We thank all reviewers for the constructive comments. We first present new experimental results requested by the reviewers, and then address reviewers' concern individually. All new results will be included in the revision.

R1 : We added a comparison to **policy sketches** (Andreas et al. '17) on the **Crafting** environment from the same work. HAL significantly outperforms policy sketch because it is off-policy and leverages hindsight relabeling, and also does not require sketch supervision (see Fig. 1, left). This also shows HAL works well on an environment from prior work.

R2 , **R3** : We found that naively training a high-level policy to output language directly works poorly, as policy optimization and language generation destabilize one another. To get around this problem, we pre-trained an autoregressive language model as the head of the policy, resulting in a **high-level policy that generates instructions directly**. This approach can leverage compositionality and does not limit the number of instructions, but (with uniform replay buffer) it currently performs comparably due to the challenges of optimizing the language model. (See Fig. 1, middle.)

R3 : We compare to using a **bag-of-words** representation that ignores the **sequential nature** of instructions. Shown in Figure 1, right, our approach significantly outperforms the bag of words.

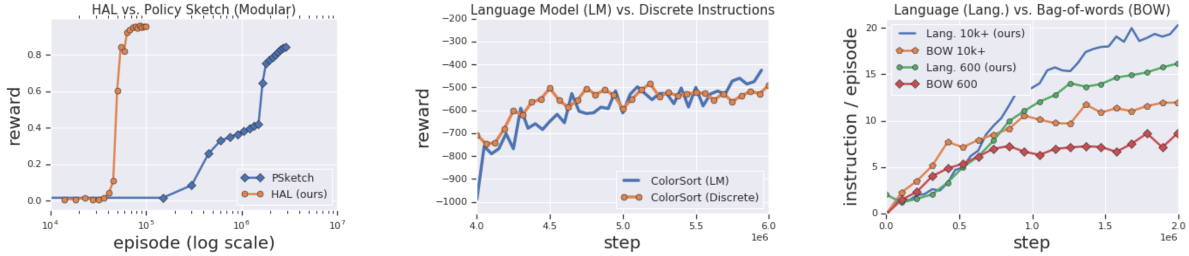

Figure 1: **Left**: Crafting Environment. **Middle**: Language Model High-level Policy. **Right**: Language vs Bag-of-words.

R1 **Re:comparison**: We clarify that only the high-level policy in the more challenging diverse setting uses object state while all other settings operate on RGB inputs. In all settings, HIRO and OC use ground truth state and perform poorly, while HAL works well both with the same ground truth state, and in the non-diverse setting, with pixel observations.

Re: similar methods: We added a comparison to Andreas et al. (see top). Both suggested methods (Andreas et al. '17 and Oh et al. '17) require direct supervision on which instructions to perform for each high-level training task. Our method does not require this kind of high-level task supervision. Hence, HAL is more directly comparable to HRL algorithms like OC and HIRO, which we compare to in the paper.

Re: environment We evaluated HAL on the suggested environment from prior work (see top). As R2 pointed out, the proposed environment exhibits significantly more compositionality than prior RL and language environments. To our knowledge, environments like Montezuma's don't have labeled data or infrastructures for complex language captioning.

Re: baseline failures We used the official implementations of HIRO, and consulted the authors of HIRO to ensure that the method is working as expected. For OC, all the options quickly start to terminate after 1 step, which is a known failure mode of OC (e.g., see Nachum et al. '18). Please see C.4 for more analysis, to which we will add more.

R2 **Re: analysis** We hypothesize that language achieves superior zero-shot generalization because the language in the training and test instructions obey the same *grammar* that specifies how concepts combine. We will discuss this further in the revised version. Successful instructions can be found in the captions of the videos on the supplementary website.

Re: failure modes Some notable failure modes are the following. 1. Increasing visual diversity (e.g. texture, sizes) poses a challenge, possibly due to the model capacity. 2. When the high-level policy outputs instructions that contain non-existent objects, the low-level policy tends to fail to induce any changes, causing the high-level policy to keep giving the same instruction (Diverse setting's videos have a few examples of this). We will add more detailed analysis.

Re: diversity The environment readily supports more instructions that will be available at release. At 1 million steps, HIR achieves 3.9 instruction/episode (0.2 for random) on the combination of CLEVR's *1, 2 and 3 hop* questions.

R3 **Re: compositionality** We added a comparison to bag-of-words (see top). We clarify that the latent code of a sequence auto-encoder is not a sequence, but a fixed-length continuous vector (Appendix C.2). This is a *latent variable* model, as suggested. We agree this latent representation may not be completely non-compositional, but our experiments showcase the benefit of language over the latent variables. We will make this more clear in the revision.

Re: assumption This work uses instructions generated by the CLEVR engine, which use ground-truth locations of the objects. In principle, these instructions could also be generated by a learned captioning-like model. We will edit Sec. 1 and 3 to make this more clear.

Re: high-level action space OC uses a smaller number of discrete options, as it does not work with large number of options (e.g. 80). HIRO specifies continuous goals in coordinates of the objects, and hence has the same dimensionality (10) which is very sensible for continuous control RL tasks (e.g. Ant has 7 DOF and Humanoid has 17 DOF).

Re: related work Thank you for the suggested references! We will cite and discuss the referenced papers.