We thank all the reviewers for their constructive feedback. Below we provide specific responses to each reviewer.

To Reviewer #1

We will re-organize Section 2. The metrics like top-K precision are applicable to S-GWL. When matching MC3 sparse nets, the top-10 precision of S-GWL is 55.26% (higher than its baselines). We will add more results in the paper.

To Reviewer #2

1. *Our main contribution* is proposing a GWL framework with a non-trivial scalable algorithm for graph matching and partitioning. In the following response 2, we further highlight our important improvements ignored by existing work.

2. *The Method* In Fig.1(e), Tables 4 and 5, S-GWL can be slightly worse than GWL on node correctness. It means that the recursive partitioning propagates some errors. However, we didn't observe catastrophic mistakes. One reason is that we just applied 1$\sim$4 recursive steps. Moreover, we suppressed this risk by setting the node distribution of each graph as normalized node degree, and i) considering a regularizer based on node distributions (line 149-155); ii) initializing the node distribution of barycenter graph as the average of the sorted distributions (Eq.(5)). These two improvements make our S-GWL match the nodes in a graph to those with comparable degrees in another graph. The regularizer improves the convergence of our algorithm greatly (Fig.3 in Appendix). We tried different K's and the optimal settings are in the Appendix (Table 6). For different K's, the fluctuations of node correction are $\pm 3.2\%$ in our experiments.

3. *Adjacency Matrices* For weighted graphs (the MC3 nets), the adjacency matrices are continuous. For unweighted graphs (the protein nets), the adjacency matrices are binary. The barycenter is an average of the observed graphs aligned by their optimal transports. The matrix $\bar{\boldsymbol{C}}$ in Eq.(4) is a "soft" adjacency matrix of the barycenter. Its elements reflect the confidence of the edges between the corresponding nodes. We will add its interpretation in the revised paper.

4. *The Node Distribution* $\boldsymbol{\mu} = \frac{1}{\|\boldsymbol{u}\|_1}\boldsymbol{u}$, where $\boldsymbol{u} = [u_i] \in \mathbb{R}^N$, u_i is the number of edges containing i (node degree).

5. *Initialization* Given two graphs, we initialize their optimal transport as the inner product of their node distributions $\boldsymbol{\mu}_s\boldsymbol{\mu}_t^\top$. For barycenter graph, we initialize its node distribution $\bar{\boldsymbol{\mu}}$ by Eq.(5) and its adjacency matrix as $\text{diag}(\bar{\boldsymbol{\mu}})$.

6. *Complexity* We have defined d in line 188, and we will add a footnote below Table 1 to emphasize it.

7. *Results* We will i) define AMI; ii) add two-sided error bars in Fig.1(e); iii) use "$|\mathcal{V}| \times q\%$" in L273; iv) explain NC@1 and NC@All clearly; v) fix typos. Given 3 graphs G_A, G_B, G_C, MultiAlign [53] learns the correspondence $\boldsymbol{T}_{A\to B}$, $\boldsymbol{T}_{B\to C}$, $\boldsymbol{T}_{A\to C}$ with a constraint $\boldsymbol{T}_{A\to C} = \boldsymbol{T}_{A\to B}\boldsymbol{T}_{B\to C}$. It does not consider the case of >3 graphs.

To Reviewer #3

1. We set the node distribution as normalized node degree mainly because of the following reason. When doing multi-graph partitioning, the key step of our S-GWL, the adjacency matrix of the barycenter graph is initialized as a diagonal matrix and its node distribution is estimated by the node distributions of observed graphs ($i.e.$, Eq.(5)). The node distribution based on node degree enhances the consistency of the partitioning across different graphs. For example, given two graphs G_A and G_B, we jointly partition them into two subgraph pairs $\{G_A^1, G_B^1\}$ and $\{G_A^2, G_B^2\}$. If we use uniform node distributions, the barycenter will be initialized with uniform node distribution $[0.5, 0.5]^\top$ and adjacency matrix $0.5\boldsymbol{I}_2$, and we may have an identification problem — G_B^2 can be finally paired with G_A^1.

2. We can view the node distribution as 1D features of nodes, then introducing the proposed regularizer indeed leads to the formulation like the fused GW distance [42, 43]. Given more features, we can use them to compute the $\boldsymbol{C}_{node}$ in the regularizer. We will highlight this point, but as you mentioned our S-GWL is applicable to non-attributed graphs.

3. We tried to partition graphs without the entropy regularizer as you suggested. It achieves at most comparable AMI but spends more time. Actually, our fuzzy partitioning does not have to be a disadvantage — the nodes with split mass may indicate the boundary of partitioning, and their mass reflects their closeness to different communities.

4. As shown in response 1, our S-GWL jointly partitions M graphs $\{G_m\}_{m=1}^M$ into K subgraph groups $\{\{G_m^k\}_{m=1}^M\}_{k=1}^K$, such that we can achieve large-scale graph matching by solving K small matching problems. In Fig.1(d), the transport between each observed graph G_m and the barycenter graph indicates how to partition G_m to $\{G_m^k\}_{k=1}^K$.

5. As reviewer #2 mentioned, the recursive partitioning in S-GWL may propagate matching errors. However, in our experiments the degradation on matching accuracy is often tolerable (1$\sim$2% generally), which means that the matching results does not change a lot compared with that of the GWL without recursive partitioning. Moreover, the acceleration achieved by S-GWL is clear. Additionally, as we highlighted in response 2 to reviewer #2, we make improvements on the algorithm to avoid catastrophic error propagation.

6. The proximal gradient method [48] indeed plays an important role in this work. However, it should be noted that to achieve a scalable GWL method, we improve it by 1) introducing a node-based regularizer, and 2) plugging it into the computation of barycenter graph. The improved algorithm is further combined with a recursive mechanism.

7. For your minor comments: i) We will polish our writing in the revised paper. ii) The graphs in our experiments do not have isolated nodes, and we ignore such nodes in practice. iii) Currently, we assign the node with equally-splitted mass to a cluster randomly. This risk also appears in traditional clustering methods like K-means and GMM. According to our experiments, it does not affect the superiority of our method. iv) The computation of node distribution is in the response 4 to reviewer #2, which will be given in the revised paper. v) For graph partitioning, the K is predefined as in Metis [21]. vi) Assuming that the graphs have comparable size is just for the convenience of complexity analysis.