Major concerns:

To R2&R4 on end-to-end GNN policy learning, and robustness of our two-stage framework w.r.t. detection quality: "end-to-end" represents the method that trains a GNN policy by RL directly, without refactorization. Firstly, the end-to-end method is not as robust as our two-stage framework. According to recent experiments, we can still generalize to 10 dots Pacman using a low recall detector (50% objects are missing) or a low precision detector (25 false positives are presented), with $\pm 8\%$ performance fluctuation. However, the end-to-end method suffers from noisy detections. With a low recall detector, it cannot converge in the training environment. With a low precision detector, its performance drops by 34% and 46% on 10 dots Pacman-CIFAR and Pacman-ImageNet, respectively. On BigFish, the end-to-end method fails to converge to a reasonable good solution in 200M steps. Secondly, it is more time-consuming to train and RL end-to-end with online object proposal generation. It takes $\sim 4.6\times$ time each training step, if we train a GNN policy by RL with proposal generation, instead of training a plain teacher CNN policy by RL. The training cost of refactorization step is negligible in comparison with the teacher policy training time.

To R1 on ablations of refactorization: As suggested, we train the Relation Net on the same demonstration dataset from a CNN teacher. For Pacman-CIFAR and Pacman-ImageNet, the mean episode reward in the environment with 10 dots increases to 79.05 and 82.21 respectively, close to ours (80.67 and 82.67). However, for BigFish, the Relation Net with refactorization got 27.18 in training set and only 15.68 in test set, significantly worse than ours. All the results are averaged by three runs. They prove that 1) refactorization is also useful for Relation Net; and 2) the advantages of our method come from *both the refactorization process and the detector+GNN architecture design*.

The two paragraphs above show that our two-stage training scheme can improve both the detector+GNN based framework and the Relation Net framework, in comparison to end-to-end training. Particularly, for the former (ours), the two-stage method gives a policy that is more robust to detector quality than end-to-end training. In principle, this two-stage scheme separates the difficulties in policy search (by RL) and in achieving generalizability by finding proper representation and inference method.

To R3 on novelty: Thanks for providing related works. However, most previous works either rely on symbolic inputs or GT objects, or only experiments on visually easy environments (no or very limited variation in fg/bg appearance). And the paper mentioned by R3 only qualitatively showed the consistence of the value function between prediction and humans at several handcrafted unseen states, without executing the policy in a new task. Differently, our settings are much harder, including complex games from public benchmarks like BigFish, and the generalizability is validated by executing the policy. We have to address them by a novel two-stage framework with self-supervised object proposals.

Other concerns:

To R1 on more environments: Thanks for the suggestion. Note that Atari does not provide a direct way to evaluate compositional generalizability studied in this paper since there lacks the targeted compositional variation. We plan to *customize and rewrite* more Atari games, and include more Procgen games, like StarPilot in the revision.

To R1 on feature visualization: L159 is correctly described; however, we will explain the terms more clearly in the revision. *Recon* refers to features from reconstruction (SPACE) and *Task* refers to those from task (Policy GNN). Features are computed in the same way for CIFAR-Task and ImageNet-Task. Our improved SPACE has few false positives (AP@0.25=92.4) on CIFAR, and thus the background cluster is too small to observe in the figure.

To R1 on analysis on Relation Net: Following the suggestion, we visualize the attention map of the Relation Net. We observe that although objects are implicitly attended to, the confidence in the attention map vary dramatically when the number of objects changes. It is inevitable if objects are not explicitly recognized.

To R1, R3 on analysis on data parameters: From Fig 2, we can observe that the images with missing detections are downweighted for training by data parameters. It prevents the model to fit a noisy (or insufficient) data point, which will in general hurts generalizability. Different from Multi-MNIST, good policies on Pacman and BigFish are not sensitive to missing detections. Thus, data parameters do not make much difference on the two games.

To R4 on other baselines: Our method gets the object proposals without additional information compared with the baselines. We do not know any other CNN-based baselines which could use the object proposals.

To R4 on visual difficulty and generalization: To tease out the difficulty of object detection, we apply our framework on Pacman without backgrounds. As expected by R4, our method has similar generalization performance to Relation Net, and is better than CNN (in the 10 dots Pacman environment, our method gets 70.2, Relation Net gets 69.6, CNN gets 12.9, averaged by three different runs). The gap between our method and Relation Net is much larger in the environments with more visual difficulty (Fig 3 in the paper). The comparison shows the robustness of our approach to visual difficulty, which aligns with our goal of achieving *compositional generalizability in more realistic environments with noisy detections*. Note that AIR can not extrapolate the number of objects, which inherently fails to generalize to environments with more objects.

To R4 on related works and typos: Thanks for providing the paper. We will cite it and correct typos in the revision.