We thank the reviewers for their valuable feedback. We will incorporate all clarifications below in the final version.

(R4) "Novelty is limited. It looks like the authors simply apply [Shu et al. 2020] to more challenging data." The key contribution of our work is using weak supervision to accelerate RL. Shu et al. does not consider the RL problem. One challenge in achieving this goal is that we must learn representations on data much more challenging than that used in [Shu et al. 2020]. Thus, a secondary contribution of our work is a set of tricks for scaling [Shu et al. 2020] to more complex tasks (L130-147). Our experiments on twelve visual robotic manipulation tasks demonstrate how using weak supervision can help goal-conditioned RL achieve significant improvements over previous SOTA methods (Section 5.1), as well as produce interpretable policies (Section 5.3). Our ablation study (Section 5.2) also studies the relative importance of disentanglement for goal generation vs. for defining reward functions.

(R4) "Why does the proposed approach work so well?" The learned disentangled representation enables the RL agent to explore along meaningful axes of variation (e.g. object position), while ignoring task-irrelevant state dimensions (e.g. lighting). This results in much faster training of goal-conditioned RL (Fig 4, 5).

(R4) "If no relevant factors are specified by the user [...] Does the disentangled representation work well? Is the proposed approach infeasible?"

- **Is our disentangled approach better than alternative methods for utilizing the supervision?** In the main paper, we compare WSC to 'SkewFit+pred', which is a variant of SkewFit that also optimizes an auxiliary supervised prediction loss on the factor identity (see L227-233 and Fig 5). We find that SkewFit+pred performs worse than our method even though it uses stronger supervision (exact labels). This comparison suggests that disentangling meaningful and irrelevant factors is important for effectively leveraging weak supervision.
- The policy learned by WSC depends on the choice of the user-specified factor indices, $\mathcal{I}$. For example, if '$\mathcal{F}_\mathcal{I}$ = {hand_xy}', then the policy will learn to move the robot arm to different positions. If '$\mathcal{F}_\mathcal{I}$ = {blue_obj_xy, red_obj_xy}', then the policy will learn to move the blue and red objects to different positions. In many cases, it is easier and more scalable to specify the relevant axes of variation along which the agent should do exploration (e.g., "Explore by changing the object XY-position"), than to design reward functions or provide demonstrations.

(R4) "Do user-specified factors alone work well if no disentangled representations are learnt?" It's unclear how to use user-specified factors without learning a representation. We did run an ablation of WSC with an oracle representation that was hand-crafted to be disentangled. Experiments with this oracle representation resulted in better performance than with a learned representation. However, such oracle representations are often infeasible to acquire in the real world.

(R4) "Is the time complexity on par with SOTA methods?" Yes, it is the same as SkewFit and RIG.

(R3) L2 distance metric: Similar to findings by [SkewFit, RIG], we found that dense rewards (e.g., L2 distance in latent goal space) work better than sparse indicator rewards for training goal-conditioned RL. We will include an ablation study of WSC using different distance metrics in the final version.

(R3) "What will happen if the relevant factors include [features] like color?" Our method explores and learns to achieve goals that vary the relevant factors. If color is specified as a relevant factor, then the agent will attempt to change the color of its environment (e.g., by turning on colored lights or painting objects in the scene).

(R3) Clarification about notation: Given any vector $v \in \mathbb{R}^K$, we use $v_\mathcal{I} \in \mathbb{R}^{|\mathcal{I}|}$ to denote the subvector extracted from z using the subindices $\mathcal{I} \subseteq \{1, \dots, K\}$. Given an observation s, the encoder outputs a latent vector $z := e(s)$ in the disentangled latent space, $\mathcal{Z} \subseteq \mathbb{R}^K$. We use $Z_\mathcal{I} \subseteq \mathbb{R}^{|\mathcal{I}|}$ to denote the latent space restricted to the indices in $\mathcal{I}$. The true factor value of the current observation, $f_\mathcal{I}(s)$, is not observed by the agent, and is only used to evaluate the true goal distance: $d(f_\mathcal{I}(s), f^*_\mathcal{I})$.

(R1) Paper presentation: (1) Thanks for the helpful feedback! We will move details about dataset generation (Appendix B.1) and MuJoCo environment factors (Appendix B.2) to the main text. (2) There was a typo: The encoder objective should *maximize* $e(z|G(z))$, to approximately invert the generator.

(R1) "Why [Shu et al. 2020] over other [algorithms]?" WSC is agnostic to the underlying representation learning algorithm. The main contribution of our work is to show how weak supervision can accelerate goal-conditioned RL. In our experiments, we chose to use [Shu et al. 2020] because it is provably guaranteed to recover the true disentangled representation under mild assumptions.

(R1) "Could the dataset be used to evaluate encoders?" We used a test dataset (drawn from the same distribution as the training set) to evaluate the learned representations of WSC and VAE (SkewFit) in Tables 3, 4, 5, 10b. [Shu et al. 2020] provides other useful eval metrics for disentanglement (e.g. SAP score)[1].

[1] `https://github.com/google-research/disentanglement_lib/tree/master/disentanglement_lib/evaluation/metrics`