We thank all reviewers for the comments and the following response will be reflected in the final version.

Complexity (Rev1234): In solving equilibrium equation (2b), (3) for fixed-point state and (8) for gradient, we iterate the equations and the iterations converge when the well-posedness condition is satisfied as we mention in line 199 and 238. At training and test time, the equilibrium equations are iterated to convergence until the 2-norm difference between the LHS and RHS is less then some threshold epsilon. The convergence is guaranteed by well-posedness theory introduced in Section 4.1. In fact, the convergence is exponential both in theory and in practice. In terms of the projection step after a gradient update, projection onto $\|W\|_\infty \leq \kappa < 1$ ball can be decomposed over rows with each row w_i given by a projection onto $\|w_i\|_1 \leq \kappa$, for which a straight forward $O(n \log n)$ algorithm exists using bisection. Duchi et al. (2008) has proposed an $O(n)$ algorithm for projection onto L_1 ball as we mention in line 246. We will offer a subsection with detailed complexity analysis and comparison with other methods in the final version.

FDGNN and GIN not in node classification (Rev1): FDGNN and GIN have mainly discussed their applications in graph classification. Thus we focus on the comparison in the graph classification task. More experiments will be added.

More experiments with Geom-GCN and larger graphs (Rev2): Please find the experimental results in Figure 1 and Table 1. Global methods like Geom-GCN employ additional embedding approaches to capture global information. However, convolutional-GNN-based methods struggle to capture very long range dependency due to the finite iterations they take. Geom-GCN is no exception. We also add a graph classification comparison on a larger and less noisy graph dataset COLLAB where IGNN continues to achieve the best performance.

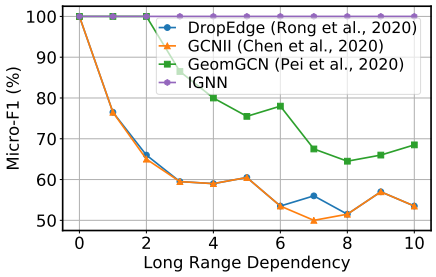

Figure 1: Micro-F_1 (%) performance w.r.t. the length of the chains. Same experimental setting as that for Figure 1. We use 10 layers for GCNII and DropEdge.

IGNN not well motivated to find equilibrium (Rev3): We strongly disagree with the criticism from reviewer 3. IGNN and other recurrent GNN models including the first GNNs (Gori et al., 2005) are all based on the idea of seeking the equilibrium in the graph. Such idea further roots from traditional graph algorithms and metrics including eigenvector centrality (Newman, 2010) PageRank (Page et al., 1999), collaborate filtering as bipartite graph (Zhou, 2018) and more. **IGNN will suffer from 'over-smoothness'. Missing discussion.** (Rev3): We do not explore the direction because our work builds on recurrent GNN which is fundamentally different from convolutional GNN that suffers from 'over-smoothness' problem. IGNN obviously does not suffer from 'over-smoothness' as reflected from experiments where the 'infinitely deep' IGNN even outperforms a variety of models on a range of tasks. Additional experiments show that latest deep models (DropEdge [3] and GCNII [4]) proposed by the reviewer that solve 'over-smoothness' cannot match the IGNN's performance in capturing long range dependency (See Figure 1). **IGNN is weak with 1 linear layer plus phi** (Rev3): The proposed architecture is not weak —- it covers a 100-layer GCN as a special case and many other models. The concise notation allows to formulate those in a way that looks like it has only one layer. See Section 4 and Appendix C for details. **PF eigenvalue (line 172) not defined.** (Rev3): Please find the definition from line 115 in the Preliminary. **State-of-the-art baseline [5] perform better than IGNN** (Rev3): Wrong. Due the space limit, we report performance on 4 graph classification tasks in Table 1 and IGNN outperforms [5] on all of them. We will add the results to the final version. **Does IGNN indeed converge to equilibrium? Why is projected GD needed?** (Rev3): Please find the well-posedness theorems in Section 4.1 which prove convergence of IGNNs that satisfy the well-posedness condition. And indeed it converges in practice. The projection step detailed in Section 4.2 is the essential procedure to enforce such well-posedness condition. We strongly encourage Rev3 to read Section 4 for better understanding.

Underperforming NC1 and DBLP (Rev4): For NCI1, IGNN ranks the second best among the all GNN variants, which is very competitive too. We believe the reason is, though GNNs learn high quality embedding, they can still underperform in distinguishing non-isomorphic (sub-)graphs compared with graph kernels (WL as the best performer). For DBLP, IGNN achieves the second best performance (after DMGI) using only 2 relationships out of 3 to be consistent with our settings on the other two datasets.

Table 1: Graph classification accuracy (%). Results are averaged (and std are computed) on the outer 10 folds.

Data sets	PTC	COX2	PROTEINS	COLLAB
WL	58.0 ± 2.5	83.2 ± 0.2	74.7 ± 0.5	78.9 ± 1.9
GIN	63.7 ± 8.2	–	75.9 ± 3.8	80.1 ± 1.9
GNTK[5]	67.9 ± 6.9	84.4 ± 3.7	75.6 ± 4.2	83.6 ± 1.0
IGNN	$\mathbf{70.1 \pm 5.6}$	$\mathbf{86.9 \pm 4.0}$	$\mathbf{77.7 \pm 3.4}$	$\mathbf{84.6 \pm 2.0}$

Additional details (Rev4): Though we use undirected graphs in the experiments, IGNN is not restricted to undirected graphs. For graph classification, we use mean pooling. Since SSE mainly discusses on learning node embedding and node classification in their paper, we would like to focus the comparison with SSE on node classification. Edge features are highly interesting direction to look at for IGNN. We will try to extend IGNN for it. For 2-layer GCN (15), $\tilde{W}X = [0, W_2; 0, 0][X_2; X_1] = [W_2 X_1; 0]$. We will update the draft accordingly for better illustration.