We would like to thank the reviewers for their time and helpful notes.

General Comments:

The use of Neural ODE framework does add overhead for ExNODE. Depending on the solver used, the running time can be quite different. `RK4` solver is considerably faster than adaptive solvers like `dopri5`, but sometimes it leads to numerical issues. We use `dopri5` for flow models and `RK4` for classification models. The generative flow models could take roughly 4 days on one TITAN XP GPU, while the classification converges within couple hours (Shown in Fig. 1). We will add some comments on this issue in the camera-ready version.

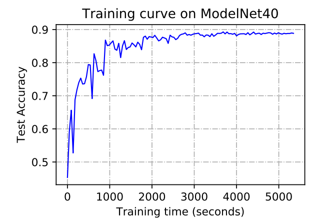

Figure 1: The testing accuracy over training time on ModelNet40 using 1000 points.

The temporal set modeling is a novel exploratory task, which could be an interesting future direction. It also has impactful applications such as modeling the traffic flow. As pointed out by Reviewer #2, the RNN encoder cannot deal with irregular time steps; thus other temporal architecture may be of use. The extrapolation is indeed harder than interpolation. We suspect that is because the VAE model is not trained to generalize beyond the seen time steps.

Reviewer 1:

Please refer to general comments for discussions about run time and temporal experiments.

Thank you for pointing out this very interesting paper. The Lipschitz continuous constraint is an important factor and we will add some discussion in the camera-ready version.

It is a good idea to use latent code with the same dimension. We experimented with exchangeable latent codes, where z is another set with the same cardinality encoded from x using ExNODE. However, we found it hard to learn and the generated samples do not look good. We will inspect into this issue in the future work.

L74-82: We will add the description about constraints for deepset.

L248: We rotate the image and then sample 50 points at each time step independently.

Reviewer 2:

Please see general comments for discussions about run time and temporal experiments.

As you said, the decoder for $p(x_t \mid z_t)$ can use other architectures, like deepsets or set transformer, but it will require the use of distance-based objectives, like the earth mover distance or Chamfer distance. Here, we employ the ExNODE based generative flow model to simplify training so that we can directly maximizing the conditional likelihood.

The method you describe seems like a particle flow model, which is interesting and was considered in the early stage. We found that learning temporal correspondence over sets is surprisingly difficult. We will inspect into this issue in the future work.

For ModelNet40 with 1000 points, ExNODE gets 89.32, while PointNet++ gets 90.7, which is close, however ExNODE uses much fewer parameters.

Reviewer 3:

Please see general comments for discussions about run time.

Intuitively, the permutation equivariant network that parametrizes the drift function should be able to learn the intradependencies. Showing an animation is a good idea and would provide further insights. We will add one in the supplementary material for camera-ready version.

Reviewer 4:

One advantage of using continuous normalizing flow for set modeling is its invertibility. We do not need to design special structures, like coupling transformation, to guarantee invertible. We can basically use any architecture as long as they are permutation equivariant.

We will add some discussions about limitations of ExNODE in the camera-ready version. As shown in general comments, the computation would be one of the limitations. It would be even more expensive for high dimensional sets, like sets of images.