We thank the reviewers for constructive feedback. We are glad that they found our idea of discovering an entire RL update rule interesting [R2, R3] and recognised our work as a clear distinction from the previous work [R3] and a promising direction [R2]. We are also encouraged that they found our empirical results impressive [R3] and convincing [R2]. We address questions raised by each reviewer below and will incorporate some of the feedback in the revision.

[R1] **"But such an approach for learning RL algorithms has already been used before." "They did not do a good review of the related work." "missed several important works, such as for example those of Francis Maes ..."**
Though we thank the reviewer for pointing out some relevant work, we would like to clarify the difference and potential misunderstandings. First of all, our approach is new and distinct from the prior work in that there has been no prior approach that attempts to discover alternatives to value functions and TD-learning as agreed by [R2, R3]. Secondly, Maes et al. [1], [2] aim to find the formula of the Bayes-optimal policy given hand-designed variables (e.g., $Q(s,a), N(s,a)$) by searching over math operations $\mathcal{F}$, where $p(a|s) = \mathcal{F}(\{Q(s,a), N(s,a)\})$. Thus, the update rules for the variables are **NOT** discovered but hand-designed in Maes et al., whereas our approach discovers an update rule for them: what these variables should be and how to update them. Finally, although our problem could be interpreted as a Bayesian inference problem, it is quite different from the conventional Bayesian RL, because the update rule does not directly interact with the environment unlike the policy in Maes et al. We will cite and discuss them in the revision.

[R1] **"Problem badly formalized." "The main contribution of this paper is how to define the candidate space of your η, something you never define very well."**
Our problem and its objective is clearly defined in Eq. 1. In addition, unlike Maes et al. [1], [2], where the solution space can be expressed by a finite set of operators, the update rule is parameterised by a neural network η in our work. Thus, the solution space of η is determined by the network architecture, which is also clearly defined in the paper.

[R1] **"Simulation results carry out on fairly simple problems and not that convincing."**
ALE (Atari games) is a very challenging benchmark. We believe that the fact that the update rule discovered solely from toy domains can generalise well to Atari games is very interesting and convincing as agreed by [R2, R3].

[R1] **"Main contribution - the way they define the set of RL algorithms - not put forward in a proper way."**
We respectfully disagree. As emphasised in the paper and acknowledged by [R2, R3], our main contribution is rather to show that it is possible to discover an entire update rule that can replace value functions and TD-learning, and that the update rule discovered from toy domains can generalise surprisingly well to complex Atari games.

[R2] **Regarding novelty of the formulation**
We would like to re-emphasise that our formulation is novel in that it is the first to discover alternatives to value functions. In terms of meta-training, we would like to point out that our method is not just a combination of the prior work. Since the problem of discovering an entire update rule is much more challenging than discovering only a policy update rule as in MetaGenRL, we had to develop several new methods (e.g., regularisers, balancing hyperparameters), which turned out to be crucial (see ablation study). We believe that they would be useful for the future work as well.

[R3] **Regarding whether LPG predicts something beyond future rewards and LPG-V**
We agree that predictions are implicitly encouraged to be a function of future rewards. However, we claim that there is still significant room for improvement by discovering a better form of such a function. For example, variations of TD-learning methods (e.g., distributional RL, Γ-net, mixtures of n-step and λ-returns, non-linear reward transformation) have been shown to perform quite differently, even though the underlying principle is the same. We suspect that LPG could discover a more efficient class of TD-learning and even beyond. In fact, Figure 5 shows that LPG captures values at various discount factors, which is already interesting in that none of the RL algorithms maintains values at various discount factors at the same time. This indicates that LPG is doing something different from TD-learning for more efficient bootstrapping. To further clarify, y-vector is mapped to a scalar only within the update rule, but y still maintains richer information compared to the value function, which is why it can capture values at various horizons. Finally, we picked the best LPG-V by tuning it with and without all the tricks (regularisers, balancing hyperparameters). Thus, we believe that the result shows that the discovered prediction semantics is crucial for the performance. In fact, this is also consistent with MetaGenRL's result, where the discovered policy update rule (given value functions) does not outperform the baseline DDPG even on the training environments.

[R3] **Is passing γ necessary? Does LPG still learn with additional information such as state representation?**
Yes. γ is treated as a part of the environment, which is varied during training. So, the optimal update rule depends on the given γ. We expect that giving domain-specific information like state representation is likely to improve LPG on training domains but hurt on unseen domains, which is why we intentionally removed such information. However, adding such information without compromising generalisation performance would be a very interesting future direction.

[1] F. Maes, L. Wehenkel, and D. Ernst, "Automatic discovery of ranking formulas for playing with multi-armed bandits," in *European Workshop on Reinforcement Learning*, Springer, 2011, pp. 5–17.

[2] M. Castronovo, F. Maes, R. Fonteneau, and D. Ernst, "Learning exploration/exploitation strategies for single trajectory reinforcement learning," in *European Workshop on Reinforcement Learning*, 2013, pp. 1–10.