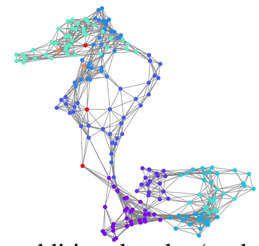

We appreciate all four reviewers' comments. Due to space limit, we focus on addressing major ones. We'd like to reiterate the novelty and significance of our key contribution: SIG-VAE integrates a carefully designed generative model, well suited to real-world sparse graphs, and a sophisticated variational inference network, which propagates the graph structural information and distribution uncertainty to capture complex posterior. SIG-VAE clearly outperforms a simple combination of SIVI (or NF) and VGAE that does not propagate uncertainty in its inference network, and provides much more interpretable latent representations than VGAE. As a flexible generative model, SIG-VAE outperforms SOTA methods in link prediction by a large margin. In addition, it is comparable with SOTA when modified to perform two additional tasks (node classification and graph clustering), even though these two tasks are more suited to supervised learning methods.

Regarding Fig.3 and interpretability (**R1,R4**), we have run HMC to infer posteriors (not shown here due to space limit but will be added into revision), confirming that the SIG-VAE's variational posterior is closer to the HMC inferred posterior, in particular in capturing multi-modality, skewness, and sharp and steep changes. To explain why multi-modality may arise, we used Asynchronous Fluid [Parés et al., 2017] to visualize the Swiss Roll graph by highlighting detected communities with different colors. The three red (two orange) nodes are the nodes with multi-modal (skewed) distributions in Fig. 3 of the paper. These nodes with multi-modal posteriors reside between different communities.

For graph generation (**R1,R5**), comparing the statistics of the generated and training graphs is a standard way for model checking. A well-trained good generative model such as SIG-VAE will have characteristics of generated graphs resemble these of the true one , and paves a way for new applications such as discovering new drugs. Following the instruction of **R5**, we compare KL(node degree distribution of generated graph || that of true). The $\{$SIG-VAE, SIG-VAE (IP)$\}$ results are $\{$3.7e-07, 0.33$\}$ and $\{$1.4e-06, 0.60$\}$ for Cora and Citeseer, respectively, clearly showing the advantage of SIG-VAE using the Bernoulli-Poisson link decoder. We will also add the MMD scores into the revision.

Data splitting (**R1**) is the same as GCN [Kipf & Welling, 2017]. Regarding node classification performance (**R1**), as stated in L301, GCN is a (semi-)supervised model for node classification while ours is a generative model. We will revise L310 to clarify *outperforming* SOTA refers to link prediction. **R1** asked to include two additional baselines. The accuracy (%) of GAT is 83.0, 72.5 & 79.0 for Cora, Citeseer & Pubmed, respectively (note GAT uses 64 hidden features, while the other methods including SIG-VAE use 16). SGC [Wu et al., 2019] gets 81.0, 71.9 & 78.9. SIG-VAE's results are 79.7, 70.4 & 79.3, close to SOTA, despite not being trained for this task in a supervised way.

We provide clarifications for Sec. 3 (**R5**): 1) $\psi$ in eq 2 consists of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ in eqs 3-4. 2)-3) In SIG-VAE, we mix and propagate the representational uncertainty across the graph while in VGAE only deterministic features are mixed. We showed that propagating distributions across graph is beneficial by comparing SIG-VAE with Naive-SIG-VAE and NF-VGAE. Fig. 1 illustrates that neighboring distributions influence the distribution of certain nodes in SIG-VAE. We note that this is not the case for Naive-SIG-VAE, NF-VGAE and VGAE where deterministic features are propagated. 4)-5) The dimension of $\boldsymbol{\epsilon}_u$ is $N \times d^{(n)}$ where $d^{(n)}$ (noise dimension) is a hyperparameter. We add noise to each layer as a part of semi-implicit construction. The dimension of $\mathbf{h}_{u-1}$ is $N \times d^{(h)}_{u-1}$ where $d^{(h)}_{u-1}$ is the number of graph convolutional filters at hidden layer $u-1$. While in eqs 3-4, we used skip connection (concatenation with $\mathbf{X}$), in our experiments (submitted code) we didn't use skip connection since we only used 2 layers for a fair comparison with the baselines. Our experiments for deeper structures of SIG-VAE showed skip connection improves the performance.

GCN-AE (GAE) is the link prediction version of GCN. While we already reported the results of GAE for Table 1 in our submission, we will include the results for Table 2, as **R4** instructed. The mean of AUCs for 10 runs are 93.09, 93.14, 93.74, 72.21 & 55.73 for USAir, NS, Yeast, Power & Router datasets, respectively. The AP results are 95.14, 95.26, 95.34, 77.13 & 67.50. **R4** also asked for a real application. We here include the results on a drug-drug interaction network capturing drug effect change due to the action of another drug. When several drugs are administered together, there might be adverse drug reactions due to drug-drug interactions. It is thus crucial to identify them during drug development. With a similar setup as in the paper, SIG-VAE achieves AUC and AP at 92.51 and 92.81, respectively. For comparison, VGAE gets 90.22 (AUC) and 90.29 (AP), respectively, and GAE gets 90.73 (AUC) and 91.15 (AP). Hyperparameters are inherited from the original paper of each method (**R4**), we will add details in the supplement.

It appears that our notation (using $l$ for latent dimension of $\mathbf{Z}$ and $L$ for number of layers) led to confusion on eq 5 (**R2**). Different from Zhou [2015] that decomposes the Poisson rate in an additive way, here we decompose it in a multiplicative way (additive inside the exponential), which removes the non-negative constraint on $\boldsymbol{z}_i$ and no longer provides the same community structure interpretation as in Zhou [2015]. The AUC results for Power dataset are 94.34, 96.23 & 96.37 for 8-, 16- & 32-dimensional latent space. The AP results are 94.70, 97.28 & 97.42, respectively.

**R2** suggested trying VGAE + VAMP prior, i.e., replacing $p(z) = N(0, 1)$ in VGAE with $p(z) = \sum_k q_\phi(z|u_k)$ , where $(u_1, ..., u_K, \phi)$ will be treated as variational parameters to be optimized. The non-trivial part is how to define $u_k$. In the VAMP prior paper, $u_k$ will be the same dimension as an input data $x_i$, but here the inputs are $X$ and $A$. On the other hand, if VAMP prior helps VGAE, then (semi-implicit) VAMP prior is likely to also help SIG-VAE (semi-implicit VAMP prior can be inferred with doubly SIVI). These potential extensions will be discussed and suggested for future study.