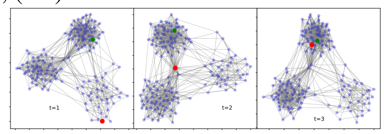
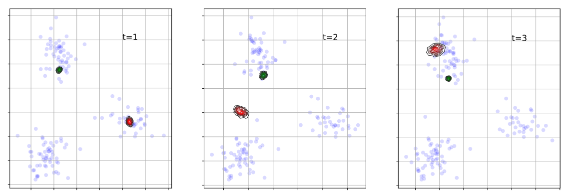
We truly appreciate helpful comments from all three reviewers. Our main modeling and methodological contributions are: 1) A novel generative model, (SI-)VGRNN, is proposed to achieve more interpretable latent representations for dynamic graphs as shown below. To the best of our knowledge, this is the first method modeling uncertainty of node latent representations for dynamic graphs, capturing both topological evolution and dynamic attribute changes simultaneously. 2) By imposing semi-implicit variational inference, we have further extended our original VGRNN model to increase the expressive power of the inferred posterior. 3) Unlike existing dynamic graph models focusing on specific tasks including link prediction and community detection [Kim et al., 2017], (SI-)VGRNN facilitates end-to-end learning of universal latent representations for various graph analytic tasks.

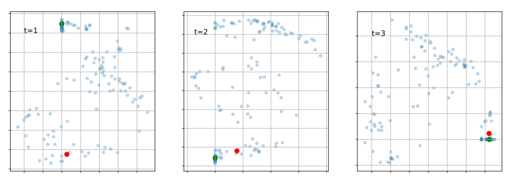


R1 asked how (SI-)VGRNN deals with deletions and additions of nodes. If the graph is growing with addition of new nodes, we assume that the prior of latent representations for the newly observed nodes is zero mean with unit variance Gaussian distribution. If node deletion occurs, we assume that the identity of nodes can be maintained thus removing a node is equivalent to removing all the edges connected to it. More specifically, the sizes of $\mathbf{A}$ and $\mathbf{X}$ can change in time while their latent space maintains across time. Note our model is not designed to predict the occurrence of new nodes.

To show that VGRNN learns more interpretable latent representations (**R1, R3, R4**), we simulated a dynamic graph with three communities in which a node (red) transfers from one community into another in two time steps (1st Fig.). We embedded the node into 2-d latent space using VGRNN (2nd Fig.) and DynAERNN (the best performed baseline; 3rd Fig.). While the advantages of modeling uncertainty for latent representations and its relation to node labels (classes) for static graphs have been discussed in Bojchevski & Gunnemann [2018], we argue that the uncertainty is also directly related to structural evolution of nodes in dynamic graphs.



More specifically, the variance of the latent variables for the desired node increases in time (left to right) colored with red contour. In time steps 2 and 3 (where the node is moving in the graph), the information from previous and current time contradicts each other; hence we expect the representation uncertainty to increase. We also plotted the variance of a node whose community doesn't change in time (colored with green contour). As we expected, the variance of this node does not increase over time. We argue that the uncertainty helps to better encode non-smooth evolution, in particular abrupt changes, in dynamic graphs. Moreover, at time step 2, the moving node have multiple edges with nodes in two communities. Considering the inner-product decoder, which is based on the angle between the latent representations, the moving node can be connected to both of the communities which is consistent with the graph topology. We note that DynAERNN fails to produce such an interpretable latent representation. We can also see that VGRNN can separate the communities in the latent space more distinctively than DynAERNN.



R4 asked what additional information $\mathbf{Z}_t$ provides in Eq. 4: While Eq. 2 constructs the "prior" distribution for $\mathbf{Z}_t$, as conditioned on the state variable $\mathbf{h}_{t-1}$, the posterior of $\mathbf{Z}_t$ has been fed to $\mathbf{h}_t$ in recurrence step, i.e. Eq. 4. Note that the posterior of $\mathbf{Z}_t$ has been inferred based on the information of $\mathbf{A}_t$, $\mathbf{X}_t$ and $\mathbf{h}_{t-1}$, i.e. Eq. 6. From this point of view, the information of $\mathbf{Z}_t$ is more than $\mathbf{h}_{t-1}$. We have to feed $\mathbf{h}_{t-1}$ in Eq. 4 to maintain the RNN structure.

R4 also asked about reconstructing node attributes. As (SI-)VGRNN contribution is to have a model for diverse dynamic graph analytic tasks, the main goal of our method is node embedding. Hence, we are only interested in reconstructing the graph topology instead of the node attributes. This is a common practice in node embedding methods that use node attributes for better node embedding. Potential extensions with other decoders can be integrated with (SI-)VGRNN to construct the node attributes if needed. Regarding the dimension of variables (**R4**), as (SI-)VGRNN is a node embedding method for dynamic graphs, each node is embedded to a point in the latent space. Hence, the first dimension of $\mathbf{X}_t$ and $\mathbf{Z}_t$ are the same and the second dimension of $\mathbf{Z}_t$ is user specified latent dimension. If we reduce the first dimension of $\mathbf{Z}_t$, it would be "graph embedding" method rather than a "node embedding" technique, which is an interesting extension to our work.

Regarding the advantages of our work compared to related work (**R1**): 1) Dynamic network embedding is pursued with various techniques such as matrix factorization [Zhu et al.,2016], deep learning [Seo et al., 2016], and random walks [Yu et al., 2018], many of which are task specific methods and do not focus on representation learning. 2) Most existing methods either capture topological evolution or attribute changes to learn dynamic node embeddings [Yang et al., 2017;Sarkar et al., 2007] but only a few model both changes simultaneously [Trivedi et al., 2019]. 3) None of the existing methods model the uncertainty of the latent representations. While generative models in form of parametric temporal point processes [Trivedi et al., 2017] and deep temporal point processes [Trivedi et al., 2019] have been used for modeling dynamic graphs, to the best of our knowledge, (SI-)VGRNN is the first variational based deep generative model for representation learning of dynamic graphs. A more comprehensive related work section will be added.