A APPENDIX

A.1 Conversion from Travel Time to Flow

The conversion from the travel time $t(e)$ to the flow $f(e)$ of a road segment $e \in \mathcal{E}$ is attained by inverting the road-segment *performance function* proposed by the U.S. Bureau of Public Roads:

$$t(e) = t_{min}(e)\left(1 + 1.5\left(\frac{f(e)}{c(e)}^4\right)\right), \tag{24}$$

where $c(e)$ is the capacity computed as follows (the formula is accessible at http://www.fhwa.dot.gov):

$$c(e) = \begin{cases} 1700 + 10t_{min}(e) & \text{if } t_{min}(e) \leq 70mph, \\ 2400 & \text{otherwise.} \end{cases} \tag{25}$$

A.2 Evaluation of The Initial Traffic Reconstruction

We have generated abundant synthetic data for evaluating our approach. These synthetic datasets are produced via the traffic models that have been extensively validated using real-world datasets in transportation engineering. Many newly proposed traffic models are evaluated using these models and real-world datasets. We have conducted our experiments in a similar vein. In this section, we provide details on the generation of the synthetic dataset.

A.2.1 Road Network and GPS Dataset. The road network[6] used in testing is from downtown San Francisco (Figure 9), which contains 5,407 nodes, 1,612 road segments, and 296 TAZs[7]. The GPS dataset is obtained from the Cabspotting project [Piorkowski et al. 2009] (Figure 9 LEFT), in which the *low-sampling-rate* is reflected as the average difference between consecutive timestamps is approximately 60 seconds.

A.2.2 Traffic Conditions via System Optimal Model. We establish the first set of heuristic network travel times by solving the system optimal (SO) model. The SO model addresses the *traffic assignment* problem by minimizing the entire travel time of a road network, and takes the following form:

$$\begin{aligned} \text{minimize} \quad & z(\mathbf{f}) = \sum_{e \in \mathcal{E}} f(e)t(e), \\ \text{subject to} \quad & u_{rs} = \sum_{k \in \mathcal{K}_{rs}} u_{rs}(k),\ \forall (r,s) \in \mathcal{O} \times \mathcal{D}, \\ & f(e) = \sum_{(r,s) \in \mathcal{O} \times \mathcal{D}} \sum_{k \in \mathcal{K}_{rs}} \delta_e(k) u_{rs}(k),\ \forall e \in \mathcal{E}, \\ & u_{rs} \geq 0,\ \forall (r,s) \in \mathcal{O} \times \mathcal{D}. \end{aligned} \tag{26}$$

The solution to Equation 26 is a set of flows and travel times of all road segments in a network. The key input is the OD pairs which we estimate by first setting $\mathcal{O} = \mathcal{V}$ and then count in-and-out GPS traces for each TAZ. As GPS traces only represent a partial network flow, we multiply the estimated OD pairs by 10 constants and solve Equation 26 accordingly. As a result, we have constructed 10 network travel times in which the corresponding congestion levels[8] range uniformly from 0.19 to 1.85.

A.2.3 Traffic Conditions via Timestamp Model. Heuristic network travel times can also be generated based on GPS timestamps. Using the Cabspotting dataset, we equally distribute the time difference of two consecutive GPS points to all paths that connect them. For road segments that are covered by multiple GPS traces, the average travel times are adopted. Using this approach, we have produced 24 network travel times for each hour in a typical weekday. An example can be seen at Figure 9 RIGHT. We refer to this method of generating network travel times as the *Timestamp* model.

A.2.4 Synthetic GPS Traces. Using established network travel times, we can generate synthetic GPS traces in which the true traversed paths and other information are known. In order to study the effect of the number of traces on the estimation accuracy, we have randomly simulated 20 batches of synthetic traces from 50 to 1000 in increments of 50. Each batch contains 30 sets of GPS traces and all set contain the same number of traces (e.g. 50). As a result, we have generated 315,000 traces for each traffic condition and over 10 million traces in total. A synthetic trace is created by selecting a random source and a target in the network and planning the route using the shortest travel time. To mimic features of the real-world GPS dataset, the sampling rate is set to be 60 seconds, and all coordinates are perturbed by the Gaussian noise $(0, 20)$ in meters [Yuan et al. 2010].

A.2.5 Evaluation and Comparison. We compare our technique with two state-of-the-art methods, namely Hunter et al. [2014] and Rahmani et al. [2015]. The first method is equivalent to the inner loop of our *travel-time estimation* process. The number of EM iterations is set to 5 and the number of random allocations per aggregate measurement is set to 100. These settings are reported to produce the highest estimation accuracy in [Hunter 2014]. The second method takes a non-parametric perspective, using a kernel-based technique to estimate travel times. The weights used to allocate travel times to individual road segments are set to be the ratio of free-flow travel times among road segments [Hellinga et al. 2008].

We set parameters of our nested iterative process as follows: retaining the same settings for the inner loop as in [Hunter 2014], we empirically set the number of iterations for the outer loop to 10. This setting is based on results shown in Figure 10, where the relationship between the normalized convergence rate (%) and the number of iterations for both types of network travel times is plotted. Each datum in the plot is the average value computed using all network travel times across all sets of synthetic GPS traces of either the SO model (6,000 trials) or the Timestamp model (14,400 trials). The measurement of each trial is the mean square error (MSE) between a recovered and a ground-truth traffic condition (i.e. $\frac{\sum_e (t_e - \hat{t}_e)^2}{|\mathcal{E}|}$). As a result, the convergence rate decreases quadratically as the number of iterations increases and tends to flatten after 10 iterations.

[6] The road network is obtained from http://openstreetmap.org/.

[7] The TAZ shape file is obtained from https://data.sfgov.org/.

[8] The congestion level is measured by VOCs (volume over capacity) computed as $\sum_{e \in \mathcal{E}} \frac{f(e)}{c(e)}$.

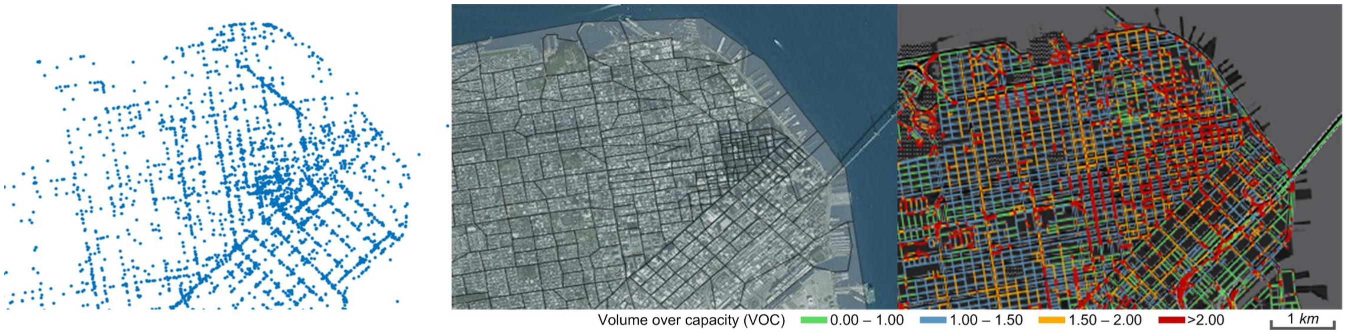

Fig. 9. LEFT: Sample GPS points from the Cabspotting dataset. MIDDLE: Road maps of downtown San Francisco overlaid with traffic analysis zones (TAZs). RIGHT: A heuristic traffic condition established via the Timestamp model (the travel times are converted to flows).

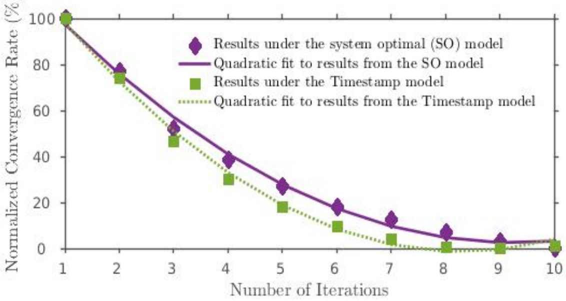

Fig. 10. The relationship between the normalized convergence rate (%) and the number of iterations of the outer loop of our iterative process is shown. The convergence rate decreases quadratically as the number of iterations increases and tends to flat after 10 iterations.

We evaluate our technique using three metrics. The first metric is the error rate of the aggregate travel time across the entire network, computed as $\frac{|\sum_e \hat{t}_e - \sum_e t_e|}{\sum_e t_e}, \forall e \in \mathcal{E}$, where $\hat{t}_e$ represents an estimated travel time and t_e represents a ground-truth travel time. The top diagram of Figure 11 LEFT shows the results by averaging the experimental outcomes of all network travel times via the SO model. The minimum error rate of our technique is 18%, of Rahmani et al. [2015] is 34%, and of Hunter et al. [2014] is 48%. Experimenting on network travel times generated via the Timestamp model, the corresponding minimum error rates are 8%, 28%, and 37%, which are shown in the bottom diagram of Figure 11 LEFT. As the number of synthetic GPS traces used in estimation increases, our technique demonstrates consistent advantages in performance over the other two methods.

The second metric is the relative improvement of our technique over existing methods on travel times of all road segments. We compute this metric based on $MSE = \frac{\sum_e (t_e - \hat{t}_e)^2}{|\mathcal{E}|}$ as:

$$RelativeImprovement = \frac{MSE_{existing} - MSE_{our}}{MSE_{our}}, \qquad (27)$$

where MSE_{our} represents the error between a recovered traffic condition using our technique and the ground-truth traffic condition, and $MSE_{existing}$ represents the error computed using an existing method with the same ground truth. The maximum relative improvements over Hunter et al. [2014] and Rahmani et al. [2015] under the SO model (shown in the top diagram of Figure 11 MIDDLE) are 78% and 97%, and under the Timestamp model (shown in the bottom diagram of Figure 11 MIDDLE) are 54% and 49%, respectively. In general, with more synthetic GPS traces used in estimation, better relative improvements are achieved. Such effects are more apparent on the SO model than the Timestamp model.

The third metric evaluates the map-matching accuracy. For one trace, we calculate the success rate as:

$$SR = \frac{\#\text{successfully identified road segments}}{\#\text{acutual road segments in the trace}}. \qquad (28)$$

We sum all success rates generated using our method and an existing approach, and derive the relative improvement as:

$$\frac{\sum SR_{our} - \sum SR_{existing}}{\sum SR_{our}}. \qquad (29)$$

The maximum relative improvements of our method over Hunter et al. [2014] and Rahmani et al. [2015] under the SO model are 28% and 34%, and under the Timestamp model are 19% and 25%, correspondingly. These results are shown in Figure 11 RIGHT. Again, as the number of GPS traces used in recovering network travel times increases, gains in the improvements are observed.

A.2.6 Analysis of Bilevel Optimization. We have shown that our approach outperforms existing methods on estimating travel times of a road network. In turn, by inverting Equation 24, we can obtain better estimations of target flows, which serve as inputs to the bilevel optimization program.

The factors affecting the program are the weighting factor η, the noisy level of target OD pairs $\bar{\mathbf{u}}$, and the noisy level of target road-segment flows $\bar{\mathbf{f}}$. The noises of $\bar{\mathbf{u}}$ and $\bar{\mathbf{f}}$ are assumed to have zero mean and diagonal variance-covariance matrices [Cascetta and Nguyen 1988]. In reality, the noisy level of $\bar{\mathbf{u}}$ is difficult to assess because not only $\bar{\mathbf{u}}$ usually comes from existing data but also the true values of $\bar{\mathbf{u}}$ are seldom. Due to these reasons, in the analysis of η and the noisy level of $\bar{\mathbf{f}}$, we set the normalized noisy level of $\bar{\mathbf{u}}$ to 50%. Subsequently, the normalized noisy levels of $\bar{\mathbf{f}}$ (%) computed based on MSE of estimated travel times to ground truth are shown in Figure 12 LEFT and MIDDLE. In general, our technique produces

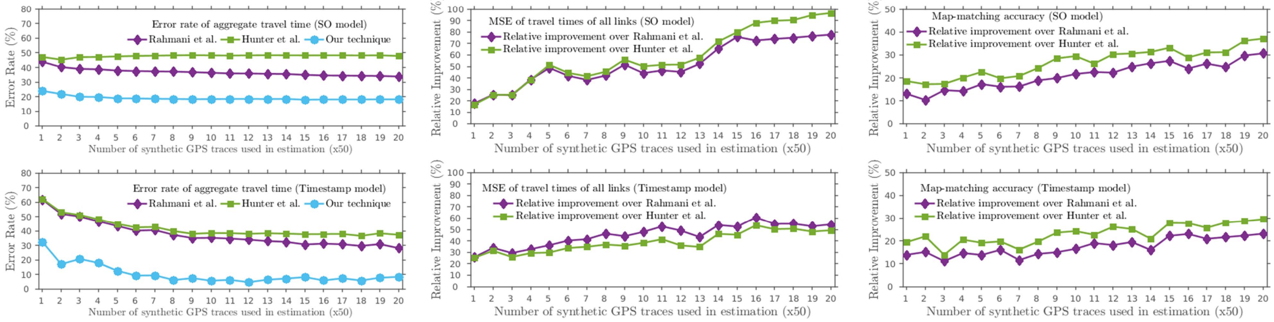

Fig. 11. This figure is copied from the main text for convenience. From LEFT to RIGHT, the top diagrams show results generated using network travel times via the system optimal (SO) model, while the bottom diagrams show corresponding results of network travel times via the Timestamp model. LEFT: The error rates (%) of various methods of aggregating travel time across the network. MIDDLE: The relative improvements (%) of travel times of all road segments measured in MSE. RIGHT: The relative improvements (%) of map-matching accuracy measured using successfully identification rates of road segments. In summary, our technique achieves consistent improvements over other methods as the number of GPS traces used in recovering network travel times increases.

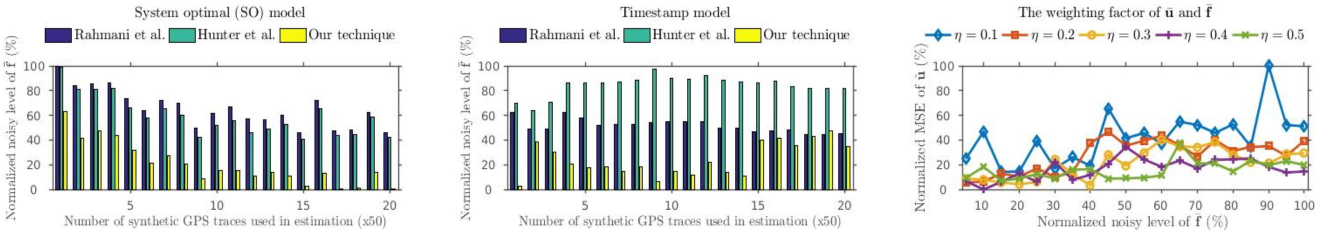

Fig. 12. LEFT and MIDDLE: The normalized noisy levels (%) of target road-segment flows $\bar{\mathbf{f}}$ computed according to MSE of estimated network travel times to ground truth. In general, for both models, our technique produces lower noisy levels than other two techniques. RIGHT: The normalized MSE of target OD pairs $\hat{\mathbf{u}}$ (%) under different values of the weighting factor η and various noisy levels of $\bar{\mathbf{f}}$ (%). When η is small, the error is more sensitive to perturbations on $\bar{\mathbf{f}}$. Overall, the error increases as the noisy level of $\bar{\mathbf{f}}$ increases. For all studies, the normalized noisy level of target OD pairs $\bar{\mathbf{u}}$ has been set to 50%. Our method has achieved consistently lower error rates compared to other methods.

lower noisy levels of $\bar{\mathbf{f}}$ than other two techniques, especially under the SO model which is considered to be a better approximation to real-world traffic than the Timestamp model [Sheffi 1985].

In order to evaluate how η and the noisy level of $\bar{\mathbf{f}}$ affect the estimation accuracy of $\hat{\mathbf{u}}$ (i.e. the estimated OD pairs), we compute the normalized MSE of $\hat{\mathbf{u}}$ (%) under different η and various noisy levels of $\bar{\mathbf{f}}$ (%). The results are shown in Figure 12 RIGHT. When η takes a small value (e.g. 0.1), the impact of $\bar{\mathbf{u}}$ is restricted, thus the MSE of $\hat{\mathbf{u}}$ reacts actively to perturbations on $\bar{\mathbf{f}}$. As we gradually increase the value of η, the impact of $\bar{\mathbf{f}}$ attenuates. Nevertheless, the MSE of $\hat{\mathbf{u}}$ increases as the noisy level of $\bar{\mathbf{f}}$ progresses.

A.3 Filter Design

Traffic exhibits periodic patterns. This phenomenon can be seen from the example loop-detector data in Figure 6 in the main text. Referring to the weekly data from a loop detector as a loop-detector signal, we have observed that such a signal can be approximated by a small set of Fourier coefficients while the energy and periodic structure are largely retained. This observation inspired us to develop a frequency domain filter that consists of most dominant frequency components.

One way of constructing such a filter is to transform all loop-detector signals to the frequency domain and take the average of all frequency components. While this approach can expect to reduce the embedded white noise, it doesn't consider other types of noise in the data. This is illustrated in the top panel of Figure 13**(c)**: the most significant frequency component is captured while other harmonics are reduced to various degrees.

Considering time-domain only methods, a naïve approach is to get the average signal, transform it to the frequency domain, and analyze its frequency components. Though this is a straightforward way in dealing with univariate and multivariate data, the resulting signal could be a poor summary of original signals both in the time domain (Figure 13**(a)**) and in the frequency domain (Figure 13**(c)** middle panel). The reason is that this approach does not take the phase variability into account. Due to these reasons, we align signals in the time domain, calculate the average signal, transform it to the frequency domain, and extract its major frequencies (see Figure 13**(b)** and Figure 13**(c)** bottom panel). As a result, we obtain both important frequency components and their corresponding magnitudes in the right ratio. The frequency-domain version of this signal then serves as our filter.

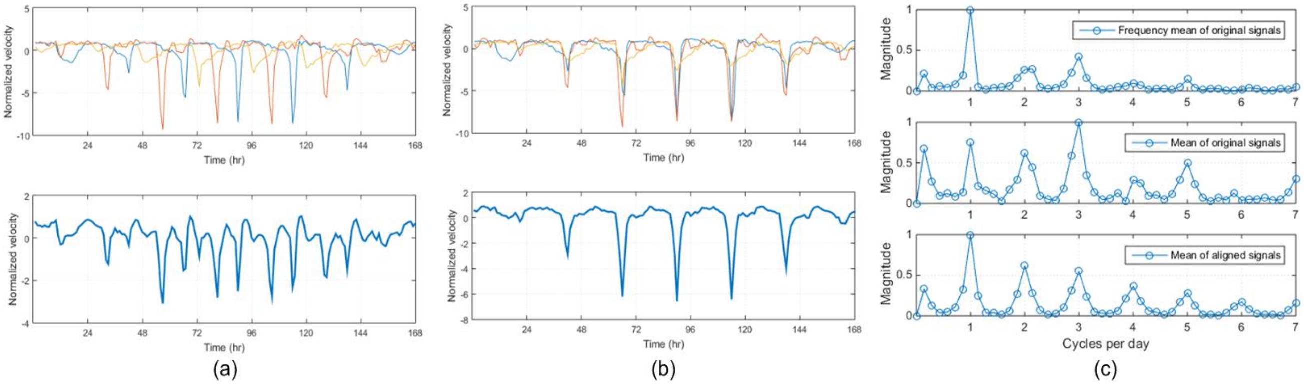

Fig. 13. (a) Several loop-detector signals are plotted showing phase shifts among them (top panel); the average signal of phase varied loop-detector signals (bottom panel). (b) Aligned loop-detector signals according to their phase responses (top panel); the average signal of aligned loop-detector signals (bottom panel). (c) Averaged frequency of phase varied loop-detector signals showing several frequencies are getting degraded (top panel); frequency of the averaged but phase varied loop-detector signals showing incorrect magnitude ratios (center panel); frequency of the averaged aligned signals with prominent frequencies and corresponding magnitude ratios (bottom panel).