Thank you for the thoughtful reviews! The main concern seems to be the need for a more thorough contextualisation of OK within the related literature, so we start by addressing this point. Many transfer methods build on the following idea: first learn a parametric representation of a policy $\pi(\cdot|s;\boldsymbol{\theta})$ that captures the structure of a set of tasks, then quickly adapt to a new task by fine-tuning $\boldsymbol{\theta}$ (Da Silva *et al.*, 2012; Gupta *et al.*, 2018) or by learning a policy that uses $\boldsymbol{\theta}$ as actions (Frans *et al.*, 2017; Haarnoja *et al.*, ICML, 2018). We call these *policy-based* methods. One of the central arguments in our paper is that working in the space of *cumulants* (rather than policies) may offer some advantages: it is a robust approach because it captures the *intentions* behind the skills being transferred (lines 38–42) and it can generate options that are not in the policy space "spanned" by their constituents (lines 165–167). Both policy- and cumulant-based approaches should have advantages and disadvantages, so it is desirable that they co-exist in the literature.

All the papers cited by the reviewers describe policy-based transfer methods, with 3 exceptions: [1,2,3] describe approaches to compose policies based on their *value function*. Although [1,2,3] are competitors among themselves, they are *not* direct competitors of OK. OK extends [1] from policies to options in order to get temporal abstraction (the benefits of which are well known). Dealing with termination and initiation of options in a principled way is not a trivial extension. This involved 3 steps: (i) augment the definition of cumulants to depend on histories and to also include a termination action, (ii) show the mapping between the resulting extended cumulants and options (Prop. 1), and (iii) adapt the machinery in [1] to this more general scenario. To the best of our knowledge, the resulting OK framework is *the only way to combine options in the space of cumulants with performance guarantees for general MDPs*. Although OK is not addressing the same problem as [1,2,3], it is reasonable to ask whether (iii) could also be applied to [2,3], as suggested by R1 and R3. To answer this question we implemented a version of OK that uses [2] rather than [1] to combine options. Specifically, we replaced GPE (eq. 6) and GPI (eq. 7) with the composition of value functions from [2]: $\tilde{\omega}_e(h) \in \text{argmax}_{a\in\mathcal{A}^+} \sum_j Q_{e_j}^{\omega_{e_j}}(h,a)$. We also implemented a "softmax" version in which $\tilde{\omega}_e(h)$ is computed using eq 2 of [2]. The results in Fig. 1 suggest that GPE and GPI are more effective than [2] in this case. We'll add a more extensive version of this comparison to the paper and also a comparison with the two methods in [3].

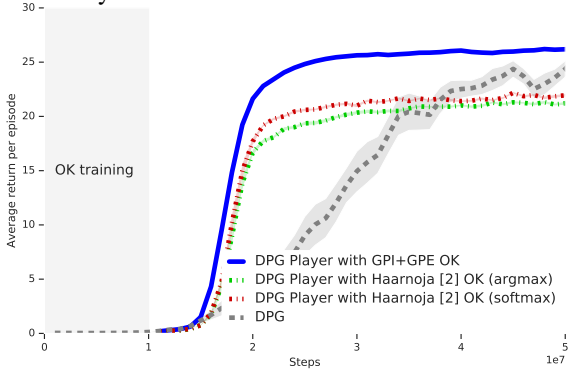

Figure 1: OK using [1] and [2]. Curves are the best result over a set of learning rates and, for [2], also over temperature parameters for the softmax policy.

R1 MAJOR COMMENTS. **(1)** We are not addressing the problem of option discovery, but we believe that the formalism we developed allows for a clean formulation of the problem in the space of cumulants (lines 584–598). **(2)** The composition of value functions proposed in [2] is inherently different from GPE and GPI because an option is never evaluated under another option's cumulant: since there is no GPE, composition is made with $Q_{e_j}^{\omega_{e_j}}(h,a)$ rather than with $Q_e^{\omega_{e_j}}(h,a)$ (compare the above with eqs. 6 and 7). **(3)** These are policy-based transfer methods as defined above, with the associated advantages and disadvantages. **(4)** The information in the appendix is outdated, thank you for pointing that out! After a food item is consumed, we can either reward the termination τ or penalize actions $a \neq \tau$. Although the resulting options are identical, the latter scheme leads to faster learning, and thus it was used for all the experiments. MINOR COMMENTS. **(1)** Q-learning uses only 2 options. This results in fast learning whose curve looks flat at the scale of the other methods' curves. But Q-learning's non-trivial performance shows that it does learn the task.

R2 (1) Some combinations of options are indeed not representable as linear combinations of cumulants. When the weights $\boldsymbol{w}$ are nonnegative, it is instructive to think of GPE and GPI as something in between an `AND` and an `OR` (as cumulants are rewarding in isolation but more so in combination). GPE and GPI cannot implement a strict `AND`, for example. **(2)** t is implicit in the definition of $\mathbb{E}^{\pi}_{s,a}[\cdot]$ (line 73). **(3)** You are correct: the $\max$ operator is applied to each (s,a) independently. We will clarify. **(4)** Suppose that $\mathcal{I}_e = \mathcal{S}$. Since in the states s where $\beta_e(s) = 1$ (*cf.* eq. 5) executing option o_e will have no effect, we simply exclude those from $\mathcal{I}_e$. This allows us to have o_e be fully determined by e, without any extra definitions. **(5)** If you think of the set $\mathcal{Q}_{\mathcal{E}}$ as a *cumulants* $\times$ *options* matrix, it is possible to disassociate these quantities. It is true nevertheless that each option must be evaluated under the cumulants we want to generalize over. The premise is that with a small number of both we can create a very diverse set of behaviours. We'll elaborate in the appendix. **(6)** We had 2 cumulants associated with goods and 1 policy induced by each cumulant, resulting in 2 policies $\times$ 2 cumulants = 4 value functions. **(7)** We will add line patterns as we did in Fig. 1, thanks!

R3 Our main theoretical result, Prop. 1, is largely independent of [1], and we believe its interest goes beyond the scope of this paper. • The options were learned before (see gray area in Fig. 3 and discussion in lines 508–509 of the appendix), but in principle OK and player can be learned together (we are currently working on it). • We kindly ask the reviewer to reconsider their assessment of the significance of the paper in light of the explanations above.

REFERENCES. **[1]** Barreto *et al.* NIPS, 2017. **[2]** Haarnoja *et al.* IEEE ICRA, 2018. **[3]** Hunt *et al.* ICML, 2019.