Common We thank the reviewers for their thorough and helpful feedback. The reviewers noted the importance of the problem and the generality and elegance of the proposed solution. We address their concerns below, and incorporate all our clarifications into the paper.

Binary goal-based future task rewards. Note that we do *not* make this assumption for the *current* task reward function, which can be arbitrary. We think that simple future task rewards of this form are sufficient to cover a wide variety of future goals and thus effectively penalize side effects. More complex future tasks can often be decomposed into such simple tasks, e.g. if the agent preserves the ability to move boxes 1 and 2 in Soko-coin, then it can also perform a task involving both boxes. Regarding proofs, the value function formula given in Proposition 1 only applies to goal-based future task rewards (otherwise the goal distance is not defined), while the formula for the general case with arbitrary future task rewards is much messier. Binary goal-based rewards allow us to cover the space of future tasks while keeping the theory simple, so we do not consider this assumption a significant limitation.

Gridworlds. Gridworld evaluation is common in this research area, e.g. in references [20] (ICLR 2019), [23] (AIES 2020), [24] (IJCAI 2018). We agree with the importance of evaluating on more complex environments in future work.

R1 **Section 4 clarity.** We add an algorithm (Figure 1) to illustrate the proposed approach. The algorithm for Section 2 can be obtained by removing lines 1, 7, 11, and setting line 5 to "if $s_t = g_i$: set $\mathbf{r_i(s_t)} := \mathbf{1}$, break" (marked in blue).

Figure 1: Algorithm for Section 4

```
1:  Set $s'_0 := s_0$
2:  for $T = 0$ to $T_{max}$ do
3:      Draw task $i \sim F$
4:      for $t = T$ to $T + T_{max}$ do
5:          if $s_t = s'_t = g_i$: set $\mathbf{r_i(s_t, s'_t)} := \mathbf{1}$, break
6:          if $s_t \neq g_i$: $a_t \sim \pi_i(s_t)$, $s_{t+1} \sim p(s_t, a_t)$
7:          if $s'_t \neq g_i$: $a'^*_t \sim \pi^*_i(s'_t)$, $s'_{t+1} \sim p(s'_t, a'^*_t)$
8:      end for
9:      if $s_T$ is terminal: break
10:     $a_T \sim \pi(s_T)$, $s_{T+1} \sim p(s_T, a_T)$
11:     $a'_T \sim \pi'(s'_T)$, $s'_{T+1} \sim p(s'_T, a'_T)$
12: end for
```

Section 3 typos. In Definition 3, the reference to the baseline policy should be omitted, and $V_\pi(s_{t+1})$ should be $V^0_\pi(s_{t+1})$. In Example 1, the V^*_i formulas are backwards - they should be $V^*_0(s_1) = 0$ and $V^*_1(s_0) = \gamma$. We thank R1 for pointing out these errors in the writeup, and apologize for the confusion.

Optimal policies. There may be a misunderstanding here - our assumption is that we can find an (approximately) optimal policy for any subgoal with a well-defined reward function (such as reaching a single goal state). This is orthogonal to the reward specification problem we highlight - that it is difficult to define a good reward function in the first place. The future tasks framework addresses this problem by allowing us to define a reward function for the goal of avoiding side effects. Note that in practice our method does not require actually running an optimal policy for every subgoal - we only need to compute the value function formula in Proposition 1, which can be approximated using UVFA.

Notation. We can reduce notation overloading for value functions by renaming V^0_π to W^π.

Goal states. (Reachability) We do not enforce that goal states are reachable. If a goal is unreachable, then the agent gets no reward for the corresponding future task, and is not penalized since the reference agent also cannot reach this goal. (Sampling) We sample 10 goal states from a stored set of 100 states encountered by the agent. Whenever a state is encountered that is not in the stored set, it randomly replaces another state in the stored set with probability 0.01.

Egg example. For high values of β, it's true that the agent will avoid using the egg in the current task because it may be needed in future tasks. The value of β needs to be set low enough for the agent to succeed at the current task.

Baseline. We agree that it's often not obvious how to choose the baseline policy. Note that the baseline policy represents what happens by default, rather than a safe course of action or an effective strategy for achieving a goal (so the baseline is task-independent). In the car example, the default outcome for the car is sitting in the garage and not getting anywhere. It is certainly true that doing nothing can have bad outcomes (e.g. standing by while a human driver causes a collision). However, since the agent does not cause these outcomes, they don't count as side effects of the agent's actions. The role of the baseline policy is to filter out these outcomes that are not caused by the agent (to avoid interference incentives).

R2 **Comparison to reversibility.** We agree that the reversibility reward covers unforeseen side effects (as long as they make the initial state harder to reach). However, as noted in the introduction, the reversibility reward is not sensitive to the magnitude of these side effects, and thus would not penalize setting the kitchen on fire relative to breaking an egg (since both are irreversible). Note that the future task approach incorporates the reversibility reward as a future task i whose goal state is the initial state ($g_i = s_0$), since the future tasks are sampled uniformly from all possible goal states. Thus, the future task approach covers all the side effects that are covered by the reversibility reward.

Comparison to safe exploration. We agree that just as safe exploration methods are insufficient to address the side effect problem, side effects methods like future tasks are not sufficient to address the safe exploration problem. We would recommend combining these methods in order to address both problems.

Bias-variance trade-off. We assume you are referring to the variance introduced by using an auxiliary reward function that changes as it is being learned during training. We agree that this probably makes the training process more unstable, and may be responsible for some of the variance in the experimental results. We have also tried freezing the auxiliary reward function after an initial period of exploration, but found that this did not reduce the variance in the results.

R3 **Stepwise baseline.** Agreed. We moved the proofs to the appendix and moved this discussion to the main paper.