We thank the reviewers for taking the time to read our submission and offer feedback. We would like to emphasize that all reviewers found several positive points of the paper. For example, R1 found the method simple and novel, R2 said that our approach "benefits a large number of relational reasoning tasks," and R3 noted that the empirical evaluation is "diverse and insightful." Below, we discuss what we found to be the three main concerns raised by the reviewers: (1) comparisons to different approaches; (2) clarity of Theorem 1, related assumptions, and relation to SRN; and (3) novelty compared to DSPN. We believe that we can address these issues for a strong camera-ready paper.

(1) Comparisons to different approaches. *R1 discussed FiLM as an alternative.* Since FiLM does not explicitly generate a set, it is difficult to compare against for set generation (although, combining the two is an interesting avenue for future research). Still, we tested FiLM using the same convolution backbone and it achieved similar performance on the furthest task (96.1%), but did much worse on the closest task (91.5%). We will include these results.

R3 discussed Relational Deep RL (RDRL). We want to emphasize that RDRL is iterative across *time steps* and is not using iterative inference to refine a set representation. Thus, in the context of our paper, RDRL has a conceptually similar role to our baseline Relational Network (RN), as they both start by assuming that they have a set of entities. Because of this, we expect RDRL to have the same issues as RNs and would also be a *beneficiary* of the SRN module.

R3 also discussed IODINE. IODINE is an unsupervised object detection method, which has limitations as it is divorced from the downstream task. For this reason, these types of methods are typically not used with the models we consider in our paper (e.g., Relational Networks or C-SWM). For example, the C-SWM paper argues that "typical failure modes [of using methods like IODINE] include ignoring visually small, but relevant features..., or wasting model capacity on visually rich, but otherwise potentially irrelevant features." In addition, methods like IODINE are sensitive to extraneous entities, as it is unaware which entities are relevant to the task at hand. On the other hand, SRN is robust to this issue, as shown by the "cluttered background" experiments in the Appendix. Finally, IODINE is only applicable to images, so we cannot run it on CLUTTR. All that said, we have tried but failed to obtain reasonable results from IODINE during the rebuttal period, largely due to the significant compute required for IODINE.

R3 was also concerned with improvements over C-SWM. As all runs on this task have large epoch-to-epoch variance, we ran updated experiments to reduce variance by selecting the best epoch for each run based on H@5, and use 10 runs. For both experiments, all of our improvements on 5/10 steps are significant, and we will include statistical tests in the paper. For example, on Space Invaders (3 objects, H@1, 10 steps), where SRN performance is most variable, a Welch's t-test provides a 95% confidence interval of 3.7% to 15.7% improvement.

(2) Clarifications on Theorem 1 and how SRN mitigates the responsibility problem. *R3 raised concerns on how realistic the assumptions are, noting MoNet.* This is a subtle point, and we thank the reviewer for raising it. Theorem 1 still applies as long it is *possible* to generate the data from a set continuously, even if there is *some* map that generates the data from an ordered list. Thus, the assumptions are broader than they may appear in the text. A method that uses a canonical ordering for a set (as in MoNet) still needs to reorder the entities discontinuously for some continuous transformation of the objects. We will clarify these ideas and issues in the text surrounding Theorem 1.

R2 and R3 were looking for more intuition on why SRN is effective. The main intuition, as we illustrate in Figure 1, is that SRN is able to model the discontinuity required to handle the responsibility problem (similar to DSPN). Adding the inner optimization loop to a feedforward network enables us to model this discontinuity. In the language of Theorem 1, this means that SRN can model the discontinuous function h for the permutation-invariant map $h \circ g$. To help illustrate this point, we can flesh out the numerical example that we used to generate parts of Figure 1.

R1 had a concern with the proof of Theorem 1. This is a misunderstanding. The output of h is an element of $\mathbb{R}^{n \times d}$, which is equivalent to a "list." Thus, $h \circ g$ is indeed a function from sets to lists. We can clarify this.

(3) Novelty compared to DSPN. *R3 had concerns on framing contributions and naming, with respect to DSPN.* We agree that the primary novelty is demonstrating how resolving the responsibility problem can improve performance and robustness in tasks that use latent set structure (and not in the inner optimization loop itself, which uses similar ideas to energy-based models, representation inversion, and DSPN). Additionally, we think there is a slight misunderstanding here. Although we started from the DSPN source code, we have made significant changes for our setting. Methodologically, SRN removes DSPN's dependency on ground truth sets during training. From the DSPN paper: "when naively using [the image embedding] as input... our decoding process is unable to predict sets correctly from it. To fix this, we add a term to the loss [that depends on ground truth sets]". In other words, DSPN cannot be used in settings other than supervised set prediction. SRN addresses this limitation by refining a predicted set instead of using a shared initialization across all inputs, which also allows easy integration into existing relational reasoning pipelines. Since the goals and methodology are fundamentally different, we think that a different name is appropriate; however, we can think of a name that more accurately reflects the connections.