We are grateful for the insightful comments of all reviewers. We will revise our manuscript carefully to address all clarity issues indicated by the reviewers. In the following, we address the concerns of each reviewer in detail:

Reviewer 1: "If my understanding is correct...Compared to the derandomized RandomGreedy, the algorithm proposed in this work is faster by an r factor" The potential derandomization approach sounds interesting, even though it would have higher time complexity than our approach. As it is unclear how to bridge the gap between the method in [14] and the performance analysis in [11], we will check it in more detail and add a discussion on it to our manuscript.

"The contribution is primarily of theoretical interest, since it focuses on deterministic algorithms and fast algorithms with better approximations are known if randomization is allowed" To the best of our knowledge, the fastest algorithm with an approximation ratio better than $\frac{1}{4}$ is Algorithm 4 in [11], which is randomized and has a 0.283 ratio. This algorithm's time complexity (regarding the queries to value and independence oracles or other general operations) is larger than TwinGreedy by at least an additive factor of $\mathcal{O}(r^3)$ (due to its Line 4). As r can be in the order of $\Theta(n)$, there might exist a tradeoff between accuracy and efficiency. Compared to this algorithm, our approach also has the following features. First, our algorithms can be directly used to address a more general p-set system constraint with good performance bounds (especially for small p). Second, our algorithms can be accelerated to achieve nearly linear running time, as shown in Section 4. We will add more discussions on [11] to our manuscript.

Reviewer 2: "For Fantom, there are...instead of the 1/3 approximation algorithm used in the paper?"We tested Fantom again using the randomized algorithm in [12] with $1/2$ expected ratio. The experimental results are almost the same as before, although Fantom has a larger $1/6$ ratio (in expectation) in such a case. We will revise Section 5 to clarify this.

"How does the performance of the FastTwinGreedy algorithm change as epsilon changes? Is it possible to...How about with TwinGreedy?" In our experiments, the utility of TwinGreedyFast slightly increases when ϵ decreases, and almost does not change when ϵ is sufficiently small (e.g., $\epsilon \leq 0.02$). It is also possible that the utilities of TwinGreedyFast (with a small ϵ) and TwinGreedy outperform Fantom on some data points (but not for every input), and we will add more experimental results about this to our manuscript.

"What is the intuition for...two sets you have three or more sets competing?" If regular greedy is used, we can only get a set S satisfying $f(S) \geq f(S \cup O)/2$, which implies that S has $\frac{1}{2}$ approximation ratio if $f(\cdot)$ is monotone. When $f(\cdot)$ is non-monotone, we cannot use $f(S \cup O) \geq f(O)$, so we have to use $f(O \cup S_1) + f(O \cup S_2) \geq f(O)$ (i.e., Eqn. (10)) to derive the ratio. As two solution sets are sufficient for Eqn. (10) to upper-bound $f(O)$ based on submodularity of $f(\cdot)$, introducing more competing sets would not help to improve the ratio but would cause extra time complexity.

"In the experiment section, is there any reason to suspect that wall clock time would differ from number of queries?" The wall clock running time of the algorithms does differ from the number of queries, but the experimental results are qualitatively similar: TwinGreedyFast is still faster than the other algorithms by more than an order of magnitude, as most of the running time is spent on oracle queries. We will add more experimental results on this to our manuscript.

Reviewer 3: "I saw that several of the prior works also make the assumption that f is non-negative-is this necessary?" It is generally believed that submodular functions may only be optimized under the non-negativity assumption, so most studies in the literature have adopted this assumption. A good reference (and also a rare exception) for this is "Submodular Maximization beyond Non-negativity: Guarantees, Fast Algorithms, and Applications. ICML 2019". In our paper, we have used the non-negativity assumption in Eqn. (10), i.e., $f(O) + f(O \cup S_1 \cup S_2) \geq f(O)$.

"I also couldn't see where the assumption f(0) = 0 was used..." For simplicity, we have followed many related studies (e.g.,[15][16][31]) to assume $f(\emptyset) = 0$. Actually, the $\frac{1}{4}$ ratio of TwinGreedy still holds when $f(\emptyset) > 0$, by minor revisions to the proof of Theorem 1: (1) In Eqn. (8), change $f(S_1) + f(S_2)$ to $f(S_1) + f(S_2) - 2f(\emptyset)$, as $\sum_{e \in S_i} \delta(e) = f(S_i \mid \emptyset)$ for $i = 1, 2$; (2) Revise the proof in Section A.3 as follows. Suppose that $S_1 \neq \emptyset$ and $S_2 = \emptyset$. According to the greedy rule of the algorithm, we have $f(O \cap S_1 \mid \emptyset) \leq \sum_{e \in O \cap S_1} f(e \mid \emptyset) \leq \sum_{e \in O \cap S_1} \delta(e) \leq \sum_{e \in S_1} \delta(e) = f(S_1 \mid \emptyset)$ and $f(O \backslash S_1 \mid \emptyset) \leq \sum_{e \in O \backslash S_1} f(e \mid \emptyset) \leq 0$. Combining these with $f(O \backslash S_1) + f(O \cap S_1) \geq f(O) + f(\emptyset)$, we get $f(S_1) \geq f(O)$. By similar minor revisions, it can also be shown that the performance bounds of TwinGreedyFast still hold when $f(\emptyset) > 0$. We will revise our paper to clarify these points.

Reviewer 4: "There already exists a deterministic 1/4-eps approximation algorithm by Lee et al. and a randomized 1/4-approximation algorithm that runs in O(nr) time by Feldman et al." We agree. While matching the ratio of these algorithms, our approach also has the following features. First, our algorithms are simple can be accelerated to achieve nearly linear running time. Second, our algorithms can be directly used to address a more general p-set system constraint. Third, our algorithms are "unified" and also perform well under the monotone case. For example, by similar reasoning with Theorems 1-3, it can be seen that TwinGreedy achieves $\frac{1}{p+1}$ ratio for monotone $f(\cdot)$ under a p-set system constraint, which is almost best possible [2]. Fourth, our algorithms are deterministic and can achieve more "stable" practical performance than randomized algorithms (with much lower time complexity), which is corroborated by Section 5. We will revise our manuscript based on the reviewer's comments to make our contributions more clear.