We thank the reviewers for their thoughtful feedback. The major concern was **whether current min-cost flow (MCF) based methods can achieve strong tracking performance and thus worth investing in designing efficient flow solvers**. We would like to take the opportunity to report the evidences and explanations to relieve this concern and emphasize the significance of our work.

First, **the strong performance of MCF based methods is evidenced** by the tracking results on two public benchmarks, MOT17 and KITTI-Car. (1) As shown in Table R1, the MCF based tracking methods (LSST17 and JBNOT) are the top 2 methods among the 32 published works on MOT17 benchmark, according to the MOTA score which is widely used to evaluate tracking performance. It is worthy to mention, among all the 99 submissions including anonymous ones, LSST17 still holds the first place. (2) On KITTI-Car benchmark (Table R2), MCF based methods also achieve quite competitive performance, where MOTBeyondPixels and AB3DMOT take the first two places among all 34 published works according to MOTA score. These comparisons are provided by MOT challenge [R1] and KITTI challenge [R2], where the corresponding references can be found.

Table R1: Top 5 methods on MOT17*

	MOTA	IDF1	MT	ML	ID Sw.	MCF
LSST17	**54.7**	**62.3**	20.4	40.1	**1243**	YES
JBNOT	52.6	50.8	19.7	35.8	6050	YES
FAMNet	52.0	48.7	19.1	**33.4**	3072	NO
eTC17	51.9	58.1	23.1	35.5	2288	NO
eHAF17	51.8	54.8	**23.4**	37.9	1834	NO

* Anonymous submissions are excluded as their method details are not accessible

Table R2: Top 5 methods on KITTI-Car*

	MOTA	MOTP	MT	ML	IDS	MCF
MOTBeyondPixels†	**84.2**	**85.7**	**73.2**	**2.8**	468	YES
AB3DMOT	83.8	85.2	66.9	11.4	**9**	YES
aUToTrack	82.3	80.5	72.6	3.5	1025	NO
JCSTD	80.6	81.8	56.8	7.4	61	NO
3D-CNN/PMBM	80.4	81.3	62.8	6.2	121	NO

† This graph design method was used in our submitted paper

Second, **investing in speeding up MCF solver is valuable to the whole field**, based on the following facts. (1) Computational efficiency has been identified as the bottleneck for MCF-based methods, as pointed out by references [14, 16, R3]. Indeed, [R3] turned to non-MCF based approach because the current MCF solvers could not meet their time requirement. Researchers in [14,16] developed faster sub-optimal approaches by sacrificing some tracking accuracy, as they were not satisfied with the global but slower MCF solvers. Our work guarantees global optimality and offers hundreds to thousands times of improvement over existing solvers. Thus, we believe this work would significantly improve the efficiency of the existing MCF based methods and enable many otherwise infeasible applications. Reference 14 and 16 are listed in our submitted paper. (2) As pointed out by Reviewer 2, the proposed strategies in muSSP are not limited to MOT problem. Except "dummy edge clipping", all the other strategies are applicable to generic unit-capacity min-cost flow problem. muSSP may also serve as an inspiration to develop specialized but more efficient network flow based algorithms.

Specific responses to Reviewer 1:

(a) The tracking results obtained by our proposed approach and baseline approaches are all the same, since MCF formulation admits a unique global optimal solution and we replace old MCF solvers by our more efficient one. For the quadratic problem (Table 2 in the paper), it can be proved that with the same initialization and step function in Frank-Wolfe method, the final solution is the same.

(b) The theoretical worst case of muSSP happens when the shortest-path tree contains only one branch in each iteration of Alg.1. This worst case is unlikely to happen in the graph of MOT problem, as the number of branches is at least equal to the number of objects in the first (or last) frame of the video. We will include these clarifications in the paper.

Specific responses to Reviewer 2:

(a) The MCF framework is able to handle missing detections as it allows linkages between detections beyond adjacent frames (e.g. from i-1 frame to i+1 frame).

(b) With the same set of detections, the nodes in the graph are fixed, but the arcs are not. Actually, the flexibility of setting different arcs and assigning different weights leads to various algorithms for MOT problems. For example, in [16], the weights of arcs were designed based on detection confidence, while [14, 20] used extra features such as detections' appearance similarity. We will include these clarifications in the paper.

Specific responses to Reviewer 3:

All comparisons were conducted on Ubuntu 16.04 LTS, compiled by g++ v5.4.0 with single core of 2.40GHz Xeon(R) CPU E5-2630 and memory speed at 2133MHz. This will be added in the revision.

References

[R1] The MOT17 Tracking Challenge. `https://motchallenge.net/results/MOT17/`
[R2] The KITTI Tracking Challenge. `http://www.cvlibs.net/datasets/kitti/eval_tracking.php`
[R3] Yoon, Young-Chul, et al. arXiv preprint arXiv:1907.00831 (2019).