We thank the reviewers for carefully reading our manuscript and providing constructive feedbacks. Below are some comments and clarifications.

Benign Loss in Efficiency

R1: High-level comments (3), Specific comments (3) | R3: Improvement. Both R1 and R3 have named Δ as one possible cause of the small efficiency loss and suggested its relation to the dataset. We would like to first clarify that Δ, by definition, is the maximum trip utility difference across all pairs of edges (v, r) and (v', r), corresponding to the *same* request r, across all requests. This utility equals to value (trip length) – cost (cruising time). Since the same request contributes the same trip value to utility when matched with any vehicle, our Δ directly translates to the cruising time of vehicle obtained from the service constraints in waiting/delay time. For instance, when we set waiting time constraint $\Omega = 120$s, the maximum difference in trip cost and thus, Δ, is also 120s.

We further note that a small Δ would *not* cause the outcomes optimized for two different objectives to align completely. Even in the extreme case, if we plug $\Delta = 0$ into Theorem 3.1 and 3.3, one could still see a clear tradeoff between the two objectives. In other words, even when $\Delta = 0$ the solution optimized for just efficiency may still have very bad fairness.

R1: High-level comments (1), Specific comments (7), Improvement. First, to address R1's questions in high-level comments (1), we do find cases, especially in our multi-period experiments, where there are multiple matchings M with same (or very close) overall utility (i.e., efficiency) but significantly different worst-off utility (i.e., fairness). However, this does not mean that fairness and efficiency will always converge in this framework. In most cases, these two objectives do not align with each other completely. In particular, one can *not* simply use a "max-flow with only regard for overall utility" as R1 suggested. Fig 1a, 2a, 2b show the mildly decreasing trade-off curves. In all these figures, the left-most point is the max-flow solution, meaning that no fairness constraint is imposed. In respective order of single-batch, multi-period single-ride, multi-period ridesharing, the level of fairness we obtained from these solutions are only 16%, 41%, and 14% of optimal fairness. Nontrivial algorithms are needed to obtain solutions with both good efficiency and good fairness. This is exactly the purpose of Algorithm 1.

We had also conducted synthetic experiments for our single-batch setting, as R1 stated, to elicit circumstances where the ratio $\frac{\mathcal{E}(M)}{\mathcal{E}(\text{opt})}$ doesn't plateau. From our observation, there are 3 such circumstances: (1) when there is a denser bipartite graph or more leeway to permutate between different vehicle-request pairs, achieved by relaxing waiting/delay time constraint Ω (thus, increasing Δ) and controlling vehicle starting nodes; (2) when efficient allocation has a small overall trip cost compare to that of a fair allocation; (3) when more requests are dropped by the reassignment to fairer solution; this case happens very rare even in synthetic situations. We will add these discussions to the paper.

Source of Unfairness

R2: Improvement (1). We agree with R2's comments on the neighbouring structure being the potential source of unfairness and we did try to alleviate this problem by (1) controlling the vehicle initialization node; for instance, we sample these locations from the set of nodes that can pick-up at least 10 requests (2) removing requests that could cause vehicles to get stuck to nodes with too few edges, while keeping the degree of removal reasonable. Both were mentioned superficially in the paper due to space constraint. We will add these remarks to the paper.

Static vs. Dynamic

R2: Improvement (2). Our paper does have a dynamic setting component discussed in Section 2.2. Specifically, in this multi-period setting, we allow riders and drivers to arrive and leave dynamically in rounds. We have also tested our algorithm in this multi-period setting in the experiment section. Though this is not the main focus of this paper.

We would be happy to study and discuss the related works R2 listed in our revised version.

Other Specific Questions

R1: High-level comments (5). Our way of redistributing utility is more corrective than preventive; our algorithm will start addressing the fairness issue once the allocated utility corresponding to these 'less-fortunate' drivers are realized.

R1: Specific comments (1). As we mentioned before, the fairest allocation M_{fair} may have bad efficiency. REASSIGN finds a matching M that reconciliates fairness and efficiency by allowing users to choose any desired degree of fairness through the input λ.

R1: Specific comments (2). We do have such results. In the single-batch experiment, we set $|\mathcal{V}| = 1.2|\mathcal{R}|$, and Fig. 1 demonstrates the result.