Computing Arterial Travel Time Distributions From Loop Detector and Probe Datasets

Rahul Sengupta, Rohith R. K. Reddy, Parth Shah, James Dika, Xiaohui Huang, Anand Rangarajan, *Member, IEEE*, and Sanjay Ranka, *Fellow, IEEE*

***Abstract*— Estimating Arterial Travel Time distributions from high-resolution loop detector data (signal events and vehicle detection events) is a challenging task. Even though high-resolution loop detector data for several years may be available, the lack of other data modes and ground-truth labels, hinders approaches that rely on additional information. Among the approaches relying exclusively on loop detector data, are deterministic physics-based approaches (which use the mechanics of a virtual probe moving down a signalized arterial) and cost-minimization approaches (which estimate travel times from detector counts directly). In this work, we propose and evaluate a hybrid model that uses virtual probe trajectories to indicate probable arrival and departure windows, within which we apply a sequence alignment algorithm on high-resolution loop detector data to match platoons. We generate a broad range of traffic conditions and signal timing plans using SUMO traffic simulator and evaluate our approaches. We also verify our approach using real-world data collected along a 5-intersection signalized corridor. We show that virtual probe trajectories can be replaced by data collected from real probes (when available), thereby improving accuracy. Results show that our approaches enable us to calculate a good estimate of the arterial travel time distribution and are robust to noise. Thus, our methods can be used both with standalone archived loop detector data and in conjunction with data from connected vehicles.**

***Index Terms*— Connected vehicles, traffic control, detector, data-driven modeling, trajectory, estimation.**

I. Introduction

MITIGATING traffic congestion and improving safety are the important cornerstones of transportation for smart cities. With growing urbanization around the world, traffic congestion along high-volume signalized traffic corridors (arterials) is a major concern [1]. Congestion negatively affects productivity, leading to loss of work-hours, thus impacting the economy. Congestion also impacts the well-being of the society and the environment [2], [3]. One of the important measures of congestion is Arterial Travel Time [4]. This value is the expected travel time that a vehicle will take to complete its journey along a signalized traffic corridor and is affected by many factors including traffic conditions, departure time etc. This quantity is easy to interpret, by traffic engineers, city authorities as well as the general public. Current performance evaluations include a limited comparison of before and after travel-time data to demonstrate the effectiveness of signal retiming efforts [5]. However, traffic patterns vary dynamically during a day as well as globally within the network and there is a need for continuous monitoring and evaluation of signal timing parameters, based on performance and demand fluctuations. To achieve this, travel times have to be calculated at regular intervals. In addition, it is important to understand the distribution of travel times rather than just average travel times, as it gives traffic engineers richer information about the tail of the distribution. Actual travel time often has a multi-modal distribution and the expected (mean) values are not always sufficient.

While the availability of high-resolution (10 Hz) loop detector logs opens a broader range of possibilities, a fundamental problem of re-identifying the same vehicle at entry and exit points in the arterial network, still remains. To overcome this, the literature has introduced "Virtual Probes" [6], a kinematic model wherein vehicle locations and velocities are estimated using simplified physics and signal phase information. The first question asked and answered (to some extent) in this paper is the extent to which virtual probes with sequence matching, are useful in estimating travel times of vehicles. The approach we take is to *infer* vehicular movement given coarse virtual probe information. This is achieved by pairing virtual probe data with a bioinformatics-based sequence matching algorithm (henceforth referred to as Platoon Matching Algorithm). We denote this style of approach as *unsupervised* travel time estimation, since the only loop detectors and signal state information is used.

We next move to *semi-supervised* travel time estimation. Here, a small fraction of actual "labelled" ground-truth vehicular movements are fed to the Platoon Matching Algorithm, instead of virtual probe trajectories. Since no kinematic model is assumed here, it is the job of the sequence alignment-based inference engine to propagate the labeled vehicular matches to the rest of the vehicles, in the quest of obtaining an improved travel time estimation.

Our work has these main contributions:

1) We develop a hybrid method that combines a deterministic physics-based virtual probe model with a cost-minimization sequence alignment algorithm. This allows

Manuscript received 30 January 2021; revised 10 November 2021, 11 August 2022, and 23 April 2023; accepted 8 June 2023. Date of publication 4 August 2023; date of current version 1 November 2023. This work was supported in part by the National Science Foundation Computer and Network Systems (NSF CNS) under Grant 1922782, in part by the Florida Department of Transportation (FDOT), and in part by FDOT District 5. The Associate Editor for this article was X. Ban. *(Corresponding author: Rahul Sengupta.)*

The authors are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: rahulseng@ufl.edu; rkessireddy@ufl.edu; parth.shah1@ufl.edu; jamesdika@ufl.edu; xiaohuihuang@ufl.edu; anand@ufl.edu; sranka@ufl.edu).

Digital Object Identifier 10.1109/TITS.2023.3298345

it to compute distributions of travel times rather than only expected values.

2) We focus on using primarily loop detector and signal state data, as these are usually available to public traffic authorities. Our methods do not rely on vehicle re-identification methods like BlueTooth, GPS, Video tracking etc., though they can be used to boost accuracy of our method.
3) We show that our method is robust to a variety of traffic scenarios (variable amount of congestion, variable signal timing patterns, variable amounts of traffic entering and exiting the corridor). This is shown in simulation where ground-truth is available for comparison. Further, we also verify our approach using real-world data collected along a signalized urban corridor.
4) We show that the accuracy of our method can be improved when trajectory information (or generally, re-identification information) for a small number (1%-4% percent) of vehicles is available. We expect this data to be collected in real-time by the state agencies in the near future using road-side units and on-board units (installed in transit buses or other vehicles).

Given the above, the stage is set for a clear-cut comparison of real and virtual probes, corresponding to the semi-supervised and unsupervised situations respectively, coupled with platoon matching. In the rest of the paper, we flesh out these approaches and empirically compare the travel time distribution estimation by simulating low, medium and high traffic volumes. We also describe the verification of our approaches using real-world data. Previous work on this topic is first reviewed in Section II, followed by a description of the methodology in Section III, and simulation and real-world evaluation results in Section IV. We conclude in Section V and speculate on possible future trajectories of this work.

II. Background and Related Work

We briefly review other related work in estimating travel times, with a focus on using loop detector data.

An analytical model based on kinematic wave theory [7] is developed in [8], based on loop detector readings and signal plans. Extensions of this work include [9], [10] which account for long queues and spillovers. Similarly, [6] and [11] introduce the Virtual Probe method and its extension. Reference [12] uses cumulative plots of vehicles crossing loop detectors with sparse probe trajectories, to estimate travel time statistics. [13] estimates average link travel time of signalized arterials with loop detectors, using a platoon dispersion model to project downstream arrivals. Reference [14] uses Particle Filter-based approach to reconstruct vehicle trajectories for signalized arterials using sparse probe data, and estimate travel times. Reference [15] uses K-nearest neighbor and Least Squares Support Vector Regression model. Neural Networks [16] too have been used for imputing urban arterial travel time: [17] uses loop detector data with State-Space Neural Networks and Kalman Filters; [18] presents a hybrid model coupling the deep learning model and quantile regression using data from microwave sensors and loop detectors. A probabilistic modeling framework [19] is developed, estimating arterial travel time distributions from sparsely observed probe vehicles (a fleet of 500 taxis). A platoon-matching algorithm using vehicle flow densities over loop detectors, to first detect platoons, and then using exponential smoothing to find average arterial travel time is mentioned in [20]. Reference [21] fuses loop detector and probe data with Recursive Least Square filter coupled with Maximum A Posteriori interpolation, to impute truck travel times. Reference [22] details a comparison study using the fusion of bus-based GPS, loop detectors and mobile phone data for urban travel time estimation. Reference [23] uses vehicle color from video data coupled with loop detector data to estimate travel times of platoons. Reference [24] introduces a framework for vehicle re-identification via signature matching using signal processing techniques and a travel time estimation algorithm, based on microloop detectors. Reference [25] uses a kinematic wave model to estimate travel time distributions for arterials in undersaturated conditions with known fixed-time traffic signals, with respect to departure and traffic flow information. An approach fusing the theory of traffic flow through signalized intersections with machine learning (Expectation Maximization algorithm) to estimate travel times using GPS data is presented in [26].

In our work, we have extended the key ideas in the physics-based Virtual Probe Model described in [6], paired with the Needleman-Wunsch Sequence Alignment algorithm [27]. The Needleman-Wunsch algorithm was originally developed to align amino acid sequences. Recently, a similar approach was used to analyze dedicated short-range communication (DSRC) data [28], and BlueTooth data [29] to identify road user classes.

III. Integration of Platoon Matching With Virtual and Real Probes

In this section, we describe our novel hybrid approach to estimating arterial travel time distributions. Our objective is to analyze loop detector data for travel times, with and without actual trajectory data (from GPS, BlueTooth or Video etc.). Loop detector data is usually readily available in large quantities to public agencies [30]. As (sparse) trajectory data also becomes available, we show how to incorporate it to further improve the algorithm and quantify the impact.

A. Intuition Behind the Method

Let us consider a platoon (i.e. a cohort) of vehicles that exit the first intersection together. Due to platoon dispersion and other effects, these vehicles may not cross the second intersection's stop-bar together as a cohort. A portion of the original cohort may go through, leaving the remaining to exit in the next upcoming green phase at the second intersection. This pattern of platoon break-up and the collapse of the "Green Wave" is likely to repeat in subsequent intersections of the corridor. This leads to the original cohort of vehicles, exiting the last intersection at different times.

It is important to note that the vehicles on the corridor are being captured by two distinct sensor modes:

- Inductive Loop Detector Data at Intersection Stop-Bars: These fixed sensors count the number of vehicles that pass

over them. Such sensors are capable of capturing the bulk of the traffic that flows across the corridor. However, it is not possible to re-identify a vehicle across the various loop detectors along the length of the corridor. Thus, this mode provides *bulk non-re-identifiable vehicle data*. Consequently, it is not possible to accurately estimate an individual vehicle's travel time with this mode.

- Probe Vehicle Trajectory Data (such as from GPS): This mode of data can re-identify the same vehicle as it crosses the co-ordinates of the stop-bars of successive intersections. However, such data is usually sparsely available with penetration rates of a few percent of the overall traffic flow. Thus, this mode provides *sparse re-identifiable vehicle data*. This mode allows us to accurately determine a vehicle's corridor travel time but it is too sparse to get the complete travel time distribution. While GPS data can be used, even intersection-mounted video data (with manual visual tracking or computer vision-based tracking [31]) can be used to ascertain the timestamps when a sample of vehicles crosses the intersection stop-bars.

In order to estimate the distribution of travel time of the bulk of the corridor traffic, we want to use the *sparse re-identifiable vehicle data* to enrich the *bulk non-re-identifiable vehicle data*.

We use the following insights:

- Corridor-traveling vehicles that are close to probe vehicle at the start, are likely to exit the corridor either in the same cycle as the probe vehicle, or a couple of cycles before/after. In the real-world settings, care is generally taken to ensure a "Green Wave".
- The relative order of these vehicles near the probe is likely to remain the same for the most part. It is highly unlikely that most vehicles behind the probe would pass it and move ahead, or the other way around.
- All these vehicles (probe as well as non-probe) will be captured by the loop detectors, though we do not know which vehicle is mapped to which loop detector actuation.
- These vehicles (by and large) will only cross the intersection stop-bar during a green (or yellow) phase, and not break the red light. While crossing, the vehicle may first have stopped at the intersection (due to a red light and/or a queue) or the vehicle may have passed through without stopping.
- For the probe vehicles, we know the timestamps when they cleared the stop-bars of the first and last intersections.

Using the above insights, we can first find out which time windows of green phases of the signal cycles (i.e. Green Phase Time Windows) the probe vehicles crossed the first and last intersections. Specifically, when using ATSPM Data [32], we can look up the ATSPM codes that indicate the start of "Phase Begin Green" and "Phase End Yellow Clearance" for the phase that serves the corridor-through traffic at the first and last intersections. These give us the timestamps of the start and end of the green times. We just need to see which window (time interval) the probe's crossing timestamp falls in.

Secondly, we can use this time window to slice out the loop detector actuation data for the detectors serving the corridor-through phase. Specifically, when using ATSPM Data, we can look up the ATSPM codes for "Detector On". Often there are multiple detectors serving the phase; we can add them up to get a single time series that indicates the total number of vehicles in that time bucket.

As an illustrative example, suppose the green times (plus yellow) at an intersection occur from timestamps 0s-60s, 120s-200s, 240s-300s. From the probe's trajectory data, we know the probe exited an intersection at 130s. We can infer that the probe exited the intersection in the second Green Phase Time Window spanning 120s-200s. We can use this interval to slice out the relevant portion of the loop detector actuation data. Notice that the Green Phase Time Windows need not necessarily be of the same length, nor necessarily be periodic. Figure 1 shows this visually and describes it as well.

We can do the above process separately for the probe's crossing of the first and last intersections. We now get two time series sequences of detector actuations, one from the first and the other from the last intersection. It is these two that we will compare and align to calculate their probable travel times. Note that we already know the probe's own travel time. The slicing and alignment procedure is for the loop detector data. Thus, we have used the *sparse re-identifiable vehicle data* to extract the most useful portions of the *bulk non-re-identifiable vehicle data* for alignment.

It is possible that the *sparse re-identifiable vehicle data* (GPS/ Video tracking) is not available at all, and only the loop detector and signal state data are available. Using the Virtual Probe Model [6], trajectories of virtual probe vehicles can be computed. The pre-existing Virtual Probe Model and our novel Platoon Matching Algorithm (described in the next section), both use only the loop detector data, but in different ways. The former uses it to estimate the growth and discharge of queues at various intersections (and thus estimate the virtual probe's trajectory) while the latter uses it to match probable platoons as they enter and exit the corridor. While Virtual Probe Model has been used here due to its popularity in the literature and because it exclusively uses loop detector data, any algorithm/sensor modality that can yield reconstructed trajectories (e.g. [14]) for probes, can be used with our Platoon Matching Algorithm. We now describe our novel algorithm for Platoon Matching.

B. Novel Platoon Matching Algorithm Using Windowed Sequence Alignment

The Needleman-Wunsch (NW) algorithm [27] is a dynamic programming algorithm, widely used in the field of bioinformatics to align nucleotide sequences. The NW algorithm takes two sequences of alphabets which are to be aligned. It also takes a "Similarity Matrix", which is an array of scores associated with matching (or mis-matching) various component alphabets with each other. It can also introduce "gaps", and there is a penalty for opening a gap, and for extending them. Thus, a potential alignment has a total score associated with it which depends on:

- Sum of scores of matches and mismatches
- Sum of scores of starting and extending gaps

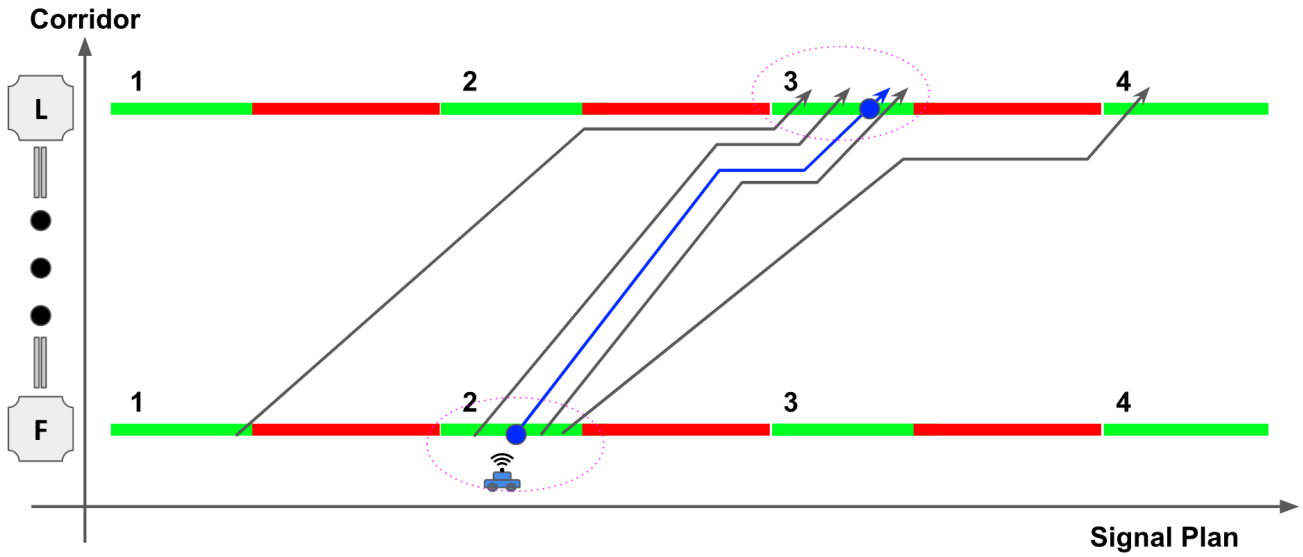

Fig. 1. Time-Space Diagram showing the selection of the Green Phase Time Windows. Thanks to the ATSPM data, we have the signal state data at the first (F) and last (L) intersections. In this above example, we can see our probe vehicle (blue car) exited F in cycle 2, and L in cycle 3. We know this by looking at the trajectory data of this probe, and noting the time duration when it cleared the intersection. This time would be during a Green Phase at the intersection, or just after the end of one. These Green Phase Time Windows (F-2 at the first and L-3 at the last intersections) tell us the portions of the loop detector actuations we must look at, and match them. The matching is done using Needleman-Wunsch Algorithm. Note that while both the windows (F-2 and L-3) contain 4 actuations each in their loop detector traces, the same 4 vehicles did not induce them. The last vehicle of F-2 passes in L-4, while the first actuation in L-3 is by vehicle that came from F-1. When the Platoon Matching occurs, the first actuation of F-2 will get matched to the first actuation of L-3. This is for just one probe vehicle and the windows F-2 and L-3 it identified. Hence, having more probe vehicles will identify more probable windows for the Platoon Matching Algorithm.

The NW algorithm scores such possible alignments and finds the one with the best score using dynamic programming. It is thus imperative that we (1) slice and (2) re-cast our detector actuation data in an appropriate format and (3) provide an appropriate "Similarity Matrix". The next three subsections discuss (1), (2) and (3).

1) Slicing and Identifying Green Phase Time Windows: As described in Section III-A for all probe vehicles, we identify the first and last Green Time Phase Windows, corresponding to the crossing times of the first and last intersections of the corridor. The vehicle is said to have crossed the stop-bar when it is 5 m or more after the stop-bar location. This way, we can be sure that the vehicle has decisively crossed the stop-bar.

We use these Green Time Phase Windows (timestamps) to slice out the relevant detector actuation time series. In the case of multiple detectors serving the corridor-through phase at an intersection, we aggregate by summing multiple detectors' time series to form a single time series.

Each of these two sequences is at 10 Hz resolution if ATSPM data is used. The lengths of the two sequences will be the length of the Green Phase Time Windows at the first and last intersections when the probe crossed them.

2) Re-Casting Detector Actuation Data: Before aligning, pre-processing is first performed on both strings:

- The loop detector actuation strings are re-sampled and bucketed at 1-second resolution with symbol "V" to indicate the time bucket when at least one vehicle crossed over it, or "S" to indicate a space between vehicles.
- Vehicles passing within 5 seconds of each other are assumed to be in the same platoon. Such "V"s in the string are replaced with "P". The remaining "V"s can be thought of as "noise" vehicles, that may not be a part of the platoon, as they could be entering or exiting the corridor, midway.

TABLE I
SCORE MATRIX FOR MATCHING DETECTOR STRINGS

	P	V	S
P	+50	+5	-5
V	+5	+20	-2
S	-5	-2	0

The main reason for performing the above transformation is that we would like the sequence matching algorithm to treat isolated vehicles and platoon vehicles differently.

3) Providing a Similarity Matrix: The score matrix for the same is shown in Table I. In addition to matching scores, we account for spaces, by allowing for the introduction of gaps i.e. extra "l". These are functionally same as spaces "S". However, to limit the number of such insertions to a minimum, we impose a gap start penalty of -1 and gap extension penalty of -0.5. This allows us to match more vehicles ("P"s and "V"s) in the sequences without greatly distorting the sequences.

C. Applying Needleman-Wunsch Algorithm

With these steps (1), (2), (3) done, the Needleman-Wunsch algorithm is applied to match the detector actuation sequences of the first and last intersections.

For example, let us say we wish to align two sequences, $VVVSSS$ and VSV. One such candidate alignment as returned by NW algorithm is shown in Equation (1). Note the gaps introduced in the second sequence to make them the

TABLE II
TOTAL SCORE CALCULATION

Correctly matching V with V	+20
Correctly matching S with S	0
Incorrectly matching V with S	NA
Incorrectly matching S with V	-2
Starting gap	-1
Extending gap	(-0.5) + (-0.5)
Total Score	+16

same length.

$$\begin{aligned} &\text{Sequence 1}: \ VVVSSS \\ &\text{Sequence 2}: \ V|||SV \end{aligned} \tag{1}$$

The total score for this alignment is calculated as shown in Table II. The NW algorithm searches through such candidate alignments and finds a matching with the optimal score; one that maximizes matching of "similar" alphabets and minimizes the matching of "dissimilar" alphabets. This example only had V and S; similarly sequences including P also, can be handled by the NW algorithm.

Once matched, we find the difference in the indices, and add the time difference between the two matching windows.

Let W_F be the start time of the Green Phase at the first intersection during which the probe crosses the first intersection of the corridor. Similarly, let W_L be the start time of the Green Phase at the last intersection during which the probe crosses the last intersection.

The Needleman-Wunsch algorithm returns a pair of aligned sequence arrays. Let A_F and A_L be the returned arrays. Let the first vehicle be matched in A_F at index j, and at k in A_L. The travel time T can be estimated using Equation (2)

$$T = (W_L - W_F) + (k - j) \tag{2}$$

We similarly find the travel times for all matched vehicles ("P"s and "V"s) in the aligned sequences. We repeat the process for all probes' trajectories.

D. Virtual Probe Model for Arterial Travel Time Estimation

We briefly describe the pre-existing Virtual Probe Model [6]. It is assumed that loop detector data, along with signal state data across the corridor is available. An imaginary probe vehicle traces a path across the corridor, governed by kinematics-based equations to calculate its trajectory from given acceleration and deceleration parameters. The maneuver decision tree of the virtual probe is shown in Figure 3. For detailed mathematical description, please refer to Section III of [6].

An important property of the Virtual Probe is its ability to self-correct to a limited extent. Whenever the probe stops (virtually) at a red light, the error in the distance with respect to an actual vehicle (which also started around the same time and place as the virtual probe) decreases. A vital component of the Virtual Probe Model is the queue length estimation algorithm. The method outlined in [6] is not effective for queues that spill beyond the advance loop detector, common during congested traffic scenarios. Hence, we use [11] that is more effective for congested scenarios. The vehicle actuation pattern at the stop-bar detector is analytically analyzed from archived loop detector and signal state data, and key breakpoints are identified. From this, the build-up and discharge of vehicles at the approach is estimated. By using an assumed average vehicle length, queue length as a function of time, is calculated. This is used to select the appropriate maneuver decision for the virtual probe, as it interacts with the queue.

E. Platoon of Virtual Probes

A natural way to use the Virtual Probe method to generate a travel time distribution is to simply run it multiple times with different arrival times.

While developing our approach, we found that as the number of intersections in the corridor grew, the estimate by just a single Virtual Probe grew worse, often leading to large errors in the order of the cycle length. This was due to the platoon breaking up, with the Virtual Probe either narrowly crossing (or missing) a green light, and thus getting significantly ahead (or left behind) with respect to the bulk of the actual vehicles that had started alongside the Virtual Probe. Even with short road links, (∼200-250m), not being a part of the "Green Wave" which crosses the entire corridor uninterrupted, can add to travel time delays in the order of the cycle length, as those left behind must wait at an intersection for the next cycle. This is pronounced due to ill-timed signal coordination and platoon dispersion due to driving behaviors of vehicles, especially those wishing to enter/exit the corridor.

Running the Virtual Probe algorithm multiple times alleviates this limitation somewhat. The start times of these multiple virtual probes are interspersed throughout the cycle, broadly based on observed arrival patterns. We assume that the end users of this travel time algorithm are not very particular about when exactly during the cycle the vehicle arrived at the intersection, but rather are interested in the general distribution of travel times for a particular traffic scenario as a whole.

In summary, the Virtual Platoon Model allows us to include the physics of vehicles by tracing the space-time trajectory of several virtual vehicles. Though the equations developed by Liu et al. [6] are relatively simple, they do broadly capture the macroscopic behaviors of the vehicles. But they can't effectively capture the behaviors such as lane-changing, overtaking etc. We thus create a platoon of virtual probe arriving at different points during the traffic signal cycle. We then use these estimated trajectories to match probable green time windows at the stop-bar of the first and last intersections of our corridor of interest. To do this, we pre-process the detector time series data and represent them in terms of sequences of vehicle actuations and interleaving gaps. These sequences are then matched based on an input score matrix. Figure 2 shows a visual overview of the efficacy of the Platoon Matching algorithm. The algorithm is broadly summarized in Figure 4. In the next subsections, we describe various combinations of the above methods, with and without sparse actual probe trajectory data.

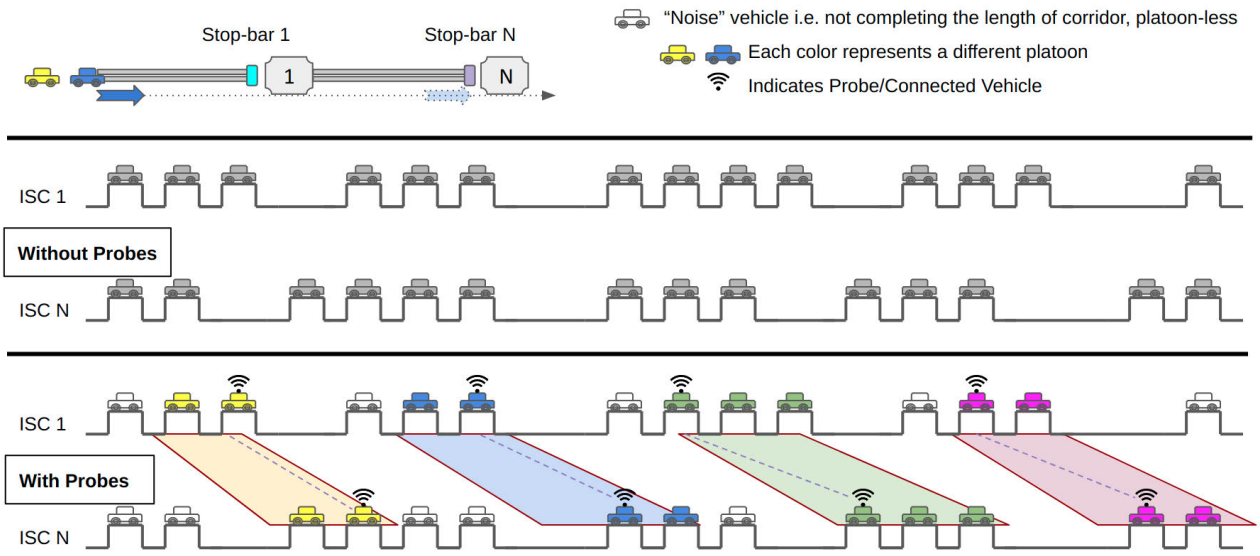

Fig. 2. Visual overview of the Platoon Matching algorithm. Stop-bar loop detectors along the length of a signalized corridor capture vehicle actuation waveforms. However, it is not possible to identify which vehicle actuation at the first intersection matches with its counterpart at the last intersection, when the vehicle exits the corridor. However, with the introduction of probe vehicle data (either Virtual Probe estimated trajectories or connected vehicle "Real Probe" trajectories), we can use the Needleman-Wunsch based Platoon Matching algorithm to match vehicles around the probe vehicle in its crossing green time windows at the first and last intersections. The key assumption is that most vehicles that were near the probe vehicle at the first intersection, are likely to stay that way when the probe crosses the final intersection. This allows us to match vehicle actuations at the first and last intersections, and compute the distribution of corridor travel times.

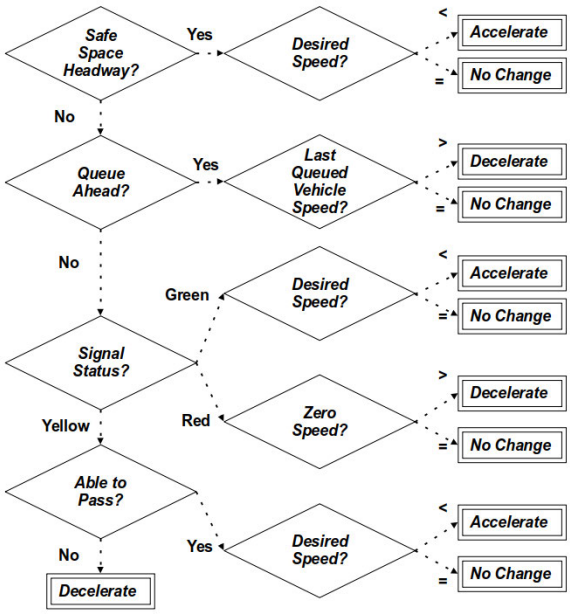

Fig. 3. Maneuver decision tree of the virtual probe. As much as possible, the probe attempts to move at the posted speed limit. However, it slows down when it sees a queue ahead or an (imminent) red light. Once the path is clear, it speeds up to attain the posted speed limit. At every second of its virtual journey, the probe must decide on a maneuver decision, whether to accelerate, decelerate or to keep the same speed.

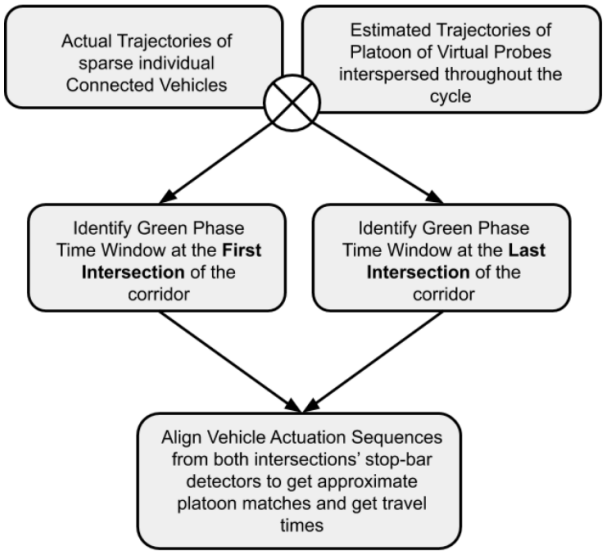

Fig. 4. Overall Algorithm Flowchart. The trajectories of sparse actual connected vehicles and estimated trajectories of virtual probes are input to the algorithm. Then, green phase windows corresponding to those trajectories are identified at the first and the last intersections of the corridor. With the appropriate stop-bar detector waveform slices identified, platoon matching is applied to get corridor travel times.

F. Inclusion of Real Probe Trajectory Data Using Additional Sensor Modes

While other works (discussed in Section II. Background and Related Work) focus on fusing loop detector data with probe trajectory data to estimate travel time distribution, our primary focus is to exclusively use only loop detector data as much as possible. Virtual Probe method and our Platoon Matching Algorithm do just that, without the need for additional sensor instrumentation.

However, with improvements in ITS, sparse data from other sensor modes may be available, in addition to high resolution loop detector data. Examples include:

- Floating Vehicle data from Global Positioning System (GPS)
- Vehicle identification data from BlueTooth detection systems

- Dedicated Short Range Communications (DSRC) data from on-board (OBU) and roadside (RSU) units
- Video data with vehicle tracking across the corridor

All such data sources/methods (such as [14]) that give a good estimate of crossing times of probes at the first and last intersections, can be used with our approach. Hence, we explore the impact of fusing this sparse additional data (if available) *on our Platoon Matching Algorithm* and quantify the impact. We do this to show that our method is extensible. Comparing our approach with other methods predicated on using multiple sensor data sources is beyond the scope of this paper.

Effectively, the virtual probes can be supplemented/replaced by real probes depending on availability of requisite data. While virtual probes are a rule-based vehicle trajectory reconstruction method which "guess" the likely trajectory of a phantom vehicle, these above-mentioned data sources can give actual observed trajectory data of real vehicles. Given a more accurate estimate of entry and exit times, estimates of travel times can be improved using our Platoon Matching algorithm. The availability of such data (like Video-based queue length estimation) may also help in better modeling of queue build-up and discharge, which improves the accuracy of the virtual probe model. Thus, the methods outlined are not only useful for analyzing historical high-resolution loop detector data but are also extensible with new data modes which are becoming prevalent.

G. Combination of Methods

For our experiments, we use combinations of the different methods described above. The two methods "Virtual Platoon (VP) with novel Platoon Matching (PM)" and "Real Probes (RP) with novel Platoon Matching (PM)", use our novel Platoon Matching method. The other two methods "Only Virtual Platoon (VP)" and "Virtual Platoon (VP) with Real Probes (RP)" are logical benchmarks against which we compare the efficacy of our Platoon Matching algorithm.

- Only Virtual Platoon (VP): We use Virtual Platoon method by itself, in order to establish a baseline. We use a virtual platoon of size 1 i.e. a single virtual probe, as well virtual platoons of sizes 1%, 5% and 10% of the number of vehicles passing through the corridor. Each probe returns an estimated travel time. We found that no incremental benefit was gained by using sizes beyond 10%.
- Virtual Platoon (VP) with novel Platoon Matching (PM) algorithm: We use Virtual Platoon method in conjunction with the Windowed Sequence Alignment for Platoon Matching. We use a virtual platoon of size 1 i.e. a single virtual probe, as well virtual platoons of sizes 1%, 5% and 10% as before. Each probe's trajectory is used to get sequences for the Platoon Matching algorithm, which then returns a collection of matched vehicle travel times. Thus, each probe now returns several estimated travel times based on matches by the Platoon Matching algorithm.
- Virtual Platoon (VP) with Real Probes (RP): In case sparse trajectory information from real probes is available, we use this along with estimated virtual platoon trajectories. As before we use a virtual platoon of size 1 i.e. a single virtual probe, as well virtual platoons of sizes 1%, 5% and 10%. We assume that real probe trajectory data is available only for 1%, 2% and 4% of the overall traffic volume across the corridor, which are realistic percentages in present times. Each probe (virtual or real) returns an estimated travel time.
- Real Probes (RP) with novel Platoon Matching (PM) algorithm: As with above, in case sparse trajectory information from real probes is available, we use this with the Platoon Matching Algorithm. Once again, we assume that real probe trajectory data is available only for 1%, 2% and 4% of the overall traffic volume across the corridor, Each real probe's trajectory is used to get sequences for the Platoon Matching algorithm, which then returns a collection of matched vehicle travel times. Thus, each real probe now returns several estimated travel times based on matches by the Platoon Matching algorithm.

IV. Experimental Results and Discussion

In this section, we describe our experiments and discuss the results obtained.

A. Experimental Setup

For the purposes of evaluating our algorithm, we run the Platoon Matching algorithm with trajectories from both virtual probes as well as real probes, in SUMO [33] microscopic traffic simulator. We compare the performance of these two methods against the ground-truth corridor travel times reported by the simulator.

We simulate an 8-intersection corridor as shown in Figure 5. The vehicles of interest start from the left to the right (Eastbound), and cross 8 signalized intersections. The following parameters are varied across each set of experiments:

- Deviation of Offsets in Signal Timing Plans: Offsets are set based on the estimated travel times between the intersections, to promote a "Green Wave" across the corridor. We use a simple heuristic of staggering the offsets of consecutive intersections based on the free-flow travel time between the two intersections. To simulate a variety of traffic scenarios, we perturb the set of offsets by a maximum deviation amount, with a higher deviation indicating ill-timed offsets. We test with offset deviations in the order of 10%, 20% and 30% of the cycle time i.e. 12 seconds, 24 seconds and 36 seconds.
- Amount of end-to-end traffic flow: These vehicle flows start at one end of the corridor and exit the corridor in the Eastbound direction, and form the bulk of the vehicle flows along the corridor. We simulate under varying degrees of traffic flows, with low (450 veh/hr), medium (750 veh/hr) and high (1050 veh/hr) flows. In addition to these, there are "noise" vehicles on the corridor, described below.

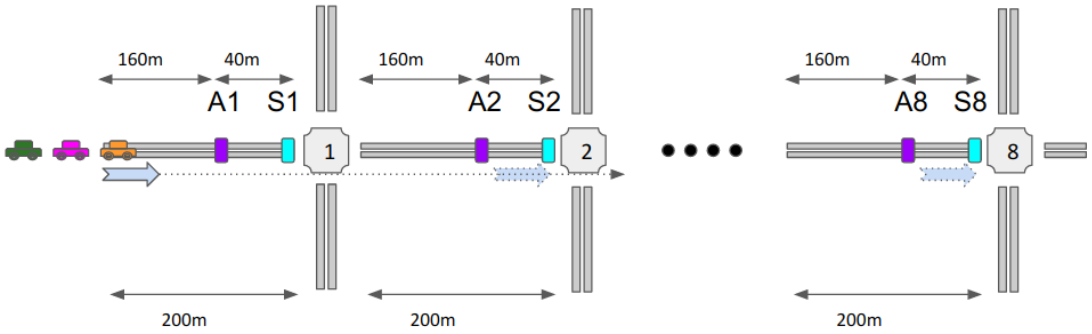

Fig. 5. 8-intersection corridor used for simulations. "A" and "S" refer to the advance and stop-bar detectors at the approaches of the intersections, as is typically seen in urban arterials. We use data from only the first and last (i.e. ISC 1 and ISC 8) stop-bar detectors for the Platoon Matching algorithm. However, the queue length estimation algorithm used by the Virtual Probe, does indeed use the intervening detector data. The vehicles of interest start from the left to the right (Eastbound), and cross 8 signalized intersections. Each approach is 200 meters long, with the distance between stopbar and advance detectors being 40 meters. We track the vehicles when they enter the approach of the first intersection (Intersection "1") and cross the stopbar of the last intersection (Intersection "8").

- Amount of "noisy" traffic: To test the robustness of our method, we add a percentage of vehicles which may either turn-in into the corridor at a random intersection or turn-out at a random intersection, at random times. But they do not complete the full journey across the corridor. We do not analyze these vehicles during our travel time calculations. Nevertheless, these vehicles do affect other vehicles on the corridor and are also captured by the various detectors. We test with 10% and 20% extra noisy traffic, above the end-to-end traffic flow. While we do not explicitly model missed detections and detector errors, these noise vehicle flows somewhat account for vehicle flow mismatches at first and last intersections.

Cycle length at each signal is set to 120 seconds, with the main corridor flow (Eastbound) having a Red time of 60 seconds, followed by a Green time of 55 seconds and a Yellow time of 5 seconds. While using common cycle lengths is a standard practice in urban corridors, our method does not rely on it being enforced. It only requires the Green Phase Time Windows (time slices) when the probe vehicles crossed the first and last intersections of analysis. These green windows need not occur in a cyclic manner, nor even be of the same length.

All vehicles follow standard behaviors with respect to vehicle safety, car-following, lane-changing etc. as implemented in SUMO simulator. Further details can be found in [34].

We run each simulation for 30 minutes of simulated time. We randomly sample 25 sets of offsets based on the allowed degree of deviation. For every set of signal offsets, we run the simulation twice with different random seeds i.e. 1 hour of simulated time per set of signal offsets. Thus, each set of experiments is run for 25 hours of simulated time.

The algorithms and the simulation control codes have been implemented in the Python[1] programming language. Specifically, we use NumPy[2] numerical computing library for implementing the Virtual Probe Model and BioPython[3] bioinformatics library for the Needleman-Wunsch Sequence Alignment algorithm.

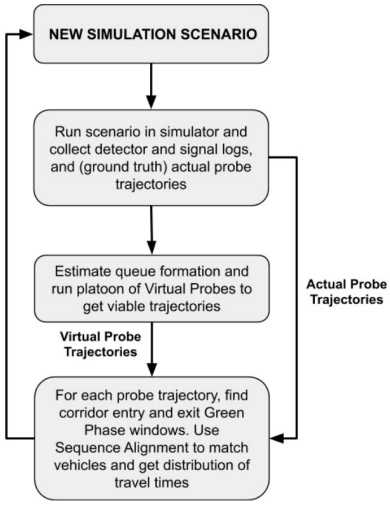

Fig. 6. Overall code flow for simulation experiments. The corridor scenario is loaded into the simulator and traffic flows are generated. Detector and signal logs are retrieved from the simulation engine, along with vehicle trajectories. With the recorded logs, a platoon of Virtual Probes is run to get viable trajectories. These are then used in conjunction with the sequence matching algorithm to match probable platoons, yielding travel times. The resulting distribution is compared against the ground-truth travel time distribution calculated using the stored vehicle trajectory information.

The overall code flow has been described in Figure 6.

For processing the results to obtain metrics, we use Pandas[4] data manipulation library, SciPy[5] and Dictances[6] computation libraries.

B. Evaluation Metrics

Figure 7 qualitatively shows a plot of the travel time distributions of only Virtual Probes (blue) and Virtual Probes with Platoon Matching (orange). We can clearly see that the orange distribution more closely matches the Ground-Truth travel time distribution (green) of all vehicles. This shows

[1] docs.python.org
[2] www.numpy.org
[3] www.biopython.org
[4] pandas.pydata.org/
[5] www.scipy.org/
[6] github.com/LucaCappelletti94/dictances

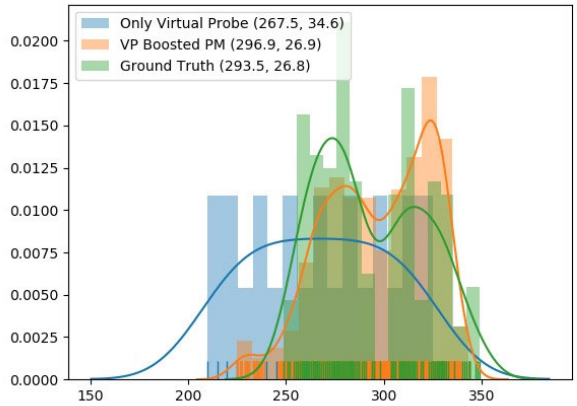

Fig. 7. Visualization of travel time distribution results from simulation experiments. Three travel time distributions from a simulation run are plotted, with their mean travel times and standard deviations mentioned in brackets. The green distribution is the ground-truth corridor travel time distribution of all vehicles. The blue distribution represents the corridor travel times estimated using only virtual probes. When augmented with the Needleman-Wunsch based Platoon-Matching algorithm, the orange distribution is obtained. As seen, the orange distribution more closely resembles the ground-truth and even shows double peaks, unlike the flatter blue distribution. This shows the boost in improvement that the Platoon-Matching algorithm provides. As real probe trajectories are added to virtual probes with platoon-matching, the orange distribution tends closer to the ground-truth green distribution.

the boost in improvement that the Needleman-Wunsch based Platoon-Matching algorithm provides.

However, we wish to quantitatively measure the improvement in performance. Thus, we use the following metrics for evaluating the performance of our algorithm under different traffic scenarios. Given n simulation runs, we calculate the following metrics:

1) Average of Mean Absolute Percentage Error : For every simulation run, we calculate the average predicted travel time and find its absolute error (as a percentage) with respect to the ground-truth travel time. We take an average of all such errors across all simulation runs of that category.

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\left|y_{pred,i}-y_{truth,i}\right|}{y_{truth,i}}$$

2) Average of Normalized Absolute Difference in Means (or Standard Error Distance) : For every simulation run, we calculate the average predicted travel time and find its absolute error with respect to the ground-truth travel time. We divide this by the square root of the sum of squares of their respective standard deviations.

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\left|\mu_{pred,i}-\mu_{truth,i}\right|}{\sqrt{\sigma^2_{pred,i}+\sigma^2_{truth,i}}}$$

We also see the difference in the distributions by computing histograms of the predicted and actual results for a simulation. We do so by taking the middle 80^{th} percentile range of the combined union of ranges of predicted and actual distributions. We bin this range into 10 bins. The distributions are individually normalized and compared using the Hellinger Divergence.

3) Hellinger Divergence (HLD) : The Hellinger Divergence [35] is a distance measure which calculates the overlap between two distributions. Let p and q be the PMFs (Probability Mass Functions) of the predicted and ground truth samples. It ranges from 0 to 1. Then the Hellinger divergence is given by

$$\mathrm{HLD}(p,q)=\sum_{k=1}^{K}\left(\sqrt{p_k}-\sqrt{q_k}\right)^2. \tag{3}$$

The primary advantage of the Hellinger divergence over the Normalized Absolute Difference in Means mentioned above (which only uses the mean and variance information of the samples) is that it takes the entire distribution into account. The primary disadvantage is that the PMF estimation (since it's based on picking the right number of bins in the histogram) may be incorrect leading to downstream errors in the Hellinger divergence. Other divergence measures (Kullback-Leibler [36] etc.) have the same problem (and so in the future we may turn to measures based on the sample Cumulative Density Functions).

C. Simulation Results and Discussion

The results obtained for low, medium and high traffic scenarios are tabulated in Table III. The results are also shown according to the performance metrics (i) Average of Mean Absolute Percentage Error (MAP) in Figure 8, Average of Normalized Absolute Difference in Means (STD) in Figure 9, and by Hellinger Divergence (HLD) in Figure 10.

We note the following trends in our evaluated results:

- With fewer Virtual Probes (Single\1% VP vs. Single\1% VP with PM), the addition of Platoon Matching Algorithm significantly boosts results. However, when the number of Virtual Probes is higher (5%\10% VP vs. 5%\10% VP with PM) the addition of Platoon Matching Algorithm doesn't help much. This is due to the fact that increasing the number of Virtual Probes gives as a better representation of the distribution of travel times, thus subsuming the boost given by the Platoon Matching Algorithm. These two sets of results (VP, VP with PM) are "unsupervised methods", as they don't use "labels" (ground-truth trajectories).
- The algorithm seems quite robust to the effect of the noise vehicles for different scenarios. This is likely due to the efficacy of the Platoon Matching depends on how the probes (Virtual or Real) are able to better identify the crossing Green Time Phase windows at the first and last intersections. Even if more noise vehicles join the platoon exiting the corridor along the way, the algorithm tries to only match the initial number of vehicles that were detected at the first intersection (preferably of type platoon "P" but also isolated vehicles "V"). Let's say 10 vehicles grouped alongside the real probe at the first intersection, 2 of them left the corridor in the middle, and 5 more joined the platoon by the end of the corridor. At the last intersection, 13 vehicles would be detected alongside the real probe. The Platoon Matching algorithm will be constrained to match only 10 of those 13 vehicles, since only 10 vehicles are there in the first intersection's detector sequence. (If there happened to be fewer than 10 vehicles in that Green Phase Time Window (due to platoon break-up), the algorithm will try

TABLE III

COMPARISON OF DIFFERENT METHODS FOR VARIABLE AMOUNTS OF TRAFFIC

Results For Low Traffic

			Virtual Probe (VP)				VP with PM				VP and RP					RP with PM		
OD	Noise	Meas.	1_s	1	5	10	1_s	1	5	10	1/1	2/2	4/4	5/4	10/4	1	2	4
Low	Low	MAP	0.29	0.25	0.21	0.22	0.21	0.23	0.22	0.23	0.09	0.08	0.07	0.09	0.10	0.09	0.08	0.06
		STD	-	3.51	1.20	1.18	1.26	1.16	1.06	1.08	0.47	0.33	0.26	0.37	0.38	0.44	0.43	0.31
		HLD	1	0.95	0.82	0.71	0.76	0.64	0.52	0.50	0.75	0.72	0.52	0.57	0.45	0.55	0.27	0.11
	High	MAP	0.34	0.22	0.19	0.22	0.23	0.24	0.21	0.22	0.09	0.07	0.06	0.09	0.11	0.09	0.06	0.06
		STD	-	2.60	1.05	1.12	1.34	1.26	0.99	1.03	0.39	0.33	0.25	0.37	0.43	0.45	0.28	0.26
		HLD	0.96	0.96	0.82	0.71	0.74	0.65	0.46	0.40	0.81	0.64	0.54	0.56	0.44	0.46	0.15	0.08
Med.	Low	MAP	0.25	0.24	0.17	0.17	0.17	0.19	0.18	0.17	0.11	0.11	0.07	0.07	0.10	0.10	0.07	0.05
		STD	-	3.40	0.87	0.89	0.98	0.98	0.89	0.85	0.51	0.46	0.29	0.30	0.43	0.47	0.31	0.20
		HLD	0.96	0.97	0.78	0.65	0.69	0.56	0.38	0.38	0.85	0.72	0.57	0.55	0.44	0.64	0.32	0.17
	High	MAP	0.22	0.19	0.17	0.16	0.20	0.16	0.19	0.19	0.12	0.09	0.08	0.09	0.11	0.10	0.06	0.06
		STD	-	2.34	0.97	0.86	1.15	0.90	0.94	0.92	0.59	0.40	0.33	0.37	0.44	0.53	0.28	0.27
		HLD	1	0.91	0.77	0.68	0.71	0.57	0.45	0.43	0.78	0.69	0.58	0.54	0.49	0.60	0.27	0.16
High	Low	MAP	0.25	0.17	0.22	0.19	0.18	0.19	0.20	0.19	0.13	0.10	0.11	0.11	0.14	0.06	0.05	0.04
		STD	-	2.00	1.50	1.24	1.24	1.23	1.21	1.15	0.69	0.45	0.54	0.49	0.65	0.32	0.26	0.20
		HLD	1	0.94	0.88	0.77	0.69	0.61	0.52	0.50	0.81	0.65	0.54	0.53	0.49	0.62	0.39	0.18
	High	MAP	0.27	0.27	0.24	0.22	0.23	0.24	0.23	0.22	0.1	0.11	0.12	0.12	0.15	0.06	0.07	0.05
		STD	-	3.55	1.82	1.45	1.68	1.67	1.39	1.37	0.51	0.54	0.53	0.56	0.68	0.39	0.33	0.25
		HLD	1	0.97	0.89	0.79	0.78	0.70	0.58	0.58	1	0.70	0.59	0.54	0.50	0.63	0.29	0.16

Results For Medium Traffic

			Virtual Probe (VP)				VP with PM				VP and RP					RP with PM		
OD	Noise	Meas.	1_s	1	5	10	1_s	1	5	10	1/1	2/2	4/4	5/4	10/4	1	2	4
Low	Low	MAP	0.29	0.17	0.18	0.17	0.19	0.19	0.20	0.21	0.08	0.08	0.07	0.07	0.10	0.09	0.06	0.06
		STD	-	1.77	0.91	0.84	1.03	0.97	0.92	0.92	0.29	0.33	0.29	0.27	0.38	0.38	0.26	0.23
		HLD	0.96	0.96	0.78	0.58	0.73	0.59	0.45	0.39	0.67	0.56	0.37	0.37	0.35	0.39	0.06	0.02
	High	MAP	0.27	0.20	0.14	0.16	0.18	0.17	0.15	0.16	0.09	0.05	0.05	0.06	0.07	0.10	0.08	0.06
		STD	-	1.70	0.68	0.74	0.98	0.82	0.69	0.71	0.35	0.21	0.18	0.23	0.28	0.42	0.32	0.25
		HLD	0.88	0.92	0.71	0.59	0.72	0.59	0.37	0.37	0.7	0.55	0.39	0.40	0.32	0.35	0.07	0.01
Med.	Low	MAP	0.23	0.20	0.17	0.17	0.17	0.16	0.17	0.17	0.09	0.09	0.07	0.09	0.10	0.07	0.05	0.06
		STD	-	3.07	1.02	0.96	1.02	0.94	0.96	0.91	0.43	0.39	0.32	0.37	0.45	0.30	0.22	0.27
		HLD	0.92	0.96	0.72	0.65	0.69	0.63	0.46	0.43	0.78	0.57	0.41	0.41	0.33	0.40	0.14	0.06
	High	MAP	0.22	0.22	0.21	0.20	0.20	0.18	0.20	0.19	0.12	0.10	0.08	0.09	0.12	0.07	0.06	0.04
		STD	-	2.65	1.15	1.02	1.12	1.01	0.96	0.94	0.52	0.39	0.31	0.36	0.49	0.30	0.28	0.18
		HLD	0.92	0.92	0.82	0.65	0.73	0.61	0.44	0.43	0.74	0.60	0.43	0.42	0.37	0.43	0.13	0.06
High	Low	MAP	0.25	0.23	0.22	0.23	0.22	0.21	0.21	0.21	0.13	0.11	0.10	0.12	0.15	0.05	0.05	0.03
		STD	-	6.70	1.51	1.48	1.67	1.44	1.35	1.27	0.66	0.53	0.47	0.54	0.72	0.27	0.28	0.16
		HLD	1	0.90	0.85	0.79	0.74	0.60	0.52	0.51	0.71	0.57	0.41	0.43	0.38	0.46	0.13	0.07
	High	MAP	0.27	0.23	0.24	0.22	0.24	0.24	0.23	0.23	0.12	0.12	0.11	0.12	0.15	0.04	0.03	0.03
		STD	-	2.86	1.88	1.53	1.80	1.59	1.54	1.46	0.62	0.54	0.51	0.56	0.71	0.23	0.16	0.16
		HLD	1	0.94	0.88	0.82	0.76	0.69	0.64	0.64	0.72	0.59	0.41	0.40	0.39	0.45	0.11	0.05

Results For High Traffic

			Virtual Probe (VP)				VP with PM				VP and RP					RP with PM		
OD	Noise	Meas.	1_s	1	5	10	1_s	1	5	10	1/1	2/2	4/4	5/4	10/4	1	2	4
Low	Low	MAP	0.18	0.18	0.13	0.10	0.16	0.11	0.12	0.11	0.08	0.06	0.07	0.06	0.07	0.08	0.06	0.05
		STD	-	1.13	0.57	0.41	0.74	0.48	0.49	0.47	0.29	0.22	0.27	0.23	0.25	0.28	0.20	0.15
		HLD	0.88	0.82	0.69	0.55	0.67	0.40	0.24	0.25	0.66	0.53	0.36	0.32	0.31	0.36	0.05	0.01
	High	MAP	0.24	0.13	0.09	0.09	0.12	0.09	0.10	0.10	0.07	0.06	0.05	0.06	0.05	0.11	0.08	0.06
		STD	-	1.06	0.36	0.39	0.58	0.39	0.39	0.40	0.26	0.19	0.17	0.20	0.18	0.40	0.26	0.21
		HLD	0.96	0.91	0.63	0.52	0.63	0.41	0.23	0.21	0.72	0.54	0.33	0.34	0.27	0.37	0.06	0.01
Med.	Low	MAP	0.22	0.21	0.17	0.17	0.18	0.16	0.17	0.17	0.12	0.08	0.09	0.09	0.10	0.06	0.04	0.03
		STD	-	3.50	1.03	1.00	1.10	0.93	0.91	0.90	0.55	0.37	0.38	0.36	0.45	0.27	0.17	0.11
		HLD	1	0.93	0.76	0.65	0.67	0.46	0.34	0.33	0.72	0.53	0.32	0.34	0.33	0.34	0.05	0.02
	High	MAP	0.26	0.19	0.20	0.18	0.21	0.18	0.18	0.17	0.1	0.08	0.09	0.08	0.10	0.07	0.05	0.02
		STD	-	1.41	1.07	0.94	1.17	0.92	0.88	0.87	0.43	0.36	0.39	0.35	0.46	0.31	0.21	0.09
		HLD	1	0.94	0.78	0.66	0.71	0.47	0.30	0.29	0.67	0.51	0.35	0.34	0.33	0.36	0.06	0.01
High	Low	MAP	0.28	0.25	0.23	0.23	0.24	0.23	0.21	0.21	0.11	0.12	0.10	0.12	0.14	0.06	0.03	0.03
		STD	-	2.20	1.55	1.47	1.74	1.54	1.28	1.30	0.55	0.55	0.47	0.56	0.66	0.33	0.15	0.11
		HLD	1	0.95	0.79	0.72	0.72	0.53	0.44	0.44	0.72	0.51	0.36	0.38	0.30	0.34	0.07	0.02
	High	MAP	0.27	0.27	0.27	0.27	0.27	0.27	0.26	0.26	0.15	0.14	0.12	0.13	0.16	0.05	0.02	0.03
		STD	-	2.63	1.79	1.78	2.00	1.84	1.63	1.65	0.7	0.60	0.55	0.60	0.75	0.31	0.11	0.13
		HLD	0.96	0.96	0.86	0.81	0.76	0.67	0.59	0.58	0.7	0.52	0.37	0.35	0.30	0.35	0.04	0.02

The results obtained for low, medium and high traffic scenarios are presented according to the performance metrics (i) Average of Mean Absolute Percentage Error (MAP), Average of Normalized Absolute Difference in Means (STD), and by Hellinger Divergence (HLD). With fewer Virtual Probes (VP) the addition of Platoon Matching (PM) Algorithm (Single\1% VP vs. Single\1% VP with PM), significantly boosts results. Further, we see that the addition of real probe (RP) trajectories greatly improves performance (VP, VP with PM vs. VP and RP, RP with PM). Overall, we find that using real probe trajectories with platoon matching (RP with PM) gives us the best overall results, across all scenarios. Within this set, increasing the percentage penetration of real probes, increases the accuracy.

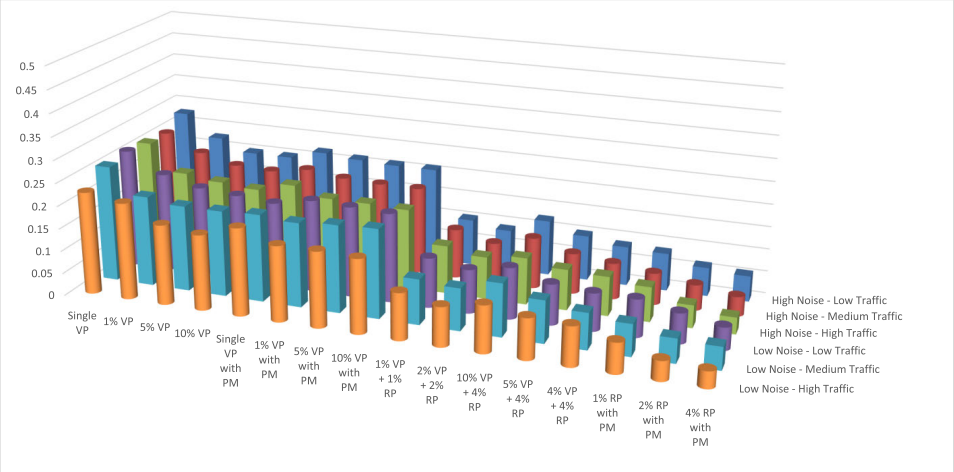

Fig. 8. Summarization of arterial travel times using the Mean Absolute Percentage Error (MAP) for various flow regimes. We see that the gradual addition of real probe trajectories (VP and RP, RP with PM) greatly improves performance (vs. just VP, VP with PM), as seen with declining MAP going left to right.

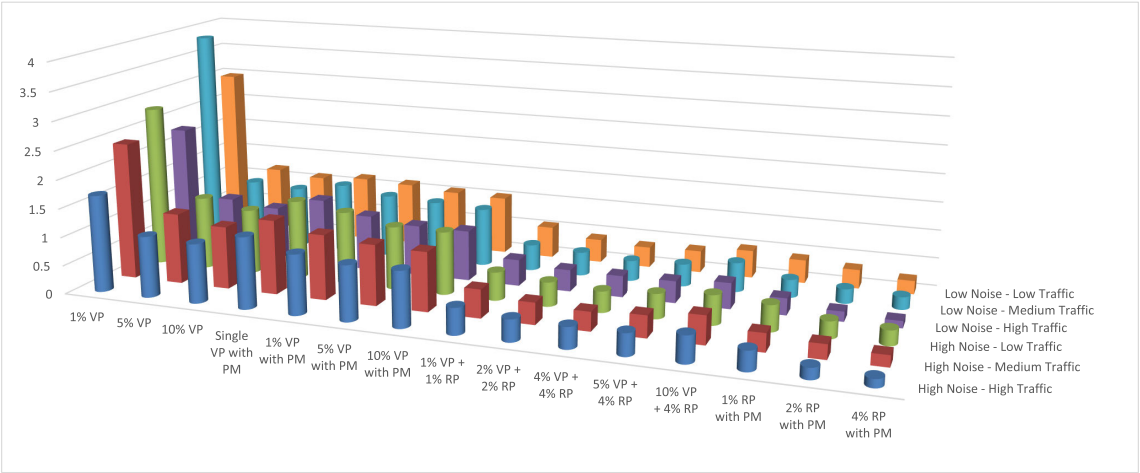

Fig. 9. Summarization of arterial travel times using the Standard Error Distance (STD) for various flow regimes. We see that the gradual addition of real probe trajectories (VP and RP, RP with PM) greatly improves performance (vs. just VP, VP with PM), as seen with declining STD going left to right.

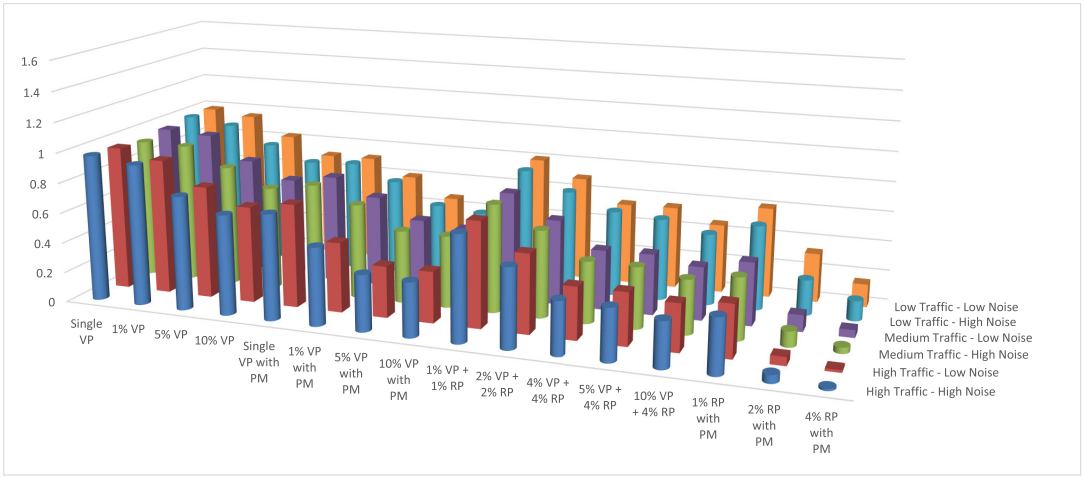

Fig. 10. Summarization of arterial travel times using the Hellinger Divergence (HLD) for various flow regimes. We see that the gradual addition of real probe trajectories (VP and RP, RP with PM) greatly improves performance (vs. just VP, VP with PM), as seen with declining HLD going left to right.

and match only that fewer number.) Further, due to the intermediate traffic lights, the noise vehicles effectively bunch up with the main platoon at stops, thus becoming a part of the platoon. Travel times of only 10 matches are calculated the Platoon Matching algorithm, but their distribution would be indicative of a hypothetical platoon of similar size, traversing the entire corridor without any addition/removal from it. Thus, the end-to-end travel time distribution estimate remains relatively unperturbed.

- While it was just a matter of additional computation to increase the number of Virtual Probes, it is not easy to greatly increase the number of real probes (i.e. connected

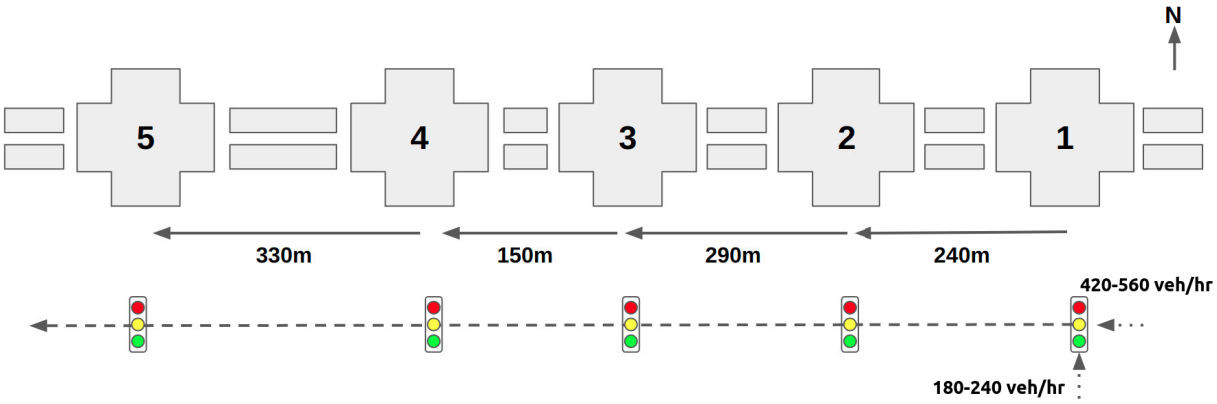

Fig. 11. Corridor used for real-world data collection. The corridor runs from East to West, with 2 lanes dedicated for through traffic along the 5-intersection stretch considered. Intersection 1 has two lanes for through traffic (along the Westbound approach) and one left-turning lane (along the Northbound approach), feeding into the corridor. We ignore the Southbound right-turning lane as it is not possible to determine from the detector actuation stream which particular vehicles took a right-turn into the corridor, and at what time. The traffic flow exiting the corridor varies between 600 to 800 vehicles per hour over the time period. The cycle length varies between 120 seconds and 190 seconds, with an average of 150 seconds. Cameras are mounted at the intersections, and data is collected for Westbound traffic. Vehicle tracking algorithms note the entry and exit times of vehicles, thus yielding the travel time distribution.

vehicles) in the real world. Thus, we restrict ourselves to 1%, 2% and 4% penetration of real probes.

- We see that the addition of real probe trajectories (VP and RP, RP with PM) greatly improves performance. These two sets of results (VP and RP, RP with PM) are "semi-supervised methods", as they use sparse "labels" (ground-truth trajectories) to guide the algorithm.
- Overall, we find that using real probe trajectories with platoon matching (RP with PM) gives us the best overall results, across all scenarios. Within this set, increasing the penetration of real probes, increases the accuracy.

D. Additional Simulation Results and Discussion

We also perform experiments investigating the effect of turn-in vehicles completing their journey across the corridor as well the effect of non-uniform cycle lengths. We present the results and discussions below.

1) Effect of Turn-In Vehicles in the Corridor: In the earlier section we assumed that the turn-in noise vehicles entered the corridor and exited at a random intersections. This was to simulate merging and lane-changing behaviors. A small portion (<5%) of these vehicles do exit alongside the main corridor flow, as the random vehicle trip generation algorithm used for these noise vehicles, generates such traffic. We now investigate the effect of the cases where a greater proportion of noise vehicles exit the corridor, after having joined the intersection in the middle. We test this for medium traffic flow (750 veh/hr) with low noise (10% extra) and high noise (20% extra) vehicles. We test the scenarios when <5%/40%/80% of the noise vehicles exit the corridor along with the main corridor flow. Note that the percentages (<5%/40%/80%) are for the (10% extra and 20% extra) noise vehicles, and not for the main corridor traffic flow (750 veh/hr).

We present the results in Table IV. We see that the error metrics for the experiments with a platoon of Virtual Probes with and without the Platoon Matching algorithm (10% VP and 10% VP with PM) perform about the same. Thus, the Platoon Matching algorithm does not seem to help in this case. This is likely due to the fact that the reconstructed trajectories are significantly erroneous, thus giving the Platoon Matching algorithm incorrect time windows to do the matching. However, with increasing percentage of Real Probe trajectories (1%/4%/10% RP with PM), the accuracy improves greatly (as evidenced by the declining error metrics). As explained earlier in subsection "Simulation Results and Discussion", the algorithm is robust to noise vehicles as it only tries to match the initial number of vehicles alongside the probe vehicle.

2) Effect of Uneven Cycle Lengths: In the earlier section, we assumed the cycle lengths and the green split for the corridor to be fixed to 120 seconds and 55 seconds respectively. In this section, we vary the cycle lengths of the intersections at random between 105 and 135 seconds, with green splits varying between 40 seconds and 70 seconds respectively. Thus, the intersections are no longer on a common cycle period, though they each have fixed cycle lengths (between 105 and 135 seconds). We test this for medium traffic flow (750 veh/hr) with low noise (10% extra) and high noise (20% extra) vehicles. We also assume that 40% of the noise vehicles exit the corridor along with the main corridor flow.

We present the results in Table V. We see that the error metrics for the experiments with a platoon of Virtual Probes with and without the Platoon Matching algorithm (10% VP and 10% VP with PM) perform about the same. However, with increasing percentage of Real Probe trajectories (1%/4%/10% RP with PM), the accuracy improves greatly (as evidenced by the declining error metrics). The algorithm seems quite robust regardless of the uneven cycle lengths, under both high and low noise scenarios when real probes are used. Here too, this is likely due to the fact that the initial reconstructed trajectories by the Virtual Probe method are significantly erroneous, thus giving the Platoon Matching algorithm incorrect time windows to do the matching. Whereas real probe trajectories are accurate, along with the fact that only the initial number of vehicles will be matched by the Platoon Matching algorithm, both of which further constrain the potential for error.

3) Vehicle-Matching Accuracy of the Platoon Matching Algorithm: We also conduct experiments assessing the vehicle-matching accuracy of the Platoon Matching

TABLE IV
RESULTS FOR THE EFFECT OF TURN-IN VEHICLES IN THE CORRIDOR

Noise Veh.	Corridor Journey %	Meas.	10% VP	10% VP with PM	1% RP with PM	4% RP with PM	10% RP with PM
Low	<5%	MAP	0.27	0.32	0.14	0.06	0.03
		STD	1.18	1.38	1.00	0.29	0.14
		HLD	0.52	0.65	0.81	0.39	0.27
	40%	MAP	0.27	0.34	0.15	0.08	0.04
		STD	1.20	1.36	1.21	0.38	0.19
		HLD	0.58	0.71	0.81	0.52	0.30
	80%	MAP	0.29	0.33	0.19	0.12	0.04
		STD	1.42	1.47	1.38	0.71	0.19
		HLD	0.64	0.70	0.86	0.51	0.30
High	<5%	MAP	0.30	0.29	0.14	0.05	0.03
		STD	1.19	1.33	0.82	0.22	0.13
		HLD	0.67	0.66	0.71	0.38	0.23
	40%	MAP	0.24	0.28	0.14	0.10	0.06
		STD	1.20	1.32	0.85	0.53	0.29
		HLD	0.59	0.65	0.76	0.46	0.23
	80%	MAP	0.25	0.30	0.15	0.11	0.10
		STD	1.22	1.33	0.93	0.57	0.48
		HLD	0.60	0.62	0.80	0.52	0.27

We see that the error metrics for the experiments with a platoon of Virtual Probes with and without the Platoon Matching algorithm (10% VP and 10% VP with PM) perform about the same. Thus, the Platoon Matching algorithm does not seem to help in this case. This is likely due to the fact that the initial reconstructed trajectories are significantly erroneous, thus giving the Platoon Matching algorithm incorrect time windows to do the matching. However, with increasing percentage of Real Probe trajectories (1%/4%/10% RP with PM), the accuracy improves greatly (as evidenced by the declining error metrics). The algorithm seems quite robust regardless of the turn-in behavior of the noise vehicles, under both high and low scenarios when real probes are used. This is likely due to the fact that the real probes are able to better identify the Green Time Phase windows at the first and last intersections. Even if more noise vehicles join the platoon exiting the corridor along the way, the algorithm tries to only match the initial number of vehicles that were detected at the first intersection.

TABLE V
RESULTS FOR THE EFFECT OF UNEVEN CYCLE LENGTHS

Noise Veh.	Meas.	10% VP	10% VP with PM	1% RP with PM	4% RP with PM	10% RP with PM
Low	MAP	0.44	0.43	0.18	0.09	0.02
	STD	2.67	2.31	1.30	0.57	0.26
	HLD	0.94	0.88	0.88	0.63	0.31
High	MAP	0.41	0.40	0.11	0.07	0.04
	STD	2.41	2.26	1.06	0.53	0.22
	HLD	0.92	0.85	0.84	0.52	0.36

We see that the error metrics for the experiments with a platoon of Virtual Probes with and without the Platoon Matching algorithm (10% VP and 10% VP with PM) perform about the same. However, with increasing percentage of Real Probe trajectories (1%/4%/10% RP with PM), the accuracy improves greatly (as evidenced by the declining error metrics). Here too, this is likely due to the fact that the initial reconstructed trajectories by the Virtual Probe method are significantly erroneous, thus giving the Platoon Matching algorithm incorrect time windows to do the matching. Whereas real probe trajectories are accurate, along with the fact that only the initial number of vehicles will be matched by the Platoon Matching algorithm, both of which further constrain the potential for error.

Algorithm. Specifically, we want to know if the pairs of loop detector actuations matched by the Needlman-Wunsch Algorithm (in order to estimate individual travel times) indeed belong to the same vehicle (or at least are close). Given that loop detector data does not allow re-identification, it is not possible to know which vehicle caused which actuation in the loop detector data. However, in simulation, we can find this, since we know the trajectories of all vehicles, not just the ones we designate as probes.

Our Platoon Matching Algorithm does not use vehicle re-identification information in any way. In the real world, some vehicles may pass others, leave the corridor, and may be joined by new vehicles coming onto the corridor. This information is not available to the Platoon Matching Algorithm, which just tries to match two sequences of loop detector actuations by distorting them the least.

We conduct experiments investigating the vehicle matching i.e. if the matches returned by the PM Algorithm capture the presence of the same vehicle. Let us say a non-probe vehicle crossed the first intersection at t = 52 and the last intersection at t = 198. Hence, we expect actuations at the two stop-bars loop detectors respectively. Since this is not a probe vehicle, our algorithm does not know this information of the crossing times i.e. (52, 198).

When the PM Algorithm matches the first and last intersection's loop detector strings, it will return a collection of purported crossing times. If these times are within a time window of +/- k (where k = 5, 10, 20 seconds), we assume it is a match, i.e. the vehicle was "re-identified" in a sense. Hence, if among the times returned by the algorithm include (49, 200), the above-mentioned non-probe vehicle of (52, 198) would be considered a match. Given the simulation conditions, it is near-impossible for the PM Algorithm to return a perfect match i.e. k = 0.

Simulation was performed for medium traffic volume (750 veh/hr) with the same conditions mentioned in Section IV. Offset Deviation (OD) and the amount of Noise vehicles were varied, as mentioned in that section. 10% of the vehicles were chosen as probe vehicles, and their trajectories were used along with the Platoon Matching Algorithm. Since the travel times seen are in the order of 250 seconds generally, we can express the windows as 2%, 4%, 8% as well. We report the accuracy of vehicle matching as the percentage of vehicles successfully matched, given a certain allowable laxity of time window size. Table VI shows the results for the Vehicle Matching metrics of the Platoon Matching Algorithm.

We can see that as the window is increased, the accuracy of vehicle-matching increases from around 14% to over 60%. It must be remembered that vehicle-matching (vehicle re-identification) is not the aim of this method, rather it is to estimate travel time distributions.

E. Real-World Results and Discussion

The above experiments and results show the efficacy of the methods presented using simulated data. We now verify our approach on real-world data collected using high-resolution loop detectors and video cameras along a busy 5-intersection signalized urban corridor in Gainesville, Florida, USA. The corridor starts at a major intersection (West University Avenue and 13th Street), with the West-bound direction being considered for analysis. The corridor borders a large University campus with dense residential areas surrounding it. The duration of analysis is between 11 am and 4 pm, with 600-800 vehicles per hour exiting the corridor along the West-bound

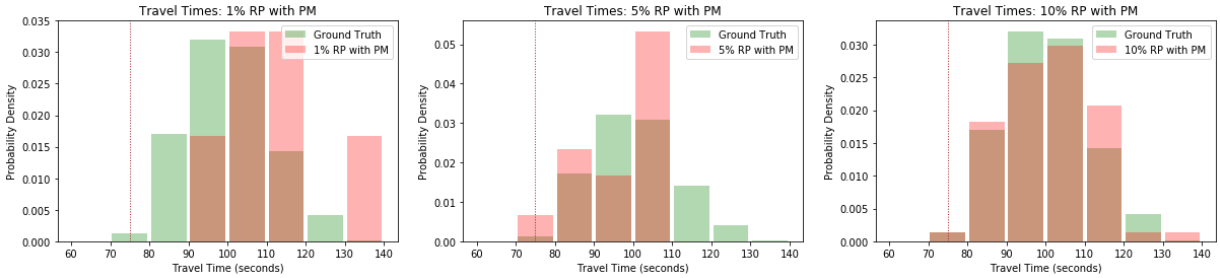

Fig. 12. We can see 3 histograms showing distributions obtained when 1%/5%/10% Real Probe trajectories are used with Platoon Matching. The plots show the Probability Density, with the area under the entire distribution summing to 1[7]. The vertical dashed line at 75 seconds indicates the free-flow travel time. As the percentage of real trajectories increases, there is greater overlap between the Ground Truth Distribution (green) and the distribution obtained by using Real Probe trajectories with Platoon Matching (red).

TABLE VI

RESULTS FOR THE ACCURACY OF THE PLATOON-MATCHING ALGORITHM

OD	Noise	Within +/- 2% (5s)	Within +/- 4% (10s)	Within +/- 8% (20s)
Low	L	12.09%	29.81%	63.51%
	H	15.89%	32.55%	66.94%
Med.	L	16.12%	35.48%	61.29%
	H	19.96%	31.12%	51.79%
High	L	11.72%	26.39%	59.68%
	H	14.85%	29.90%	63.47%

We can see that as the window is increased, the accuracy of vehicle-matching increases from around 14% to over 60%. It must be remembered that vehicle-matching (vehicle re-identification) is not the aim of this method, rather it is to estimate travel time distributions.

TABLE VII

RESULTS FOR REAL-WORLD TRAFFIC

	1% RP with PM	5% RP with PM	10% RP with PM
MAP	0.049	0.014	0.0006
STD	0.367	0.078	0.004
HLD	0.333	0.138	0.110

We selected 1%, 5% and 10% of these ground-truth trajectories as "Real Probes" (RP) and used them with the Platoon Matching algorithm (i.e. RP with PM). We compute the same distance metrics i.e. Average of Mean Absolute Percentage Error (MAP), Average of Normalized Absolute Difference in Means (STD) and Hellinger Divergence (HLD) as before. With the addition of increasing percentage of real world probe trajectories, we can see the improved performance of our approach as evidenced by declining MAP, STD and HLD measures, going left to right. This is the same trend as seen in the simulation results.

direction. The two major contributing flows are the Through traffic along West University Avenue (70%, including the Right-turning traffic coming from the north), and the Left-turning traffic (30%) approaching from the south on 13th Street. The turn-in traffic from the other 4 intersections was found to be about 10% of the flow volume. Detailed flow volumes of the side streets are not available. We obtain high-resolution loop detector data and signal state data at 10 Hz resolution from deployed equipment at the first intersection (West University Avenue and 13th Street). This data is then aggregated at 1 Hz resolution. Video cameras along the West-bound direction were used for re-identifying vehicles at the first and last intersections (details below). Figure 11 shows the diagram for the 5-intersection corridor considered.

An important aspect of the real-world data was that the left-turning traffic accounted for over 30% of the traffic entering the corridor (at the first intersection). In the simulations, where over 80% of the traffic was through-traffic, we could ignore the contribution of left-turning traffic feeding into the corridor. In order to account for both the left-turning and through input traffic components, the PM algorithm was run twice, once for the left-turning traffic and once for the through-traffic. This was done by using the different green-time windows at the first intersection, once when the through-movement was allowed and once when the left-turning movement. These were matched with the green-time windows at the last intersection where the vehicles exit. Travel times obtained from both were combined to get the estimated travel time distribution.

For verification, ground-truth travel time distribution was computed using intersection-mounted cameras paired with an automated computer vision-based vehicle tracking algorithm [31]. Such vision-based methods can yield a reasonable estimate of the actual travel time distribution, by re-identifying the same vehicles at successive intersections and calculating their time differences. Using methods outlined in [31], we were able to obtain entry and exit times of vehicles that were crossing the corridor for the duration of interest. This travel time distribution was used as ground-truth to verify our algorithm. “Real Probes” here refer to ground-truth entry-exit times obtained that we give to our PM algorithm; e.g. 1% Real Probes (RP) means we randomly sample 1% of all collected ground-truth entry-exit times (at first and last intersections) and use them with the PM algorithm.

A similar trend as seen in the simulation results (Table III) is seen for the real-world verification results presented in Table VII. As the percentage penetration of real probes increases from 1% to 10%, the accuracy of the algorithm increases, as evidenced by decreasing distance measures, left to right. We can see this visually in Figure 12. We can see 3 histograms showing distributions obtained when 1%/5%/10% Real Probe trajectories are used with Platoon Matching. The plots show the Probability Density, with the area under the entire distribution summing to 1[8]. The vertical dashed line at 75 seconds indicates the free-flow travel time. As the percentage of real trajectories increases, there is greater overlap between the Ground Truth Distribution (green) and the distribution obtained by using Real Probe trajectories with Platoon Matching (red). Thus, we have verified the performance of our algorithm on both simulated data and real-world data.

[8]matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html

V. Conclusion and Future Work

The goal of this work is the estimation travel time distributions in arterial roads using (i) High-resolution loop detector data, (ii) Kinematics-based models of traffic movement (via Virtual Probe model) and (iii) Connected Vehicle (Real Probe) data (when available). Studying travel time distributions is useful for improving traffic management policies for arterials (such as signal re-timing efforts) etc. which can reduce overall travel time in networks.

A causal survey of the literature would lead us to believe (prior to any empirical evidence), that sound kinematic models of traffic (in the form of virtual probes) would be sufficient to estimate travel time distributions when only high-resolution loop detector data is available. Instead, our results show that a combination of virtual (reconstructed) or real probes (akin to semi-supervised labels in machine learning, obtained from a small subset of connected vehicles) trajectories, combined with a sequence alignment algorithm for platoon matching, perform the best. To reach this conclusion, we ran comprehensive tests in different traffic situations on four different algorithm scenarios: (i) Virtual Platoons alone, (ii) Virtual Platoons with Platoon Matching, (iii) Virtual Platoons and Real Probes, and (iv) Real Probes with Platoon Matching. While in some situations the combination of virtual and real probes returned promising results, it is the combination of a small number of real probes (the semi-supervised situation referred to above) and platoon matching that performed the best. We also verified our algorithm using real-world data and obtained the same conclusion.

Our results suggest:

- Introduction of trajectory information of even a small number of "labeled" vehicles (i.e. real probes), greatly improves the results obtained by methods relying on loop detector data alone. As loop detectors are fairly widespread in developed countries, this work provides support for increased investment in obtaining a small amount of "labels" from additional sensor modalities like GPS, BlueTooth, DSRC, Computer-Vision based vehicle tracking algorithms etc.
- Platoon Matching methods (such as using sequence alignment), which analyze loop detector data for platoon movements, can greatly magnify sparse probe trajectory data, yielding many more estimated travel times.
- Our method is tested both in simulation and using real-world data and is robust to noisy traffic. It can also be used when the left-turning traffic feeding into the corridor is significant compared to the through-traffic by running the algorithm twice and merging the travel time distributions.

We also plan to conduct a more comprehensive three-way comparison between virtual probes, real probes and platoon matching, and use the "best of breed" methods in the creation of better signal policies.

We intend to continue this line of research with more emphasis on Connected Vehicles. In particular, we wish to determine how to better fuse loop detector and Connected Vehicle data from public transit buses for real-time dynamic estimation of travel times, including during traffic incidents and blockages. Also, since our techniques can output level of service measures such as queue lengths at intersections, number of stops etc., we hope to study them to guide signal timing policies at a city-wide scale.

VI. Acknowledgment

The authors would also like to thank the City of Gainesville (Florida). The opinions, findings and conclusions expressed in this publication are those of the author(s) and not necessarily those of the Florida Department of Transportation or the National Science Foundation.

References

[1] S. El-Tantawy and B. Abdulhai, "Multi-agent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC)," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 319–326.

[2] J. I. Levy, J. J. Buonocore, and K. von Stackelberg, "Evaluation of the public health impacts of traffic congestion: A health risk assessment," *Environ. Health*, vol. 9, no. 1, p. 65, Dec. 2010.

[3] K. Zhang and S. Batterman, "Air pollution and health risks due to vehicle traffic," *Sci. Total Environ.*, vols. 450–451, pp. 307–316, Apr. 2013.

[4] H. X. Liu, W. Ma, H. Hu, X. Wu, and G. Yu, "SMART-SIGNAL: Systematic monitoring of arterial road traffic signals," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2008, pp. 1061–1066.

[5] R. L. Gordon, *Traffic Signal Retiming Practices in the United States*, vol. 409. USA: National Academies Press, 2010.

[6] H. X. Liu and W. Ma, "A virtual vehicle probe model for time-dependent travel time estimation on signalized arterials," *Transp. Res. C, Emerg. Technol.*, vol. 17, no. 1, pp. 11–26, Feb. 2009, doi: 10.1016/j.trc.2008.05.002.

[7] M. J. Lighthill and G. B. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proc. Roy. Soc. London, Ser. A, Math. Phys. Sci.*, vol. 229, pp. 317–345, May 1955.

[8] A. Skabardonis and N. Geroliminis, "Real-time estimation of travel times on signalized arterials," in *Transportation and Traffic Theory: Flow, Dynamics and Human Interaction: Proceedings of the 16th International Symposium on Transportation and Traffic Theory*. College Park, MD, USA: Univ. of Maryland, Jul. 2005.

[9] A. Skabardonis and N. Geroliminis, "Real-time monitoring and control on signalized arterials," *J. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 64–74, 2008.

[10] N. Geroliminis and A. Skabardonis, "Identification and analysis of queue spillovers in city street networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1107–1115, Dec. 2011.

[11] H. X. Liu, W. Ma, X. Wu, and H. Hu, "Real-time estimation of arterial travel time under congested conditions," *Transportmetrica*, vol. 8, no. 2, pp. 87–104, Mar. 2012.

[12] A. Bhaskar, E. Chung, and A.-G. Dumont, "Fusing loop detector and probe vehicle data to estimate travel time statistics on signalized urban networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 26, no. 6, pp. 433–450, Aug. 2011.

[13] R. L. Cheu, Q. Liu, and D.-H. Lee, "Arterial travel time estimation using SCATS detectors," in *Applications of Advanced Technologies in Transportation*. USA: American Society of Civil Engineers Library, 2002, pp. 32–39.

[14] L. Wei, Y. Wang, and P. Chen, "A particle filter-based approach for vehicle trajectory reconstruction using sparse probe data," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 2878–2890, May 2021.

[15] Q. Bing, D. Qu, X. Chen, F. Pan, and J. Wei, "Arterial travel time estimation method using SCATS traffic data based on KNN-LSSVR model," *Adv. Mech. Eng.*, vol. 11, no. 5, May 2019, Art. no. 168781401984192.

[16] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[17] H. Liu, H. Van Zuylen, H. Van Lint, and M. Salomons, "Predicting urban arterial travel time with state-space neural networks and Kalman filters," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1968, no. 1, pp. 99–108, Jan. 2006.

[18] L. Tran, M. Y. Mun, M. Lim, J. Yamato, N. Huh, and C. Shahabi, "DeepTRANS: A deep learning system for public bus travel time estimation using traffic forecasting," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 2957–2960, Aug. 2020.
[19] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen, "Estimating arterial traffic conditions using sparse probe data," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 929–936.
[20] D. E. Lucas, P. B. Mirchandani, and N. Verma, "Online travel time estimation without vehicle identification," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1867, no. 1, pp. 193–201, Jan. 2004.
[21] A. Karimpour, A. Ariannezhad, and Y. Wu, "Hybrid data-driven approach for truck travel time imputation," *IET Intell. Transp. Syst.*, vol. 13, no. 10, pp. 1518–1524, Oct. 2019.
[22] L. Zhu, F. Guo, J. W. Polak, and R. Krishnan, "Urban link travel time estimation using traffic states-based data fusion," *IET Intell. Transp. Syst.*, vol. 12, no. 7, pp. 651–663, Sep. 2018.
[23] C. C. Sun, G. S. Arr, R. P. Ramachandran, and S. G. Ritchie, "Vehicle reidentification using multidetector fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 3, pp. 155–164, Sep. 2004.
[24] M. Ndoye, V. F. Totten, J. V. Krogmeier, and D. M. Bullock, "Sensing and signal processing for vehicle reidentification and travel time estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 119–131, Mar. 2011.
[25] E. Hans, N. Chiabaut, and L. Leclercq, "Clustering approach for assessing the travel time variability of arterials," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2422, no. 1, pp. 42–49, Jan. 2014, doi: 10.3141/2422-05.
[26] A. Hofleitner, R. Herring, and A. Bayen, "Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning," *Transp. Res. B, Methodol.*, vol. 46, no. 9, pp. 1097–1122, Nov. 2012.
[27] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, no. 3, pp. 443–453, Mar. 1970.
[28] S. Lee and K. Jang, "Regularity of vehicle trips in urban areas," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2651–2658.
[29] F. Crawford, D. P. Watling, and R. D. Connors, "Identifying road user classes based on repeated trip behaviour using Bluetooth data," *Transp. Res. A, Policy Pract.*, vol. 113, pp. 55–74, Jul. 2018.
[30] G. A. Davis and I. Chatterjee, "Using detailed signal and detector data to investigate intersection crash causation," Dept. Civil Eng., Center Transp. Stud., Univ. Minnesota, USA, Tech. Rep. CTS 13-05, 2013.
[31] X. Huang, P. He, A. Rangarajan, and S. Ranka, "Intelligent intersection: Two-stream convolutional networks for real-time near-accident detection in traffic video," *ACM Trans. Spatial Algorithms Syst.*, vol. 6, no. 2, pp. 1–28, Jun. 2020.
[32] J. R. Sturdevant et al., "Indiana traffic signal Hi resolution data logger enumerations," Purdue e-Pubs, USA, Tech. Rep., 2012, doi: 10.4231/K4RN35SH.
[33] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.
[34] *SUMO Documentation: Definition of Vehicles, Vehicle Types, and Routes*, German Aerosp. Center, Germany, Feb. 2020.
[35] E. Hellinger, "Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen," *J. Für Die Reine Und Angewandte Mathematik*, vol. 1909, no. 136, pp. 210–271, 1909.
[36] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

Rahul Sengupta is currently pursuing the Ph.D. degree with the Computer and Information Science Department, University of Florida, Gainesville, USA. His research interests include applying machine learning models to sequential and time-series data, especially in the field of transportation engineering.

Rohith R. K. Reddy received the Bachelor of Science degree in computer science engineering from the Indian Institute of Technology Hyderabad, India, and the Master of Science degree in computer science from the University of Florida, USA. His research interests include algorithms, databases, machine learning, and deep learning.

Parth Shah received the bachelor's degree in electronics and electrical engineering from BITS Pilani, India, and the master's degree in computer science from the University of Florida in 2021. His research interests include traffic optimization, deep learning, and data compression.

James Dika received the bachelor's degree in computer science from the University of Florida.

Xiaohui Huang received the Ph.D. degree from the Department of Computer and Information Science and Engineering, University of Florida, in December 2020. She is currently an industry research scientist. Her research interests include machine learning, computer vision, and intelligent transportation systems.

Anand Rangarajan (Member, IEEE) is currently a Professor with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA. His research interests include computer vision, machine learning, medical and hyperspectral imaging, and the science of consciousness.

Sanjay Ranka (Fellow, IEEE) is currently a Distinguished Professor with the Department of Computer Information Science and Engineering, University of Florida. His current research interests include machine learning, the Internet of Things, and cloud computing for transportation and health care. His research is currently funded by NIH, NSF, USDOT, DOE, and FDOT. From 1999 to 2002, he was the Chief Technology Officer and the Co-Founder of Paramark, Sunnyvale, CA, USA, where he conceptualized and developed a machine learning-based service for optimizing advertising campaigns, which was acquired in 2002.