Appendix

The Appendix is organized as follows.

- Section A presents the detailed examples and derivations of consensus equations.
- Section B includes proofs and other details about our theoretical results. Particularly,
  - Section B.1 proves the uniqueness of $\boldsymbol{T}$.
  - Section B.2 justifies the feasibility of assumption $|E_3^*| = \Theta(N)$
  - Section B.3 shows the proof for Lemma 1
  - Section B.4 shows the proof for Theorem 2.
- Section C presents more discussions, e.g., the soft 2-NN label clusterability, more details on local $\boldsymbol{T}(X)$, and the feasibility of our Assumption 1 & 2 to guarantee the uniqueness of $\boldsymbol{T}$.
- Section D shows more experimental settings and results.

A. Derivation of Consensus Equations

For the first-order consensuses, we have

$$\mathbb{P}(\widetilde{Y}_1 = j_1) = \sum_{i \in [K]} \mathbb{P}(\widetilde{Y}_1 = j_1 | Y_1 = i)\mathbb{P}(Y_1 = i).$$

For the second-order consensuses, we have

$$\begin{aligned}
&\mathbb{P}(\widetilde{Y}_1 = j_1, \widetilde{Y}_2 = j_2) \\
=& \sum_{i \in [K]} \mathbb{P}(\widetilde{Y}_1 = j_1, \widetilde{Y}_2 = j_2 | Y_1 = i, Y_2 = i)\mathbb{P}(Y_1 = Y_2 = i) \\
\overset{(a)}{=}& \sum_{i \in [K]} \mathbb{P}(\widetilde{Y}_1 = j_1, \widetilde{Y}_2 = j_2 | Y_1 = i, Y_2 = i) \cdot \mathbb{P}(Y_1 = i) \\
\overset{(b)}{=}& \sum_{i \in [K]} \mathbb{P}(\widetilde{Y}_1 = j_1 | Y_1 = i) \cdot \mathbb{P}(\widetilde{Y}_2 = j_2 | Y_2 = i) \cdot \mathbb{P}(Y_1 = i),
\end{aligned}$$

where equality (a) holds due to the 2-NN label clusterability, i.e., $Y_1 = Y_2 (= Y_3)$ w.p. 1, and equality (b) holds due to the conditional independency between $\widetilde{Y}_1$ and $\widetilde{Y}_2$ given their clean labels.

For the third-order consensuses, we have

$$\begin{aligned}
&\mathbb{P}(\widetilde{Y}_1 = j_1, \widetilde{Y}_2 = j_2, \widetilde{Y}_3 = j_3) \\
=& \sum_{i \in [K]} \mathbb{P}(\widetilde{Y}_1 = j_1, \widetilde{Y}_2 = j_2, \widetilde{Y}_3 = j_3 | Y_1 = i, Y_2 = i, Y_3 = i)\mathbb{P}(Y_1 = Y_2 = Y_3 = i) \\
\overset{(a)}{=}& \sum_{i \in [K]} \mathbb{P}(\widetilde{Y}_1 = j_1, \widetilde{Y}_2 = j_2, \widetilde{Y}_3 = j_3 | Y_1 = i, Y_2 = i, Y_3 = i)\mathbb{P}(Y_1 = i) \\
\overset{(b)}{=}& \sum_{i \in [K]} \mathbb{P}(\widetilde{Y}_1 = j_1 | Y_1 = i)\mathbb{P}(\widetilde{Y}_2 = j_2 | Y_2 = i)\mathbb{P}(\widetilde{Y}_3 = j_3 | Y_3 = i)\mathbb{P}(Y_1 = i).
\end{aligned}$$

where equality (a) holds due to the 3-NN label clusterability, i.e., $Y_1 = Y_2 = Y_3$ w.p. 1, and equality (b) holds due to the conditional independency between $\widetilde{Y}_1$, $\widetilde{Y}_2$ and $\widetilde{Y}_3$ given their clean labels.

With the above analyses, there are 2 first-order equations,

$$\begin{aligned}
\mathbb{P}(\widetilde{Y}_1 = 1) &= p_1(1 - e_1) + (1 - p_1)e_2, \\
\mathbb{P}(\widetilde{Y}_1 = 2) &= p_1 e_1 + (1 - p_1)(1 - e_2).
\end{aligned}$$

There are 4 second-order equations for different combinations of $\widetilde{Y}_1, \widetilde{Y}_2$, e.g.,

$$\begin{aligned}
&\mathbb{P}(\widetilde{Y}_1 = 1, \widetilde{Y}_2 = 1) = p_1(1-e_1)^2 + (1-p_1)e_2^2,\\
&\mathbb{P}(\widetilde{Y}_1 = 1, \widetilde{Y}_2 = 2) = p_1(1-e_1)e_1 + (1-p_1)e_2(1-e_2),\\
&\mathbb{P}(\widetilde{Y}_1 = 2, \widetilde{Y}_2 = 1) = p_1(1-e_1)e_1 + (1-p_1)e_2(1-e_2),\\
&\mathbb{P}(\widetilde{Y}_1 = 1, \widetilde{Y}_2 = 1) = p_1e_1^2 + (1-p_1)(1-e_2)^2.
\end{aligned}$$

There are 8 third-order equations for different combinations of $\widetilde{Y}_1, \widetilde{Y}_2, \widetilde{Y}_3$, e.g.,

$$\begin{aligned}
&\mathbb{P}(\widetilde{Y}_1 = 1, \widetilde{Y}_2 = 1, \widetilde{Y}_3 = 1) = p_1(1-e_1)^3 + (1-p_1)e_2^3,\\
&\mathbb{P}(\widetilde{Y}_1 = 1, \widetilde{Y}_2 = 1, \widetilde{Y}_3 = 2) = p_1(1-e_1)^2e_1 + (1-p_1)e_2^2(1-e_2),\\
&\mathbb{P}(\widetilde{Y}_1 = 1, \widetilde{Y}_2 = 2, \widetilde{Y}_3 = 1) = p_1(1-e_1)^2e_1 + (1-p_1)e_2^2(1-e_2),\\
&\mathbb{P}(\widetilde{Y}_1 = 1, \widetilde{Y}_2 = 2, \widetilde{Y}_3 = 2) = p_1(1-e_1)e_1^2 + (1-p_1)e_2(1-e_2)^2,\\
&\mathbb{P}(\widetilde{Y}_1 = 2, \widetilde{Y}_2 = 1, \widetilde{Y}_3 = 1) = p_1(1-e_1)^2e_1 + (1-p_1)e_2^2(1-e_2),\\
&\mathbb{P}(\widetilde{Y}_1 = 2, \widetilde{Y}_2 = 1, \widetilde{Y}_3 = 2) = p_1(1-e_1)e_1^2 + (1-p_1)e_2(1-e_2)^2,\\
&\mathbb{P}(\widetilde{Y}_1 = 2, \widetilde{Y}_2 = 2, \widetilde{Y}_3 = 1) = p_1(1-e_1)e_1^2 + (1-p_1)e_2(1-e_2)^2,\\
&\mathbb{P}(\widetilde{Y}_1 = 2, \widetilde{Y}_2 = 2, \widetilde{Y}_3 = 2) = p_1e_1^3 + (1-p_1)(1-e_2)^3.
\end{aligned}$$

For a general K-class classification problem, we show one first-order consensus below:

$$\begin{aligned}
&\boldsymbol{e}_j^\top \boldsymbol{c}^{[1]} = \mathbb{P}(\widetilde{Y}_1 = j)\\
=&\sum_{i\in[K]} \mathbb{P}(\widetilde{Y}_1 = j | Y_1 = i)\mathbb{P}(Y_1 = i)\\
=&\sum_{i\in[K]} T_{ij}\cdot p_i = \boldsymbol{e}_j^\top \boldsymbol{T}^\top \boldsymbol{p}.
\end{aligned}$$

The second-order consensus follows the example below:

$$\begin{aligned}
&\boldsymbol{e}_j^\top \boldsymbol{c}_r^{[2]} = \mathbb{P}(\widetilde{Y}_1 = j, \widetilde{Y}_2 = (j+r)_K)\\
\stackrel{(a)}{=}&\sum_{i\in[K]} \mathbb{P}(\widetilde{Y}_1 = j | Y_1 = i)\mathbb{P}(\widetilde{Y}_2 = (j+r)_K | Y_2 = i)\mathbb{P}(Y_1 = i)\\
=&\sum_{i\in[K]} T_{i,j}\cdot T_{i,(j+r)_K}\cdot p_i \stackrel{(b)}{=} \boldsymbol{e}_j^\top (\boldsymbol{T}\circ\boldsymbol{T}_r)^\top \boldsymbol{p},
\end{aligned}$$

where equality (a) holds again due to the 2-NN label clusterability the conditional independency (similar to binary cases), and equality (b) holds due to $\boldsymbol{T}_r[i,j] = T_{i,(j+r)_K}$. We also show one third-order consensus below:

$$\begin{aligned}
&\boldsymbol{e}_j^\top \boldsymbol{c}_r^{[3]} = \mathbb{P}(\widetilde{Y}_1 = j, \widetilde{Y}_2 = (j+r)_K, \widetilde{Y}_3 = (j+s)_K)\\
\stackrel{(a)}{=}&\sum_{i\in[K]} \mathbb{P}(\widetilde{Y}_1 = j | Y_1 = i)\mathbb{P}(\widetilde{Y}_2 = (j+r)_K | Y_2 = i)\mathbb{P}(\widetilde{Y}_3 = (j+s)_K | Y_3 = i)\mathbb{P}(Y_1 = i)\\
=&\sum_{i\in[K]} T_{i,j}\cdot T_{i,(j+r)_K}\cdot T_{i,(j+s)_K}\cdot p_i \stackrel{(b)}{=} \boldsymbol{e}_j^\top (\boldsymbol{T}\circ\boldsymbol{T}_r\circ\boldsymbol{T}_s)^\top \boldsymbol{p},
\end{aligned}$$

where equality (a) holds again due to the 3-NN label clusterability the conditional independency (similar to binary cases), and equality (b) holds due to $\boldsymbol{T}_r[i,j] = T_{i,(j+r)_K}$, $\boldsymbol{T}_s[i,j] = T_{i,(j+s)_K}$.

B. Theoretical Guarantees

B.1. Uniqueness of $\boldsymbol{T}$

We need to prove the following equations have a unique solution when $\boldsymbol{T}$ is non-singular and informative.

Consensus Equations

- First-order (K equations):

$$\boldsymbol{c}^{[1]} := \boldsymbol{T}^\top \boldsymbol{p},$$

- Second-order (K^2 equations):

$$\boldsymbol{c}_r^{[2]} := (\boldsymbol{T} \circ \boldsymbol{T}_r)^\top \boldsymbol{p},\ r \in [K],$$

- Third-order (K^3 equations):

$$\boldsymbol{c}_{r,s}^{[3]} := (\boldsymbol{T} \circ \boldsymbol{T}_r \circ \boldsymbol{T}_s)^\top \boldsymbol{p},\ r, s \in [K].$$

Firstly, we need the following Lemma for the Hadamard product of matrices:

Lemma 2. *(Horn & Johnson, 2012) For column vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, and corresponding diagonal matrices $\boldsymbol{D}_{\boldsymbol{x}}$ and $\boldsymbol{D}_{\boldsymbol{y}}$ with these vectors as their main diagonals, the following identity holds:*

$$\boldsymbol{x}^*(\boldsymbol{A} \circ \boldsymbol{B})\boldsymbol{y} = \mathrm{tr}\left(\boldsymbol{D}_{\boldsymbol{x}}^* \boldsymbol{A} \boldsymbol{D}_{\boldsymbol{y}} \boldsymbol{B}^\top\right),$$

where $\boldsymbol{x}^*$ denotes the conjugate transpose of $\boldsymbol{x}$.

The following proof focuses on the second and third-order consensuses. It is worth noting that, although the first-order consensus is not necessary for the derivation of the unique solution, it still helps improve the stability of solving for $\boldsymbol{T}$ and $\boldsymbol{p}$ numerically.

Step I: Transform the second-order equations. Denoted by $\boldsymbol{T}_r = \boldsymbol{T}\boldsymbol{S}_r$, where $\boldsymbol{S}_r$ permutes particular columns of $\boldsymbol{T}$. Let $\boldsymbol{e}_i$ be the column vector with only the i-th element being 1 and 0 otherwise. With Lemma 2, the second-order consensus can be transformed as

$$\boldsymbol{e}_i^\top \boldsymbol{c}_r^{[2]} = \boldsymbol{e}_i^\top (\boldsymbol{T} \circ \boldsymbol{T}_r)^\top \boldsymbol{p} = \mathrm{tr}\left(\boldsymbol{D}_{\boldsymbol{e}_i} \boldsymbol{T}^\top \boldsymbol{D}_{\boldsymbol{p}} \boldsymbol{T} \boldsymbol{S}_r\right)$$

Then the $(i, (i+r)_K)$-th element of matrix $\boldsymbol{T}^\top \boldsymbol{D}_{\boldsymbol{p}} \boldsymbol{T}$ is

$$(\boldsymbol{T}^\top \boldsymbol{D}_{\boldsymbol{p}} \boldsymbol{T})[i, (i+r)_K] = \boldsymbol{e}_i^\top \boldsymbol{c}_r^{[2]}.$$

With a fixed $\boldsymbol{e}_i^\top \boldsymbol{c}_r^{[2]}, \forall i, r \in [K]$, denote by

$$\boldsymbol{T}^\top \boldsymbol{D}_{\boldsymbol{p}} \boldsymbol{T} = \boldsymbol{T}_\dagger, \tag{11}$$

where $\boldsymbol{T}_\dagger[i, (i+r)_K] = \boldsymbol{e}_i^\top \boldsymbol{c}_r^{[2]}$. Note $\boldsymbol{T}_\dagger$ is fixed given $\boldsymbol{c}_r^{[2]}, \forall r \in [K]$.

Step II: Transform the third-order equations. Following the idea in Step I, we can also transform the third-order equations. First, notice that

$$\boldsymbol{e}_i^\top \boldsymbol{c}_{r,s}^{[3]} = \boldsymbol{e}_i^\top [(\boldsymbol{T} \circ \boldsymbol{T}_s) \circ \boldsymbol{T}_r]^\top \boldsymbol{p} = \mathrm{tr}\left(\boldsymbol{D}_{\boldsymbol{e}_i} (\boldsymbol{T} \circ \boldsymbol{T}_s)^\top \boldsymbol{D}_{\boldsymbol{p}} \boldsymbol{T} \boldsymbol{S}_r\right).$$

Then the $(i, (i+r)_K)$-th element of matrix $(\boldsymbol{T} \circ \boldsymbol{T}_s)^\top \boldsymbol{D}_{\boldsymbol{p}} \boldsymbol{T}$ is

$$((\boldsymbol{T} \circ \boldsymbol{T}_s)^\top \boldsymbol{D}_{\boldsymbol{p}} \boldsymbol{T})[i, (i+r)_K] = \boldsymbol{e}_i^\top \boldsymbol{c}_{r,s}^{[3]}.$$

With a fixed $\boldsymbol{e}_i^\top \boldsymbol{c}_{r,s}^{[3]}, \forall i, r \in [K]$, denote by

$$(\boldsymbol{T} \circ \boldsymbol{T}_s)^\top \boldsymbol{D}_{\boldsymbol{p}} \boldsymbol{T} = \boldsymbol{T}_{\ddagger,s} \Rightarrow \boldsymbol{T}^\top \boldsymbol{D}_{\boldsymbol{p}} (\boldsymbol{T} \circ \boldsymbol{T}_s) = \boldsymbol{T}_{\ddagger,s}^\top, \tag{12}$$

where $\boldsymbol{T}_{\ddagger,s}[i,(i+r)_K] = \boldsymbol{e}_i^\top \boldsymbol{c}_{r,s}^{[3]}$. According to Eqn. (11), we have

$$\boldsymbol{T}^\top \boldsymbol{D}_{\boldsymbol{p}}(\boldsymbol{T} \circ \boldsymbol{T}_s) = \boldsymbol{T}^\top \boldsymbol{D}_{\boldsymbol{p}} \boldsymbol{T}\boldsymbol{T}^{-1}(\boldsymbol{T} \circ \boldsymbol{T}_s) = \boldsymbol{T}_\dagger \boldsymbol{T}^{-1}(\boldsymbol{T} \circ \boldsymbol{T}_s) = \boldsymbol{T}_{\ddagger,s}^\top.$$

Thus

$$(\boldsymbol{T} \circ \boldsymbol{T}_s) = \boldsymbol{T}\boldsymbol{T}_\dagger^{-1}\boldsymbol{T}_{\ddagger,s}^\top, \forall s \in [K]. \tag{13}$$

Step III: From matrices to vectors With Step I and Step II, we could transform the equations formulated by the second and the third-order consensuses to a particular system of multivariate quadratic equations of $\boldsymbol{T}$ in Eqn. (13). Generally, these equations could have up to 2^{K^2} solutions introduced by different combinations of each element in $\boldsymbol{T}$. To prove the uniqueness of $\boldsymbol{T}$, we need to exploit the structure of the equations in (13).

For a clear representation of the structure of equations and solutions, we first consider one subset of the equations in (13). Specifically, let $s = 0$ we have

$$(\boldsymbol{T} \circ \boldsymbol{T}) = \boldsymbol{T}\boldsymbol{T}_\dagger^{-1}\boldsymbol{T}_\ddagger^\top. \tag{14}$$

Then we need to study the number of feasible $\boldsymbol{T}$ satisfying Eqn. (14). Denote by $\boldsymbol{A} = \boldsymbol{T}_\ddagger(\boldsymbol{T}_\dagger^{-1})^\top$. Then each row of $\boldsymbol{T}$, denoted by $\boldsymbol{u}^\top$, is a solution to the equation

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{D}_{\boldsymbol{u}}\boldsymbol{u} \quad \text{(a.k.a. } \boldsymbol{A}\boldsymbol{u} = \boldsymbol{u} \circ \boldsymbol{u}\text{)}. \tag{15}$$

Till now, in Step III, we split the matrix $\boldsymbol{T}$ to several vectors $\boldsymbol{u}$, and transform our target from finding a matrix solution $\boldsymbol{T}$ for (13) to a set of vector solutions $\boldsymbol{u}$ for (15).

Assume there are M feasible $\boldsymbol{u}$ vectors. We collect all the possible $\boldsymbol{u}$ and define $\boldsymbol{U} := [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_M], \boldsymbol{u}_i \neq \boldsymbol{u}_{i'}, \forall i, i' \in [M]$. If $M = K$, we know there exists at most $K!$ different $\boldsymbol{T}$ (considering all the possible permutations of $\boldsymbol{u}$) that Eqn. (14) holds. Further, by considering an informative $\boldsymbol{T}$ as Assumption 2, we can identify a particular permutation. Therefore, if $M = K$ and $\boldsymbol{T}$ is informative, we know there exists and only exists one unique $\boldsymbol{T}$ that Eqn. (14) holds.

Step IV: Constructing the M-th vector Supposing $M > K$, we have

$$\boldsymbol{A}\boldsymbol{U} = \boldsymbol{A}[\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_K, \cdots \boldsymbol{u}_M] = [\boldsymbol{D}_{\boldsymbol{u}_1}\boldsymbol{u}_1, \boldsymbol{D}_{\boldsymbol{u}_2}\boldsymbol{u}_2, \cdots, \boldsymbol{D}_{\boldsymbol{u}_K}\boldsymbol{u}_K, \cdots \boldsymbol{D}_{\boldsymbol{u}_M}\boldsymbol{u}_M].$$

With a non-singular $\boldsymbol{T}$ (Assumption 1), without loss of generality, we will assume the first K columns are full-rank. Then $\boldsymbol{u}_M$ must be a linear combination of the first K columns, i.e., $\boldsymbol{u}_M = \sum_{i\in[K]} \lambda_i \boldsymbol{u}_i = \boldsymbol{U}\boldsymbol{\lambda}_0$, where $\boldsymbol{\lambda}_0 = [\lambda_1, \lambda_2, \cdots, \lambda_K, 0, \cdots, 0]$. According to the equation $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{D}_{\boldsymbol{u}}\boldsymbol{u} = \boldsymbol{u} \circ \boldsymbol{u}$, we have

$$\boldsymbol{A}\boldsymbol{u}_M = \boldsymbol{D}_{\boldsymbol{u}_M}\boldsymbol{u}_M = \boldsymbol{D}_{\boldsymbol{U}\boldsymbol{\lambda}_0}\boldsymbol{U}\boldsymbol{\lambda}_0,$$

and

$$\boldsymbol{A}\boldsymbol{u}_M = \sum_{i\in[M]} \boldsymbol{\lambda}_0[i]\boldsymbol{A}\boldsymbol{u}_i = \sum_{i\in[M]} \boldsymbol{\lambda}_0[i]\boldsymbol{u}_i \circ \boldsymbol{u}_i = (\boldsymbol{U} \circ \boldsymbol{U})\boldsymbol{\lambda}_0.$$

Thus

$$(\boldsymbol{U} \circ \boldsymbol{U})\boldsymbol{\lambda}_0 = \boldsymbol{D}_{\boldsymbol{U}\boldsymbol{\lambda}_0}\boldsymbol{U}\boldsymbol{\lambda}_0 = (\boldsymbol{U}\boldsymbol{\lambda}_0) \circ (\boldsymbol{U}\boldsymbol{\lambda}_0).$$

Note that, the matrix $\boldsymbol{U}$ can be written as $\boldsymbol{U} = [\boldsymbol{U}_K, \boldsymbol{U}_{M-K}]$, and the vector $\boldsymbol{\lambda}_0$ can be written as $\boldsymbol{\lambda}_0 = [\boldsymbol{\lambda}^\top, 0, \cdots, 0]^\top$, where $\boldsymbol{\lambda} := [\lambda_1, \cdots, \lambda_K]^\top$. Then the above equation can be transformed as follows:

$$(\boldsymbol{U}_K \circ \boldsymbol{U}_K)\boldsymbol{\lambda} = \boldsymbol{u}_M \circ \boldsymbol{u}_M, \text{ and } \boldsymbol{U}_K\boldsymbol{\lambda} = \boldsymbol{u}_M.$$

Similarly, $\forall s \in [K]$, we have

$$(\boldsymbol{U}_K \circ (\bar{\boldsymbol{S}}_s\boldsymbol{U}_K))\boldsymbol{\lambda} = \boldsymbol{u}_M \circ (\bar{\boldsymbol{S}}_s\boldsymbol{u}_M), \text{ and } \boldsymbol{U}_K\boldsymbol{\lambda} = \boldsymbol{u}_M,$$

where $\bar{\boldsymbol{S}}_s\boldsymbol{u}_M$ denotes a row circular shift such that $(\bar{\boldsymbol{S}}_s\boldsymbol{u}_M)[i] = \boldsymbol{u}_M[i+s]$. Note $\bar{\boldsymbol{S}}_s = \boldsymbol{S}_s^\top$. Applying Lemma 2, we have

$$\text{tr}(\boldsymbol{D}_{\boldsymbol{e}_i}\boldsymbol{U}_K\boldsymbol{D}_{\boldsymbol{\lambda}}\boldsymbol{U}_K^\top\bar{\boldsymbol{S}}_s^\top) = \text{tr}(\boldsymbol{D}_{\boldsymbol{e}_i}\boldsymbol{U}_K\boldsymbol{D}_{\boldsymbol{\lambda}}\boldsymbol{U}_K^\top\boldsymbol{S}_s) = (\boldsymbol{u}_M \circ (\bar{\boldsymbol{S}}_s\boldsymbol{u}_M))[i]$$

Then the $(i, (i+s)_K)$-th element of matrix $\boldsymbol{U}_K \boldsymbol{D}_{\boldsymbol{\lambda}} \boldsymbol{U}_K^\top$ is

$$(\boldsymbol{U}_K \boldsymbol{D}_{\boldsymbol{\lambda}} \boldsymbol{U}_K^\top)[i, (i+s)_K] = (\boldsymbol{u}_M \circ (\bar{\boldsymbol{S}}_s \boldsymbol{u}_M))[i] = \boldsymbol{u}_M[i] \cdot \boldsymbol{u}_M[(i+s)_K].$$

Then we have

$$\boldsymbol{U}_K \boldsymbol{D}_{\boldsymbol{\lambda}} \boldsymbol{U}_K^\top = \boldsymbol{Q}, \text{ and } \boldsymbol{Q} = \boldsymbol{u}_M \boldsymbol{u}_M^\top.$$

When $\boldsymbol{T}$ is non-singular, we know $\boldsymbol{U}$ is invertible (full-rank), then

$$\boldsymbol{D}_{\boldsymbol{\lambda}} = (\boldsymbol{U}_K^{-1} \boldsymbol{u}_M)(\boldsymbol{U}_K^{-1} \boldsymbol{u}_M)^\top.$$

Thus $\mathsf{Rank}(\boldsymbol{D}_{\boldsymbol{\lambda}}) = 1$. Recalling $\boldsymbol{1}^\top \boldsymbol{\lambda} = 1$, the vector $\boldsymbol{\lambda}$ could only be one-hot vectors, i.e. $\boldsymbol{e}_i, \forall i \in [K]$. This proves $\boldsymbol{u}_M$ must be the same as one of $\boldsymbol{u}_i, i \in [K]$.

Wrapping-up: Unique $\boldsymbol{T}$ From Step III, we know that, if $M = K$, we have a unique $\boldsymbol{T}$ under the assumption that $\boldsymbol{T}$ is informative and non-singular. Step IV proves the M-th ($M > K$) vector $\boldsymbol{u}$ must be identical to one of $\boldsymbol{u}_i, i \in [K]$, indicating we only have $M = K$ non-repetitive $\boldsymbol{u}$ vectors. Therefore, our consensus equations are sufficient for guaranteeing a unique $\boldsymbol{T}$. Besides, note there is no approximation applied during the whole proof. Thus with a perfect knowledge of $\boldsymbol{c}^{[\nu]}, \nu = 1, 2, 3$, the unique $\boldsymbol{T}$ satisfying the consensus equations is indeed the true noise transition matrix.

B.2. Feasibility of Assumption $|E_3^*| = \Theta(N)$

We discuss the feasibility of our assumption on the number of 3-tuples. According to the definition of E_3^*, we know there are no more than $|E_3^*| \le \lfloor N/3 \rfloor$ feasible 3-tuples. Strictly deriving the lower bound for $|E_3^*|$ is challenging due to the unknown distributions of representations. To roughly estimate the order of $|E_3^*|$ (i.e., the maximum number of non-overlapping 3-tuples), we consider a special scenario where those high-dimensional representations could be mapped to a 2-D square of width $\sqrt{N/3}$, each grid of width 1 has exactly 3 mapped representations, and one mapped representation is at the center of each grid (also the center of each circle). Consider a particular construction of feasible 3-tuples as illustrated in Figure 4. We require that, for each grid, the 2-NN fall in the corresponding circle. Otherwise, they may become the 2-NN of representations in other nearby girds. Assume the 2-NN are independently and uniformly distributed in the unit square, thus the probability of both 2-NN falling in the circle is $(\pi/4)^2$. Noting there are $N/3$ grids in the big square illustrated in Figure 4, the expected number of feasible 3-tuples in this case is $\frac{\pi^2}{48} \cdot N = \Theta(N)$. Although this example only considers a special case, it demonstrates the order of $|E_3^*|$ could be $\Theta(N)$ with appropriate representations.

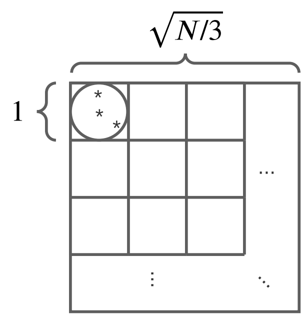

Figure 4. Illustration of a special case.

B.3. Proof for Lemma 1

Then we present the proof for Lemma 1.

Proof. Recall in Eqn. (7), each high-order consensus pattern could be estimated by the sample mean of $|E_3^*|$ independent and identically distributed random variables, thus according to Hoeffding's inequality (Hoeffding, 1963), w.p. $1 - \delta$, we have

$$|\hat{\boldsymbol{c}}^{[i]}[j] - \boldsymbol{c}^{[i]}[j]| \le \sqrt{\frac{\ln \frac{2}{\delta}}{2|E_3^*|}}, i = 1, 2, 3, \forall j,$$

which is at the order of $O(\sqrt{\ln(1/\delta)/N})$. $\square$

B.4. Proof for Theorem 2

Consider a particular uniform off-diagonal matrix $\boldsymbol{T}$, where the off-diagonal elements are $T_{ij} = \frac{1-T_{ii}}{K-1}$. Recall the clean prior probability for the i-th class is p_i. To find the upper bound for the sample complexity, we can only consider a subset of our consensus equations. Specifically, we consider the equations related to the i-th element of Eqn. (2) and Eqn. (3) when $r=0$. Then a solution to our consensus equations will need to satisfy at least the following two equations:

$$\hat{p}_i \hat{T}_{ii} + (1-\hat{p}_i)\frac{1-\hat{T}_{ii}}{K-1} = \hat{c}_1, \tag{16}$$

$$\hat{p}_i \hat{T}_{ii}^2 + (1-\hat{p}_i)\frac{(1-\hat{T}_{ii})^2}{(K-1)^2} = \hat{c}_2, \tag{17}$$

where $\hat{p}_i$ and $\hat{T}_{ii}$ denote the estimated clean prior probability and noisy transition matrix, $\hat{c}_1$ and $\hat{c}_2$ denote the corresponding estimates of first- and second-order statistics. Lemma 1 shows, with probability $1-\delta$:

$$|\hat{c}_i - c_i| \le O\left(\sqrt{\frac{\ln(1/\delta)}{N}}\right).$$

Multiplying both sides of Eqn. (16) by T_{ii} and adding Eqn. (17), we have

$$K(K-1)\hat{p}_i\hat{T}_{ii}^2 + (1-\hat{p}_i)(1-\hat{T}_{ii}) = (K-1)\hat{c}_1\hat{T}_{ii} + (K-1)^2\hat{c}_2.$$

Note the above equality also holds for the true values p_i, T_{ii}, c_1, c_2. Taking the difference we have

$$\begin{aligned}&(\hat{T}_{ii}-T_{ii})(K(K-1)p_i(T_{ii}+\hat{T}_{ii}) - (1-p_i) - (K-1)c_1)\\ =&(K-1)^2(\hat{c}_2-c_2) + (K-1)(\hat{c}_1-c_1)\hat{T}_{ii} - K(K-1)\hat{T}_{ii}^2(\hat{p}_i-p_i) - (\hat{T}_{ii}-1)(\hat{p}_i-p_i).\end{aligned}$$

Taking the absolute value for both sides yields

$$\begin{aligned}&|\hat{T}_{ii}-T_{ii}|\cdot|K(K-1)p_i(T_{ii}+\hat{T}_{ii}) - (1-p_i) - (K-1)c_1|\\ \le&(K-1)^2|\hat{c}_2-c_2| + (K-1)|\hat{c}_1-c_1| + (K(K-1)+1)|\hat{p}_i-p_i|\end{aligned}$$

From Eqn. (16), we have

$$\hat{p}_i = \frac{K-1}{K}\frac{\hat{c}_1 - 1/K}{\hat{T}_{ii}-1/K} + \frac{1}{K}.$$

Thus

$$|\hat{p}_i - p_i| \le \frac{K-1}{K}\frac{|\hat{c}_1-c_1|}{\min(\hat{T}_{ii}, T_{ii}) - 1/K},$$

indicating $|\hat{p}_i - p_i|$ is at the order of $|\hat{c}_1 - c_1|$. Note that

$$K(K-1)p_i(T_{ii}+\hat{T}_{ii}) - (1-p_i) - (K-1)c_1 \ge K(K-1)p_iT_{ii} - (1-p_i) - (K-1)c_1.$$

When $K(K-1)p_iT_{ii} - (1-p_i) - (K-1)c_1 > 0$, we have

$$|\hat{T}_{ii}-T_{ii}| \le \frac{(K-1)^2|\hat{c}_2-c_2| + (K-1)|\hat{c}_1-c_1| + (K(K-1)+1)\frac{K-1}{K}\frac{|\hat{c}_1-c_1|}{\min(\hat{T}_{ii},T_{ii})-1/K}}{K(K-1)p_iT_{ii} - (1-p_i) - (K-1)c_1}.$$

Then by union bound we know, w.p. $1-2\delta$, the estimation error $|\hat{T}_{ii}-T_{ii}|$ is at the same order as $|\hat{c}_i - c_i|$, i.e. $O(\sqrt{\frac{\ln(1/\delta)}{N}})$.

C. More Discussions

C.1. Soft 2-NN Label Clusterability

The soft 2-NN label clusterability means one's 2-NN may have a certain (but small) probability of belonging to different clean classes. Statistically, if we use a new matrix $\boldsymbol{T}^{\text{soft}}$ to characterize the probability of getting a different nearest neighbor, i.e. $T^{\text{soft}}_{ij} = \mathbb{P}(Y_2 = j|Y_1 = i) = \mathbb{P}(Y_3 = j|Y_1 = i)$, the second-order consensuses become $\boldsymbol{c}^{[2]}_r := (\boldsymbol{T} \circ (\boldsymbol{T}^{\text{soft}}\boldsymbol{T}_r))^\top \boldsymbol{p}$ and the third-order consensuses become $\boldsymbol{c}^{[3]}_{r,s} := (\boldsymbol{T} \circ (\boldsymbol{T}^{\text{soft}}\boldsymbol{T}_r) \circ (\boldsymbol{T}^{\text{soft}}\boldsymbol{T}_s))^\top \boldsymbol{p}$. Specifically, if $T^{\text{soft}}_{ij} = e, \forall i \neq j$ and $T^{\text{soft}}_{ii} = 1 - (K-1)e, 0 \leq e < 1/K$, where e captures the small perturbation of the 2-NN assumption, our solution will likely output a transition matrix that affects the label noise between the effects of $\boldsymbol{T}^{\text{soft}}\boldsymbol{T}$ and $\boldsymbol{T}$. The above observation informs us that our estimation will be away from the true $\boldsymbol{T}$ by at most a factor e. When $e = 0$, we recover the original 2-NN label clusterability condition.

C.2. Local $\boldsymbol{T}(X)$

Sparse regularizer Compared with estimating one global $\boldsymbol{T}$ using the whole dataset of size N, each local estimation will have access to only M instances, where $M \ll N$. Thus the feasibility of returning an accurate $\boldsymbol{T}(x_n)$ requires more consideration. In some particular cases, e.g., HOC Local in Table 1, when $\boldsymbol{p}$ is sparse due to the local datasets, we usually add a regularizer to ensure a sparse $\boldsymbol{p}$, such as $\sum_{i\in[K]} \ln(c_i + \varepsilon), \varepsilon \to 0_+$, where c_i is the i-th element of $\boldsymbol{p}$. Note the standard sparse regularizer, i.e. ℓ_1-norm $\|\boldsymbol{p}\|_1$, could not be applied here since $\|\boldsymbol{p}\|_1 = 1$. Therefore, with a regularizer that shrinks the search space and fewer variables, we could get an accurate estimate of $T(X)$ with a small M.

Other extensions Even with M-NN noise clusterability, estimating $\boldsymbol{T}(X)$ for the whole dataset requires executing Algorithm 1 a numerous number of times ($\sim N/M$). If equipped with prior knowledge that the label noise can be divided into several groups and $\boldsymbol{T} = \boldsymbol{T}(X)$ within each group (Xia et al., 2020b; Wang et al., 2021), we only need to estimate $\boldsymbol{T}$ for each group by treating instances in each group as a local dataset and directly apply Algorithm 1. As a preliminary work on estimating $\boldsymbol{T}$ relying on clusterability, the focus of this paper is to provide a generic method for estimating $\boldsymbol{T}$ given a dataset. Designing efficient algorithms to split the original dataset into a tractable number of local datasets is interesting for future investigation.

C.3. Feasibility of Assumption 1 and Assumption 2

1. Denote the confusion matrix by $\boldsymbol{C}[h]$, where each element is $C_{ij}[h] := \mathbb{P}(Y = i, h(X) = j)$ and $h(X) = j$ represents the event that the classifier predicts j given feature X. Then the noisy confusion matrix could be written as $\widetilde{\boldsymbol{C}}[h] := \boldsymbol{T}^\top \boldsymbol{C}[h]$. If $\boldsymbol{T}$ is non-singular (a.k.a. invertible), statistically, we can always find the inverse matrix $\boldsymbol{T}^{-1}$ such that the clean confusion matrix could be recovered as $\boldsymbol{C}[h] = (\boldsymbol{T}^{-1})^\top \widetilde{\boldsymbol{C}}[h]$. Otherwise, we may think the label noise is too "much" such that the clean confusion matrix is not recoverable by $\boldsymbol{T}$. Then learning $\boldsymbol{T}$ may not be meaningful anymore. Therefore, Assumption 1 is effectively ensuring the necessity of estimating $\boldsymbol{T}$.

2. We require $T_{ii} > T_{ij}$ in Assumption 2 to ensure instances from observed class i (observed from noisy labels) are informative (Liu & Chen, 2017). Intuitively, this assumption characterizes a particular permutation of row vectors in $\boldsymbol{T}$. Otherwise, there may exist $K!$ possible solutions by considering all the permutations of K rows (Liu et al., 2020).

D. More Detailed Experiment Settings

D.1. Generating the Instance-Dependent Label Noise

In this section, we introduce how to generate instance-based label noise, which is illustrated in Algorithm 2. Note this algorithm follows the state-of-the-art method (Xia et al., 2020b; Zhu et al., 2021). Define the noise rate (the global flipping rate) as η. To calculate the probability of x_n mapping to each class under certain noise conditions, we set sample instance flip rates q_n and sample parameters W. The size of W is $S \times K$, where S denotes the length of each feature.

First, we sample instance flip rates q_n from a truncated normal distribution $\mathbf{N}(\eta, 0.1^2, [0, 1])$ in Line 2. The average flipping rate (a.k.a. average noise rate) is η. q_n avoids all the instances having the same flip rate. Then, in Line 3, we sample parameters W from the standard normal distribution for generating the instance-dependent label noise. Each column of W acts as a projection vector. After acquiring q_n and W, we can calculate the probability of getting a wrong label for each

Algorithm 2 Instance-Dependent Label Noise Generation

Input:

1: Clean examples $(x_n, y_n)_{n=1}^N$; Noise rate: η; Size of feature: $1 \times S$; Number of classes: K.

Iteration:

2: Sample instance flip rates q_n from the truncated normal distribution $\mathcal{N}(\eta, 0.1^2, [0, 1])$;

3: Sample $W \in \mathcal{R}^{S \times K}$ from the standard normal distribution $\mathcal{N}(0, 1^2)$;

for $n = 1$ to N **do**

4: $p = x_n \cdot W$ *// Generate instance dependent flip rates. The size of p is $1 \times K$.*

5: $p_{y_n} = -\infty$ *// Only consider entries that are different from the true label*

6: $p = q_n \cdot \texttt{SoftMax}(p)$ *// Let q_n be the probability of getting a wrong label*

7: $p_{y_n} = 1 - q_n$ *// Keep clean w.p. $1 - q_n$*

8: Randomly choose a label from the label space as noisy label $\tilde{y}_n$ according to p;

end for

Output:

9: Noisy examples $(x_i, \tilde{y}_n)_{n=1}^N$.

instance(x_n, y_n) in Lines 4 – 6. Note that in Line 5, we set $p_{y_n} = -\infty$, which ensures that x_n will not be mapped to its own true label. In addition, Line 7 ensures the sum of all the entries of p is 1. Suppose there are two features: x_i and x_j where $x_i = x_j$. Then the possibility p of these two features, calculated by $x \cdot W$, from the Algorithm 2, would be exactly the same. Thus the label noise is strongly instance-dependent.

Note Algorithm 2 cannot ensure $T_{ii}(X) > T_{ij}(X)$ when $\eta > 0.5$. To generate an informative dataset, we set $0.9 \cdot T_{ii}(X)$ as the upper bound of $T_{ij}(X)$ and distribute the remaining probability to other classes.

D.2. Basic Hyper-Parameters

To testify the classification performance, we adopt the flow: 1) Pre-training → 2) Global Training → 3) Local Training. Our HOC estimator is applied once at the beginning of each above step. Each training stage re-trains the model. In Stage-1, we load the standard ResNet50 model pre-trained on ImageNet to obtain basic representations. At the beginning of Stage-2 and Stage-3, we use the representations given by the current model. All experiments are repeated three times. *HOC Global* only employs one global $\boldsymbol{T}$ with $G = 50$ and $|E| = 15k$ as inputs of Algorithm 2. *HOC Local* uses 300 local matrices (250-NN noise clusterability, $|D_{h(n)}| = 250$, $G = 30$, $|E| = 100$) for CIFAR-10 and 5 local matrices (10k-NN noise clusterability, $|D_{h(n)}| = 10k$, $G = 30$, $|E| = 5k$) for CIFAR-100. Note the local matrices may not cover the whole dataset. For those uncovered instances, we simply apply $\boldsymbol{T}$.

Other hyperparameters:

- Batch size: 128 (CIFAR), 32 (Clothing1M)
- Learning rate:
  - CIFAR-10: Pre-training: 0.1 for 20 epochs → 0.01 for 20 epochs. Global Training: 0.1 for 20 epochs → 0.01 for 20 epochs. Local Training: 0.1 for 60 epochs → 0.01 for 60 epochs → 0.001 for 60 epochs.
  - CIFAR-100: Pre-training: 0.1 for 30 epochs → 0.01 for 30 epochs. Global Training: 0.1 for 30 epochs → 0.01 for 30 epochs. Local Training: 0.1 for 30 epochs → 0.01 for 30 epochs → 0.001 for 30 epochs.
  - Clothing1M: 0.01 for 25 epochs → 0.001 for 25 epochs → 0.0001 for 15 epochs → 0.00001 for 15 epochs (Pre-training, Global training, and local training)
- Momentum: 0.9
- Weight decay: 0.0005 (CIFAR) and 0.001 (Clothing1M)
- Optimizer: SGD (Model training) and Adam with initial a learning rate of 0.1 (solving for $\boldsymbol{T}$)

For each epoch in Clothing1M, we sample 1000 mini-batches from the training data while ensuring the (noisy) labels are balanced. The global $\boldsymbol{T}$ is obtained by an average of $\boldsymbol{T}$ from 5 random epochs. We only use $\boldsymbol{T}(X) = \boldsymbol{T}$ in local training. Estimating local transition matrices using HOC on Clothing1M is feasible, e.g., assuming M-NN noise clusterability, but it may be time-consuming to tune M. Noting our current performance is already satisfying, and the focus of this paper is on the ability to estimate $\boldsymbol{T}$, we leave the combination of $\boldsymbol{T}(X)$ with loss correction or other advanced techniques for future

Algorithm 3 Local Datasets Generation

Input:

1: Maximal rounds: G'. Local dataset size: L. Noisy dataset: $\widetilde{D} = \{(x_n, \tilde{y}_n)\}_{n\in[N]}$. Noisy dataset size: $|D|$.

Iteration:

2: Initialize the $|D|$-dimensional index list: $S = \mathbf{1}$

for $k = 1$ to G' **do**

if(size($S[S > 0]$) > 0) **then**

3: $\text{Idx}_{\text{selected}} = \texttt{random.choice}(S[S > 0])$ *// Choose a local center index randomly from the unselected index of $\widetilde{D}$.*

else

4: $\text{Idx}_{\text{selected}} = \texttt{random.randint}(0, |D|)$ *// If the selected index has covered $\widetilde{D}$, we choose local center randomly.*

end if

5: $\text{Idx}_{\text{local}} = \texttt{SelectbyDist}(\text{Idx}_{\text{selected}}, L)$ *// Select the index of L features closest to $\text{Idx}_{\text{selected}}$.*

6: $S[\text{Idx}_{\text{local}}] = -1$ *// Mark the state of the selected index in S to avoid duplicate selection.*

7: $\widetilde{D}_k = \widetilde{D}[\text{Idx}_{\text{local}}]$ *// Build a local dataset by selecting $(x_i, \tilde{y}_i), i \in Idx_{local}$.*

end for

Output:

8: Local Datasets $\widetilde{D}_k = \{(x_n, \tilde{y}_n)\} \cup \{(x_{n_1}, \tilde{y}_{n_1}), \cdots, (x_{n_M}, \tilde{y}_{n_M})\}, n_i, k \in [L], i \in [M]$.

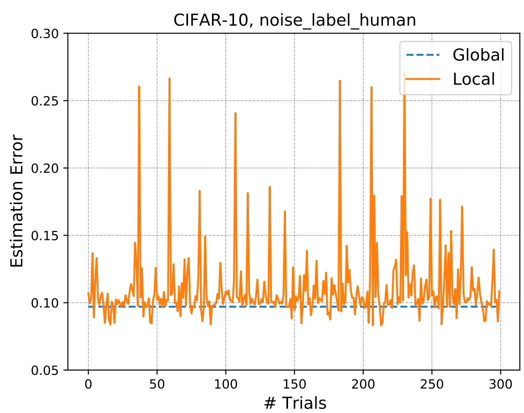

Figure 5. Illustration of the global and local estimation errors. Global estimation error: 0.0970. Local estimation errors: mean = 0.1103, standard deviation = 0.0278.

works.

D.3. Global and Local Estimation Errors on CIFAR-10 with Human Noise

Algorithm 3 details the generation of local datasets. Notice the fact that the i-th row of $\boldsymbol{T}(x_n)$ could be any feasible values when $p_i = 0$, so as the estimates $\hat{\boldsymbol{T}}_{\text{local}}$. In such case, we need to refer to $\boldsymbol{T}$ to complete the information. Particularly, we calculate the weighted average value with the corresponding $\hat{\boldsymbol{T}}$ as

$$\hat{\boldsymbol{T}}_{\text{local}}[i] = (1 - \zeta + \hat{p}_i)\hat{\boldsymbol{T}}_{\text{local}}[i] + (\zeta - \hat{p}_i)\hat{\boldsymbol{T}}[i],$$

where $\hat{\boldsymbol{T}}_{\text{local}}[i]$ and $\hat{\boldsymbol{T}}[i]$ denote the i-th row of estimates $\hat{\boldsymbol{T}}_{\text{local}}$ and $\hat{\boldsymbol{T}}$, $\hat{p}_i$ denotes the estimated clean prior probability of class-i given the local dataset. We use $\zeta = 1$ for local estimates of CIFAR-10, and $\zeta = 0.5$ for local estimate of CIFAR-100.

Figure 5 illustrates the variation of local estimation errors on CIFAR-10 with human noise using HOC.