We thank the reviewers for their thoughtful comments and suggestions. We will incorporate them in our revised version. Below, we address the main questions and concerns that were raised in the reviews.

(R1, R3, R4) "...compare the computational complexity, or the actual computational time of the models..." This is a great suggestion. Table 1 compares the training time for all of the models on the particle physics experiment. All models ran on the same hardware. Stopping criteria is after 20 epochs with no f1-score improvement. Table 2 presents the computational complexity analysis of our method **S2G** for the graph ($k = 2$) case in two scenarios: sparse and dense tensors. We assume that the S2G model represents a $f : \mathbb{R}^{n\times d} \to \mathbb{R}^{n^2}$ function, and that the feature dimension is constant across all layers (e.g., taking it to be the maximum across all layers). The S2G model is composed of the following functions: $\boldsymbol{\phi} : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}, \boldsymbol{\beta} : \mathbb{R}^{n\times d} \to \mathbb{R}^{n^2\times 2d}, \boldsymbol{\psi} : \mathbb{R}^{n^2\times 2d} \to \mathbb{R}^{n^2}$. We will add both tables to the final version.

Model	Epochs	Run-time (minutes)
S2G	193	62
S2G+	139	47
GNN	91	21
SIAM	77	24
SIAM3	22	322
MLP	132	22

Table 1: training time comparison between models. Middle column states the number of epochs needed for training using early stopping. Right column states the total training time in minutes.

	Function	Computational	Memory
Dense	$\boldsymbol{\phi}$	$O(n \cdot d^2)$	$O(n \cdot d)$
	$\boldsymbol{\beta}$	$O(n^2 \cdot d)$	$O(n^2 \cdot d)$
	$\boldsymbol{\psi}$	$O(n^2 \cdot d^2)$	$O(n^2 \cdot d)$
Sparse	$\boldsymbol{\phi}$	$O(n \cdot d^2)$	$O(n \cdot d)$
	$\boldsymbol{\beta}$	$O(e \cdot d)$	$O(e \cdot d)$
	$\boldsymbol{\psi}$	$O(e \cdot d^2)$	$O(e \cdot d)$

Table 2: Complexity analysis. For the sparse case, we assume to have e edges.

(R1) "...the method may not be best suited computationally when there is high sparsity ... an incident structure is a more compact representation..." Great Suggestion! proposing a set-to-incidence architecture is an interesting direction, however it is a slightly different problem in nature. We can list it as a future work direction.

(R1) "...add a figure for section 4.2 which could explain the setup towards proving theorem 3." We will make an honest effort to add an illustration explaining the structure of equation 6 which could be seen as the basis of the subsequent proofs.

(R2) "I don't see how the set2graph formulation is clearly justified." Indeed learning to cluster can be formulated in several ways and not necessarily as a set2graph function. Having said that, our focus is on **universal** set-2-graph functions. As far as we are aware, our method is the first one that possess this universality property and therefore an approximate arbitrary continuous clustering functions.

(R2) Why not use link prediction models? Link prediction models mostly use GNNs to predict per-vertex features followed by some predictor acting on pairs of features (see e.g., Link Property Prediciton in Open Graph Benchmark by Hu et al. 2020). Our GNN baseline is therefore a standard link prediction model, where since GNN takes a graph as input we used k-nn graphs with $k \in \{0, 5, 10\}$.

(R3) Is ϕ (from DeepSets) guaranteed to be set-to-set equivalent? We assume the reviewer meant "equivariant". If so, yes, ϕ (DeepSets) is guaranteed to be set-to-set equivariant.

(R3) How should we set the loss function? This is application dependent: for $k = 2$, if we want to learn symmetric edge function then we back-propagate from $\frac{n\cdot(n-1)}{2}$ edge losses, and if directed edge function then $n \cdot (n - 1)$.

(R4) Compared with S2G, S2G+ does not show obvious improvements. Could authors provide more analysis on this phenomenon? The theoretical part of the paper implies that both S2G+ and S2G have universal approximation power, hence equivalent in that aspect. The empirical results show no obvious improvement in practice as well. We decided to include S2G+ in this experiment in order to be sure that there is no real gain (i.e., better generalization) from using the full equivariant function basis of S2G+, that grows exponentially with k. We will make it clearer in the final version.

(R4) I suggest to introduce some learnable operations in β, which may be beneficial for the scalability of the method. A very interesting future work idea, however outside the scope of our paper.

(R4) Can the authors discuss the permutation-equivariance of the proposed method? Our method is permutation equivariant by construction. This is one of the key design considerations, and we will make it clear in the text.

(R4) Visualization results for the generated graphs. We will add examples of generated graphs to the supplementary.