}|\mathbf{w}_i) \log \left(p(C_m^{r,K}|\mathbf{w}_i) + \sum_{j\neq r} p(C_m^{j,K}|\mathbf{w}_i) \right) \right) \\
\geq&\sum_{m=1}^{|C^K|} p(\mathbf{w}_i) \left(\sum_{r=1}^{R_m^i} p(C_m^{r,K}|\mathbf{w}_i) \log p(C_m^{r,K}|\mathbf{w}_i) \right) \\
=&\sum_{m=1}^{|C^K|} \sum_{r=1}^{R_m^i} p(C_m^{r,K}, \mathbf{w}_i) \log p(C_m^{r,K}|\mathbf{w}_i) \\
=&\sum_{r=1}^{R} p(C_m^{r,K}, \mathbf{w}_i) \log p(C_m^{r,K}|\mathbf{w}_i),
\end{aligned}
\tag{18}
$$

where $R = \frac{N!}{(N-K)!} = R_m^i \times |C^K|$, the equal holds if and only if $p(C_m^K|\mathbf{w}_i) = p(C_m^{r,K}|\mathbf{w}_i)$ and $\sum_{j\neq r} p(C_m^{j,K}|\mathbf{w}_i) = 0$. This conclusion is intuitive because the entropy is smaller when the probability of the optimal route is larger.

A.2 Derivation of Eq. (5) and (6)

Eq. (5) is derived as follows:

$$
\begin{aligned}
&\log p(C_m^{r,K}|\mathbf{w}_i) \\
=&\log \left[p(b^{r,K}|C_{i,m}^{r,K-1}) p(b^{r,K-1}|C_{i,m}^{r,K-2}) \cdots p(b^{r,1}|\mathbf{w}_i) \right] \\
=&\sum_{k=1}^{K} \log p\left(b^{r,k}|C_{i,m}^{r,k-1} \right) \\
=&\sum_{k=1}^{K} \log \frac{p\left(b^{r,k}|C_{i,m}^{r,k-1} \right)}{p(b^{r,k})} + \sum_{k=1}^{K} \log p(b^{r,k}),
\end{aligned}
\tag{19}
$$

where we further denote $\xi(N,K) = \sum_{k=1}^{K} \log p(b^{r,k})$ for simplicity, which can be calculated as follows:

$$
\xi(N,K) \simeq K + \log \frac{(N-K)^{(N-K+\frac{1}{2})}}{N^{(N+\frac{1}{2})}},
\tag{20}
$$

where the approximation is more accurate when $N \gg K$, which is derived from Stirling's formula [34]. Besides, it has a more roughly lower bound $-K \log N$ for computation convenience.

Eq. (6) is derived as follows:

$$
\begin{aligned}
&p(C_m^{r,K}, \mathbf{w}_i) = p(C_{i,m}^{r,k}, b^{r,k+1}, ..., b^{r,K}) \\
=&p(C_{i,m}^{r,k}) \prod_{k'=k+1}^{K} p(b^{r,k'}|C_{i,m}^{r,k'-1}) \\
=&p(C_{i,m}^{r,k}) \exp \left(\sum_{k'=k+1}^{K} \log p(b^{r,k'}|C_{i,m}^{r,k'-1}) \right) \\
\simeq&p(C_{i,m}^{r,k}) \exp \left((K-k) \mathop{\mathbb{E}}_{b^k} \left[\log p(b^k|C_{i,m}^{k-1}) \right] \right),
\end{aligned}
\tag{21}
$$

where the sum is estimated by Monte Carlo.

A.3 Relationship of Local and Global MI

Based on Eq. (19) and (21), Eq. (4) can be derived as follows:

$$
\begin{aligned}
&I(\mathbf{W}; C^K) \geq \sum_{i=1}^{N} \sum_{r=1}^{R} p(C_m^{r,K}, \mathbf{w}_i) \log p(C_m^{r,K}|\mathbf{w}_i) + H(C_m^K) \\
\geq&\sum_{i=1}^{N} \sum_{r=1}^{R} \left(\sum_{k=1}^{K} p(C_m^{r,K}, \mathbf{w}_i) \log \frac{p\left(b^{r,k}|C_{i,m}^{r,k-1} \right)}{p(b^{r,k})} \right) + \xi(N,K) + H(C_m^K) \\
\simeq&H(C_m^K) + \xi(N,K) + \sum_{i=1}^{N} \sum_{r=1}^{R} \sum_{k=1}^{K} \Bigg(e^{(K-k)\, \mathbb{E}_{b^k} \left[\log p(b^k|C_{i,m}^{k-1}) \right]} \\
&\left(p(C_{i,m}^{r,k}) \log \frac{p(b^{r,k}|C_{i,m}^{r,k-1})}{p(b^{r,k})} \right) \Bigg) \\
=&H(C_m^K) + \xi(N,K) + \sum_{i=1}^{N} \sum_{k=1}^{K} \Bigg(e^{(K-k)\, \mathbb{E}_{b^k} \left[\log p(b^k|C_{i,m}^{k-1}) \right]} \\
&\sum_{r=1}^{R} \left(p(C_{i,m}^{r,k}) \log \frac{p(b^{r,k}|C_{i,m}^{r,k-1})}{p(b^{r,k})} \right) \Bigg) \\
=&H(C_m^K) + \xi(N,K) + \sum_{i=1}^{N} \sum_{k=1}^{K} \Bigg(e^{(K-k)\, \mathbb{E}_{b^k} \left[\log p(b^k|C_{i,m}^{k-1}) \right]} \\
&\sum_{b^{r,k}} \sum_{C_{i,m}^{r,k-1}} \left(p(b^{r,k}, C_{i,m}^{r,k-1}) \log \frac{p(b^{r,k}|C_{i,m}^{r,k-1})}{p(b^{r,k})} \right) \Bigg) \\
=&\sum_{i=1}^{N} \sum_{k=1}^{K} \left(e^{(K-k)\, \mathbb{E}_{b^k} \left[\log p(b^k|C_{i,m}^{k-1}) \right]} I\left(b^k; C_{i,m}^{k-1} \right) \right) \\
&+ H(C_m^K) + \xi(N,K) \\
=&\sum_{i=1}^{N} \sum_{k=1}^{K} \left(e^{(K-k) \log(N-k) + (K-k)\, \mathbb{E}_{b^k} \left[\log \frac{p(b^k|C_{i,m}^{k-1})}{p(b^k)} \right]} I\left(b^k; C_{i,m}^{k-1} \right) \right) \\
&+ H(C_m^K) + \xi(N,K) \\
=&\sum_{i=1}^{N} \sum_{k=1}^{K} \left(I\left(b^k; C_{i,m}^{k-1} \right) e^{(K-k)\, \mathbb{E}_{b^k} \left[\log \frac{p(b^k|C_{i,m}^{k-1})}{p(b^k)} \right]} (N-k)^{(K-k)} \right) \\
&+ H(C_m^K) + \xi(N,K)
\end{aligned}
\tag{22}
$$

B RELATIONSHIP BETWEEN SPGCL AND CPC

The original problem described in section 3 is to find K neighbors for N nodes, and we simplify it to find one neighbor for one nodes, i.e., $K = 1$ and $\mathbf{W}' = \{\mathbf{w}\}$. Besides, we denote $b = b^k$ for simplicity and Eq. (22) can be rewritten as follows:

$$
\begin{aligned}
& I(\mathbf{W}'; C^K) \\
&= \sum_{i=1}^{|\mathbf{W}'|} \sum_{k=1}^{1} \left(I\left(b; \mathbf{w}_i\right) e^{(K-k)\, \mathbb{E}_b\left[\log \frac{p(b|\mathbf{w}_i)}{p(b)}\right]} (N-k)^{(K-k)} \right) \\
&\quad + H(C_m^1) + \log p(b) \\
&= \sum_{i=1}^{|\mathbf{W}'|} I\left(b; \mathbf{w}_i\right) - \sum_{j=1}^{|\mathbf{W}|} p(b_j) \log p(b_j) + \log p(b) \\
&= I\left(b; \mathbf{w}\right) + \log N - \log N \\
&= I\left(b; \mathbf{w}\right),
\end{aligned} \tag{23}
$$

where $p(\mathbf{w}) = 1$ and $p(b) = \frac{1}{N}$. The first equal holds when $K = 1$ since the route from $\mathbf{w}$ to b is determined. Therefore we can conclude that, SPGCL degrades to CPC and InfoNCE loss [39] under some assumptions and constraints.

C PROOF OF LEMMA 3.1

$$
\begin{aligned}
& I(C, b; \mathbf{w}) - I(C; \mathbf{w}) \\
&= \sum_{C,b,\mathbf{w}} p(C, b, \mathbf{w}) \log \frac{p(C, b, \mathbf{w})}{p(C, b) p(\mathbf{w})} - \sum_{C,\mathbf{w}} p(C, \mathbf{w}) \log \frac{p(C, \mathbf{w})}{p(C) p(\mathbf{w})} \\
&= \sum_{C,b,\mathbf{w}} p(C, b, \mathbf{w}) \log \frac{p(C, b, \mathbf{w})}{p(C, b) p(\mathbf{w})} - \sum_{C,b,\mathbf{w}} p(C, b, \mathbf{w}) \log \frac{p(C, \mathbf{w})}{p(C) p(\mathbf{w})} \\
&= \sum_{C,b,\mathbf{w}} p(C, b, \mathbf{w}) \log \frac{p(C, b, \mathbf{w})}{p(C, b)} \cdot \frac{p(C)}{p(C, \mathbf{w})} \\
&= \sum_{C,b,\mathbf{w}} p(C, b, \mathbf{w}) \log \frac{p(C, b, \mathbf{w})}{p(b \mid C) p(C, \mathbf{w})} \\
&= \sum_{C,b,\mathbf{w}} p(b, \mathbf{w} \mid C) p(C) \log \frac{p(b, \mathbf{w} \mid C)}{p(b \mid C) p(\mathbf{w} \mid C)} \\
&= I(b; \mathbf{w} \mid C) \geq 0,
\end{aligned} \tag{24}
$$

where $\sum_{C,b,\mathbf{w}} = \sum_C \sum_b \sum_{\mathbf{w}}$.

D TRAINING AND ADAPTIVE LABELING ALGORITHMS

We train SPGCL according to Algorithm 1, and at the end of each epoch, edge labels are update via Algorithm 2, including label edges with high confidence and unlabel edges with low confidence. In practice, some edges are hard to classify, resulting in $E^U \neq \emptyset$. Next, we analyse the time complexity of SPGCL as follows:

- In Algorithm 1, learning embedding via Eq. (13) and (14) costs $O(NF'^2 + NF')$, where N is number of nodes and F' is the length of embeddings. The aggregation requires $O(NH(TFF' + TF + TFT'))$ time, where H is number of multi-heads, T and T' are the length of input and output sequence respectively. Furthermore, the pair-wise similarity requires $O(N^2F')$. Both $\mathcal{L}_N$ and $\mathcal{L}_{PU}$ need $O(N^2)$ computation, and $\mathcal{L}_{mse}$ costs $O(NTF^2)$. At the end of each epoch, Algorithm 2 costs $O(N^2)$, which is introduced below. Since $N \gg F' \geq F$ in practice, the overall complexity of Algorithm 1 is $O(NHTFF' + N^2F')$.
- The complexities of KNN and DTW are $O(NF)$ and $O(N^2)$ respectively when calculating $\mathbf{W}$ in Eq. (12). Once the data is prepared, it can be reused during training and testing.
- The cost of Algorithm 2 is expected to be $O(N^2)$, which mainly comes from Hadamard product and calling $topn$ on the matrix.

Algorithm 1 The training of SPGCL.

```
Require: Locations $\mathbf{V} \in \mathbb{R}^{N \times d}$, deformations $\mathbf{X} \in \mathbb{R}^{N \times (T+T') \times F}$, initial
    neighbor volume $k$.
 1: function $sim(\mathbf{W}, E^P)$:                                ▷ Matching score function
 2:     Obtain $\mathbf{R}$ and $\mathbf{R}_c$ via Eq. (13) and (14) respectively;
 3:     Calculate matching score $\mathbf{S}$ via Eq. (15);
 4:     return $\mathbf{S}$
 5: end function
 6: Calculate $\mathbf{W}$ via Eq. (12);
 7: Initialize $E^P$ and $E^N$ via KNN with $k$, the mistakenly-labeled counter
    mask matrix $E^M$ with zeros;
 8: Minimizing $\mathcal{L}_N$ and $\mathcal{L}_{PU}$ via Eq. (8) and (11) to warm $sim$ up;
 9: while SPGCL is not converged do
10:     Apply augmentations to $\mathbf{W}$ and $E$;
11:     Obtain matching score $\mathbf{S} = sim(\mathbf{W}, E^P)$;
12:     Calculate $\mathcal{L}_N$, $\mathcal{L}_{PU}$ and $\mathcal{L}_{mse}$ via Eq. (8) (11) and (16) respectively;
13:     Minimize hybrid loss $\mathcal{L}$ via Adam optimizer [23];
14:     Update labels via Algorithm 2;
15: end while
```

Algorithm 2 The pseudo-code of updating labels.

```
Require: Positive edges mask $E^P$ and the initialized $E^P_{k=0}$, unlabeled edges
    mask $E^U$ and the initialized $E^N_{k=0}$, negative edges mask $E^N$, mistakenly-
    labeled counter matrix $E^M$, score matrix $\mathbf{S}$, numbers of candidates $\alpha$
    and $\beta$, threshold $\delta^{\pm}$, patience $\lambda$. Note that masks are all 0-1 matrixes,
    and $\Delta = 5e^{-4}$ is predefined parameter.
 1: function $topn(\mathbf{S}, n, \text{largest})$:
 2:     if largest == True then
 3:         Set scores to 0, except the largest $n$ for each node;
 4:     else
 5:         Set scores to 0, except the smallest $n$ for each node;
 6:     end if
 7:     return $\mathbf{S}$
 8: end function
 9: $\mathbf{S}^P = topn(\mathbf{S} \odot E^U, \alpha, \text{True})$                   ▷ Positive candidates
10: $E^P = E^P + [\mathbf{S}^P \geq \delta^+]$;
11: $\mathbf{S}^N = topn(\mathbf{S} \odot E^U, \beta, \text{False})$                  ▷ Negative candidates
12: $E^N = E^N + [\mathbf{S}^N \leq \delta^-]$;
13: $E^M_e = [\mathbf{S} \odot E^P < \delta^+] + [\mathbf{S} \odot E^N > \delta^-]$; ▷ Count if the edge is labeled
    mistakenly
14: $E^M = E^M \odot E^M_e + E^M_e$;
15: $E^P = [E^P \odot E^M < \lambda] \bigcup E^P_{k=0}$; ▷ Unlabel edge if the count exceeds the
    patience, but always remember the initial labels
16: $E^N = [E^N \odot E^M < \lambda] \bigcup E^N_{k=0}$;
17: $E^U = 1 - E^P - E^N$;                                        ▷ Update the $E^U$
18: $E^M = E^M \odot E^U$;                                         ▷ Reset the counter
19: $\delta^+ = \delta^+ - \Delta$ and $\delta^- = \delta^- + \Delta$;                    ▷ Update threshold
```