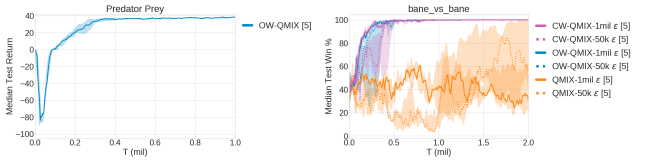
We thank the reviewers for their feedback on our submission. We have fixed a config error when running OW-QMIX on Predator Prey (there is now very little variance). We have also run additional experiments demonstrating significantly better performance of weighted QMIX on another *hard* SMAC map *bane_vs_bane*.



(Left) Corrected experimental results for OW-QMIX on Predator Prey. (Right) Weighted QMIX vs QMIX on `bane_vs_bane` for ϵ annealed over 50k and 1mil.

Reviewer 1: >*"Why not include QTRAN, MADDPG, or MASAC on any of the SMAC experiments?"* We have already included QTRAN, MADDPG and MASAC on the SMAC experiments in Figure 2. Due to their relatively poor performance there we did not run them on the *Super-Hard* SMAC maps (Figures 3 and 4) due to the large computational cost of those experiments.

>*"How does using a QMIX approximation to Q^* work?...And won't this cause the policy induced by Q^* to be different?"* In order to enable tractable maximisation, we use our QMIX approximation (Q_{tot}) to Q^* to suggest the best joint action. In general, the greedy action for Q^* and Q_{tot} can differ during training. However, Theorems 1 and 2 prove that given sufficient training (and an appropriate α) they will be the same.

>*"...the modified architecture uses a hypernetwork layer—does this mean it is restricted to positive weights like QMIX?"* Yes, the first layer of $\hat{Q}^*$'s mixing network is restricted to non-negative weights, but $\hat{Q}^*$ is not restricted to being monotonic due to subsequent layers.

Reviewer 2: >*"The weighted QMIX only modifies QMIX by using a weighting function to get the Q_{tot}, and the two kinds of weighting functions seem too simple, so the contribution seems incremental."* We disagree that the simplicity of the weighting function makes our approach too incremental. The use of a weighting function in order to train a monotonic approximation to a learned unrestricted Q^* is a significant algorithmic change over QMIX. Additionally, we have proven that the two weighting functions we have considered are guaranteed to ensure the maximal joint action is correct (given sufficient training and an appropriate α) in contrast to QMIX which can fail to recover the optimal joint action for the simple matrix game in Table 2. Furthermore, the framework we have introduced for analysing Weighted QMIX can be used to analyse QTRAN and explain its empirical performance.

>*"Extra computation cost also restricts its scalability."* Compared to QMIX, during training we must perform inference and train an additional model (with the same complexity as QMIX). This does **not** restrict the scalability of Weighted QMIX compared to QMIX, as demonstrated by our experiments on `bane_vs_bane` featuring 24 agents. We will include a discussion of the two papers you have provided.

Reviewer 3: >*"The proof of your theory lacks discussion of POMDP settings."* We deliberately restricted our theoretical analysis to the MMDP setting in order to avoid the additional complexity of partial observability. The MMDP setting allows for a cleaner presentation that focuses on our main goal of analysing the effect of the limited representation of QMIX on the learned Q_{tot} (and thus the learned policy).

>*"...performance of QMIX+$\hat{Q}^*$ has a significance difference ... in Figure 8... The use of weighting is not that convinced."* On `3s5z` the weighting does not affect performance, but on `5m_vs_6m` it has a significant effect and on Predator Prey every method without the weighting is unable to solve the task, showing that it is crucial to our method.

>*"In Section 6.2.3, the performance of the Weighted QMIX method is unacceptable."* That's the point: Section 6.2.3 aims to show the limitations of our method, which we believe is important for identifying areas for future research.

>*"The authors argue that the complexity introduced by $\hat{Q}^*$ is responsible for the regression in performance."* Figures 4 and 5 demonstrate a clear performance difference for Weighted QMIX when **only** changing the architecture used to represent $\hat{Q}^*$. This provides evidence that the poor performance is due to the architecture used to represent $\hat{Q}^*$.

>*"...α of weighing Function, although the author gives a basis for selection in the appendix, the value of α seems not to be verified."* Our theoretical results only show that there exists an α which works in **all** cases. They are not intended to provide a method for selecting an appropriate α. We will discuss the selection of α for experiments in the main paper.

>*"... agent's ordering of actions is pointed to be important in representing value functions. But the proposed architecture seems to be incapable for dealing that case."* $\hat{Q}^*$ is capable of representing **any** joint action Q-value function.

>*"... sampling uniformly from a replay buffer does not strictly lead to a uniform weighting schema. However in the realization of Weighted QMIX you provided in Section 5, the loss in Equation 8 also suffers from the same problem."* A uniform weighting is an assumption we make to simplify our analysis (to make it clearer). It is not required for the Deep RL realisation of QMIX or Weighted QMIX.

>*"How does the content of lines 163-167 relate to context?"* Lines 163-167 explain why the failure modes discussed in Section 3.1 are problematic. In particular, they show fundamental limitations of QMIX that cannot be addressed without a significant algorithmic change, even in the idealised setting we consider. We will fix points 4, 5, and 6 on Clarity.

>*"How will input s to the Mix network portion be handled during execution?"* During decentralised execution only the agent parts of Q_{tot} are required. We do not need access to the state, the mixing network, or $\hat{Q}^*$ during execution.