# Probabilistic Image-Driven Traffic Modeling via Remote Sensing

Scott Workman and Armin Hadzic

DZYNE Technologies

**Abstract.** This work addresses the task of modeling spatiotemporal traffic patterns directly from overhead imagery, which we refer to as image-driven traffic modeling. We extend this line of work and introduce a multi-modal, multi-task transformer-based segmentation architecture that can be used to create dense city-scale traffic models. Our approach includes a geo-temporal positional encoding module for integrating geo-temporal context and a probabilistic objective function for estimating traffic speeds that naturally models temporal variations. We evaluate our method extensively using the Dynamic Traffic Speeds (DTS) benchmark dataset and significantly improve the state-of-the-art. Finally, we introduce the DTS++ dataset to support mobility-related location adaptation experiments.

**Keywords:** Traffic Modeling · Geo-Temporal Context · Remote Sensing

## 1 Introduction

The relationship between humans, the physical environment, and motion guides a number of real-world applications. As Chen et al. [6] note, "transportation researchers have long sought to develop models to predict how people travel in time and space and seek to understand the factors that affect travel-related choices." For example, mobility has been shown to have a strong influence on traffic safety [43], public health [33], housing prices [4], and more. Recently, with the growth of autonomous driving efforts, there has been an increased focus on using vision-based methods to characterize the physical environment [16,23] and how it is traversed [9,45].

A primary challenge that remains is how to scale mobility-related analysis to the size of a city. For example, traffic speed data is predominately collected using fixed-point sensors deployed at static locations on roads [3]. Though an increasing amount of traffic speed data is available from alternative sources, such as automatic vehicle location systems, much of this data is still proprietary [32]. Even in the best case scenario, not every road is traversed at every possible time, resulting in partial, incomplete models of traffic flow (Figure 1, left).

This work addresses the task of using overhead imagery to directly model spatiotemporal mobility patterns, which we refer to as image-driven traffic modeling. We introduce a multi-modal, multi-task transformer-based segmentation architecture that operates on overhead imagery and can be used to create dense,

**Fig. 1:** We propose a method for image-driven traffic modeling, which can be used to create dense city-scale traffic models. (left) Historical ground-truth traffic data is often sparse as not all roads are traversed at all times. For example in Brooklyn on Monday at 8am, many roads are missing empirical speed data. (right) Our method can create a dense model of traffic flow at the same time.

city-scale models of traffic flow (Figure 1, right). Our approach integrates geo-temporal context (i.e., geographic location and time metadata) to enable location and time-dependent traffic speed predictions, along with two auxiliary tasks (road segmentation and orientation estimation). These auxiliary tasks provide synergy for traffic speed estimation through multi-task learning, as well as allowing our approach to generalize to locations where the road network is unknown or has changed.

Our approach has several key components. First, we integrate context into our method by introducing a novel geo-temporal positional encoding (GTPE) module. GTPE operates on geo-temporal context through three distinct pathways corresponding to location, time, and space-time features, ultimately producing a positional encoding that captures mobility-context. Second, we propose a probabilistic formulation for estimating traffic speeds that naturally models temporal variation in empirical speed data. Specifically, we capture uncertainty by estimating per-pixel prior distributions over traffic speeds, instead of regressing traffic speeds directly. To support this, we introduce an objective function that explicitly accounts for the number of traffic observations at a given time, during model training. This allows our method access to an implicit form of confidence in the underlying traffic speed averages for a given road segment.

Extensive experiments on a recent benchmark dataset, including ablation studies, demonstrate how our approach achieves superior performance versus

baselines. In addition, we extend this dataset to include a new, diverse, city and show how it can be used to support mobility-related location adaptation experiments. We also illustrate several scenarios where our approach could be applied to urban planning applications. Overall, the contributions of this work can be summarized as follows:

- a multi-modal, multi-task architecture for image-driven traffic modeling,
- a novel approach for integrating geo-temporal context in the form of a geo-temporal positional encoding module,
- a probabilistic formulation for estimating traffic speeds that incorporates knowledge of the number of traffic observations as a form of confidence,
- an extensive evaluation of our methods on the Dynamic Traffic Speeds (DTS) benchmark, achieving state-of-the-art results,
- and a new dataset, DTS++, to support mobility-related location adaptation experiments.

## 2 Related Work

Traffic modeling is important for many applications including urban planning [19] and autonomous driving [5]. These applications ultimately require: 1) descriptions of the underlying environment and 2) knowledge of how environments are traversed. For the former, descriptions of the static environment might include properties such as the location of roads, number of lanes, traffic directions, traffic signs, etc. For the latter, information relating human activity to the underlying infrastructure is necessary, such as time-varying traffic speeds, models of traffic congestion and safety, and other characterizations of driver and pedestrian behavior. Decades of research has focused on understanding related topics in urban mobility.

Numerous learning-based methods have been proposed for relating properties of the environment to how it is traversed. For example, recent work in autonomous driving has explored lane detection from ground-level images [14] and how to estimate road layout and vehicle occupancy in top views given a front-view monocular image [50]. Related to motion, Chen et al. [7] use traffic cameras to generate mobility statistics from pedestrians and vehicles. Kumar et al. [26] propose a method for estimating traffic flow using vehicle-mounted cameras and showcase it in a popular driving simulator. Zhang et al. [51] introduce a graph convolutional framework for traffic forecasting that captures spatiotemporal dependencies. Many other vision-based methods have been proposed for estimating vehicle speeds from ground-level imagery; refer to [15] for an in-depth overview.

Though much progress has been made, a primary challenge that remains is how to build descriptions of the environment at a global scale. For example, traditional approaches for modeling traffic flow assume prior knowledge of the road network and its connectivity [17], as well as the existence of large quantities of historical traffic data for each road segment [27]. As such, these methods are unable to function in locations where the road network is unknown or where

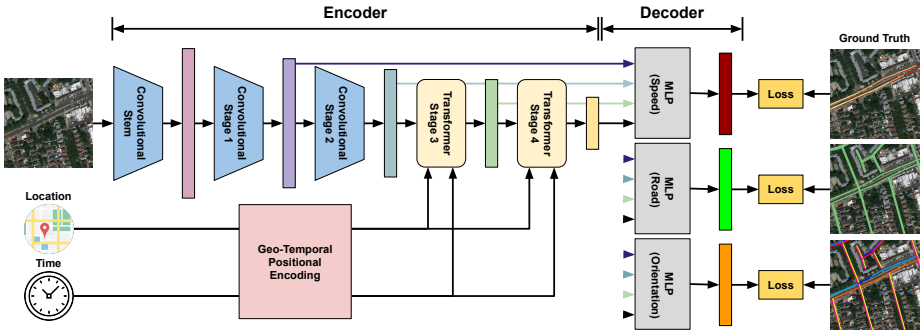


**Fig. 2:** An overview of our architecture for image-driven traffic modeling.

road segments have insufficient historical data. To circumvent issues of scale, overhead imagery has become a useful resource for many tasks due to its dense coverage, high resolution, and ever increasing revisit rates [2, 30].

For traffic modeling, overhead imagery has been used to automatically generate maps of road networks [34, 52], detect lane boundaries [20], understand roadway safety [36], approximate traffic noise [12], estimate emissions [35], and many other forms of image-driven mapping [38, 39, 42, 46–48]. Related to our work, Song et al. [41] and Hadzic et al. [18] show how overhead imagery can be used to understand motion in the form of free-flow traffic speeds (i.e., average speed without adverse conditions). Workman et al. [45] introduce the task of dynamic traffic modeling and a new benchmark dataset for traffic speed estimation. We extend this line of work in several ways, including introducing a probabilistic formulation for estimating traffic speeds.

## 3 An Architecture for Image-Driven Traffic Modeling

We address the problem of image-driven traffic modeling using overhead imagery. Our network architecture, depicted in Figure 2, has three inputs: an overhead image, $S(l)$, a geolocation, $l$, and a time, $t$. It has three outputs, corresponding to our primary task of traffic speed estimation and two auxiliary tasks: road segmentation and orientation estimation. We start with an overview of our multi-task transformer-based segmentation architecture (Section 3.1). Next, we describe our approach for integrating geo-temporal context via a novel geo-temporal positional encoding module (Section 3.2). Then, we describe our proposed probabilistic formulation for estimating traffic speeds (Section 3.3). Finally, we detail the loss functions for the auxiliary tasks (Section 3.4). Please refer to the supplemental material for an in-depth description of architecture design choices.

### 3.1 Architecture Overview

Our segmentation architecture has two primary components: 1) a multi-stage visual encoder for extracting features from an input overhead image, and 2) task-specific decoders which use these features to generate a segmentation output.

*Multi-stage Visual Encoder* Drawing inspiration from CoAtNet [10], which demonstrates that combining convolutional and attention layers can achieve better generalization and capacity, we use a robust multi-stage pipeline for visual feature encoding that integrates both convolutional stages and transformer stages. First, an input image is passed through an initial convolutional stem with three convolutional layers (each with BatchNorm and ReLU), downsampling the spatial resolution by two. This is followed by two convolutional stages, each of which uses multiple inverted residual blocks in the style of EfficientNet [21]. Following the two convolutional stages are two transformer stages inspired by SwinV2 [28]. Each stage consists of an overlapping patch embedding [44] followed by a sequence of multi-head self-attention (MHSA) blocks to allow global information to be propagated throughout the visual features. Each self-attention block uses a learned relative positional encoding [24, 29].

*Task-Specific MLP Decoder* To support semantic segmentation, we extend the multi-stage visual encoder with a task-specific decoder for each prediction task. Taking inspiration from SegFormer [49], we use a decoder consisting only of linear layers. Features from each encoder stage are fused via the following process. First, a stage-specific linear layer is used to unify the number of channels across features. The resulting features are then upsampled to a fourth of the resolution of the input image using bilinear interpolation and are then concatenated. This is followed by an additional linear layer that fuses information across features. To produce the segmentation output, a final linear layer operates on the fused feature and generates a task-specific number of outputs. Then, we rescale the output to match the spatial resolution of the input image.

### 3.2 Incorporating Geo-Temporal Context

We infuse our model with an understanding of location and time metadata by introducing a geo-temporal positional encoding module, shown visually in Figure 3.

**Geo-Temporal Positional Encoding** We structure the geo-temporal positional encoding (GTPE) module across three pathways corresponding to location, time, and space-time (referred to as loc+time). The location and time pathways reflect that location and time have utility independently, while the space-time pathway allows them to interact. Ultimately, the output of each pathway is added together to form a geo-temporal positional encoding. Next, we detail our choice of context parameterizations and encoding network.

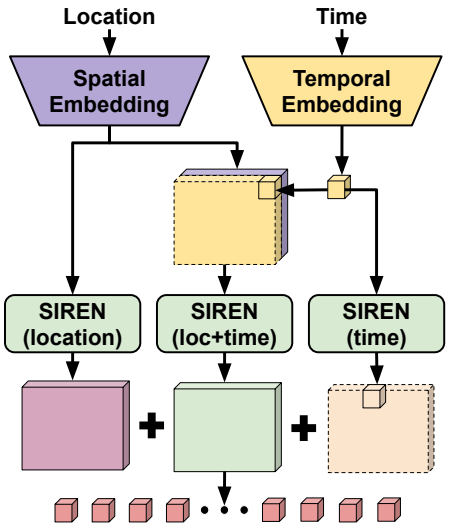


**Fig. 3:** An overview of our proposed geo-temporal positional encoding module.

*Location* Overhead imagery is unique in that images are typically georeferenced, enabling computation of the world coordinate of each pixel. We leverage this to generate dense location maps in easting ($x$-axis) and northing ($y$-axis) web mercator world coordinates, which are then normalized to a $[-1, 1]$ range using the bounds of the input region.

*Time* We represent temporal context using day of week, $d$, and hour of day, $h$, each scaled to $[-1, 1]$. These variables are parameterized using a cyclic representation, similar to Aodha et al. [31]:

$$f(d, h) = [\sin(\pi d), \cos(\pi d), \sin(\pi h), \cos(\pi h)]. \tag{1}$$

*Contextual Encoding* We select SIREN [40] as the foundation of our encoding network due its ability to extract fine detail from natural data and its representation power for spatial and temporal derivatives. A SIREN block is composed of a single linear layer followed by a weighted ($W$) sinusoidal activation function ($\sin(Wa)$). For each of our contextual features, $a$, we pass the parameterized inputs through an encoding network made up of three SIREN blocks, each with a hidden dimension of 64 (128 for loc+time), followed by a final linear layer which produces a 64-dimensional embedding. When fusing time with location, we replicate the spatial dimensions to match.

**Fusing with Visual Features** Similar to a traditional positional encoding, we merge the GTPE output with the visual features in the transformer stages of our multi-modal fusion architecture. At each stage, we linearly reproject the GTPE output to match the stage-specific embedding size. This is followed by bilinear

interpolation to scale the spatial dimensions to match the spatial resolution at that stage. Finally, we treat the output as a positional encoding and add it to the corresponding visual tokens to infuse the network with geo-temporal context. While our primary task is mobility analysis, GTPE is applicable to other problem domains (see supplemental material).

### 3.3 Probabilistic Traffic Speed Estimation

We propose a probabilistic approach for traffic speed estimation that naturally models variations in likely traffic speeds, along with an objective function that explicitly accounts for the number of traffic observations (i.e., a form of confidence in the underlying traffic speed averages for a given road segment). For this we use the Student's t-distribution, a symmetric distribution similar to the normal distribution that is used when estimating the mean of a normally distributed population in scenarios where: 1) there are a small number of observations and 2) the standard deviation of the population is unknown. In other words, a Student's t-distribution relates the distribution of the sample mean to the true mean.

For our purposes, we use the generalized form of the Student's t-distribution denoted as $t_\nu(\mu, \sigma)$, where $\mu$ is a location (shift) parameter, $\sigma > 0$ is a scale parameter, and $\nu > 0$ is a shape parameter (often referred to as degrees of freedom). The probability density function for this form of the Student's t-distribution is given by:

$$p(x|\nu, \mu, \sigma) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\nu\pi}\sigma} \left(1 + \frac{1}{\nu}\left[\frac{x-\mu}{\sigma}\right]^2\right)^{-\frac{\nu+1}{2}}, \tag{2}$$

where $\Gamma$ is the gamma function,

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}\,dx. \tag{3}$$

Instead of regressing traffic speeds directly, we capture uncertainty by estimating per-pixel prior distributions over traffic speeds. Given an overhead image and geo-temporal context as input, the output of the traffic speed decoder is per-pixel estimates of the shift-scale parameters of the prior distribution (i.e., $\mu$ and $\sigma^2$). A softplus activation is used to ensure these outputs are positive. During model training, we combine the estimated shift-scale parameters with the true count of traffic observations for each road segment, as the shape parameter $\nu$, to form a Student's t distribution.

To optimize our approach, we minimize the negative log likelihood of the resulting distributions, treating ground-truth traffic speeds on a given road segment as samples from the corresponding Student's t-distribution. Specifically, given an observed traffic speed, $y$, we compute the likelihood under the distribution (2). Our objective function then becomes:

$$\mathcal{L}_{speed} = -\log G(S(l), l, t; \Theta)(y), \tag{4}$$

where $G$ represents our proposed approach which takes as input an overhead image $S$, geolocation $l$, and time $t$, and outputs prior distributions over traffic speeds, and $\Theta$ are the weights of the network, which we optimize. At inference, we use the estimated shift parameter of our output distributions, $\mu$, as our prediction for traffic speed.

*Region Aggregation* We assume ground-truth traffic speeds are provided as averages over road segments (i.e., in aggregate form). However, our network generates dense, full-resolution predictions. We use a variant of the region aggregation layer [22] to facilitate comparison to the ground-truth values. This process averages predictions across a given road segment to produce a single aggregated estimate. In our case, we aggregate the estimated shift and scale parameters for each road segment before forming a per-road-segment Student's t-distribution.

### 3.4 Auxiliary Tasks

For the auxiliary tasks of road segmentation and orientation estimation, we create a task-specific decoder as described in Section 3.1. For road segmentation, we follow recent work and formulate this as a binary segmentation task. As our objective function, we use a combined loss that incorporates binary cross entropy and the Dice loss [52] ($\mathcal{L}_{bce} + (1 - \mathcal{L}_{dice})$). For orientation estimation, we treat this as a classification task over $K$ angular bins, and use standard cross entropy as the objective function. The cumulative objective function for primary and auxiliary tasks becomes:

$$\mathcal{L} = \mathcal{L}_{speed} + \mathcal{L}_{road} + \mathcal{L}_{orientation}. \tag{5}$$

### 3.5 Implementation Details

Our methods are implemented using PyTorch [37] and PyTorch Lightning [13] and optimized using Adam [25] ($\lambda = 1e^{-4}$). We simultaneously optimize the entire network for all tasks (loss terms weighted equally) and train for 50 epochs. Model selection is performed using a validation set. For each transformer stage, we use eight attention heads in all instances of MHSA. The expansion rate for each inverted residual block is 4. The expansion/shrink rate for the squeeze-and-excitation layers is 0.25. For road segmentation, we buffer road geometries assuming a two meter half width. For orientation estimation, we use $K = 16$ angular bins. We train on full size images and traffic speeds are represented in kilometers per hour.

## 4 Experiments

We evaluate our approach for the task of traffic speed estimation through a variety of experiments.

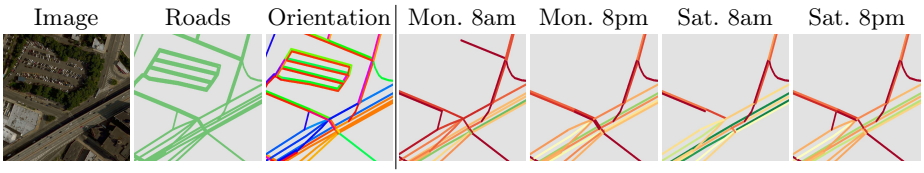


**Fig. 4:** Example images from the Dynamic Traffic Speeds (DTS) dataset and corresponding labels. The right four labels depict available historical traffic speeds at different times, where green (red) corresponds to faster (slower) speeds.

### 4.1 Dynamic Traffic Speeds Dataset

We train and evaluate our method using the recently introduced Dynamic Traffic Speeds (DTS) dataset [45], a fine-grained road understanding benchmark. DTS relates overhead imagery and road metadata with a year of historical traffic speeds for New York City, NY. The traffic speed data originates from Uber Movement Speeds [1], a collection of aggregated speed data at the road segment level, with hourly frequency, from Uber rideshare trips. The dataset contains 11,902 non-overlapping overhead images ($1024 \times 1024$) at approximately 0.3 meters per-pixel, with associated road segment information, and corresponding historical traffic data. Figure 4 visualizes example data from DTS. We use the original data splits, consisting of 85% training, 5% validation, and 10% testing. Similar to [45], we dynamically generate speed masks during training.

*Evaluation Metrics* We report three metrics for traffic speed (km/h) estimation: root-mean-square error (RMSE), mean absolute error (MAE), and the coefficient of determination ($R^2$), a proportion which describes how well variations in the observed values can be explained by the model. When computing evaluation metrics, we apply region aggregation to average predictions along each road segment to enable comparison to the ground truth.

### 4.2 Traffic Speed Estimation

As each test image is associated with time-varying historical traffic data, we consider two strategies for selecting a time for evaluation. For our first experiment, we sample a time for each test image from the set of observed traffic data across all contained road segments, and use this time to generate a ground-truth speed mask. We refer to this as a micro strategy, as metrics are computed globally across time. Table 1 shows the results of this study. Across all metrics, our method significantly outperforms the prior state-of-the-art.

For the second experiment, we employ a macro strategy that considers a specific set of times (Monday & Saturday, with hours 12am, 4am, 8am, 12pm, 5pm, and 8pm). For each time, we select all images in the test set that contain a road segment with observed traffic speed data at that time. Metrics are then averaged across time such that all times are weighted equally. Table 2 shows the

results of this study for a subset of times, with the overall average performance shown in the bottom row. As before, our method significantly outperforms prior work independent of the day of the week or hour of day.

Figure 5 shows qualitative results from our method. The top row shows ground-truth traffic speeds, which are provided as aggregates across road segments. The middle row shows the results of our method, after applying region aggregation to individual road segments. The bottom row shows the dense output of our approach, without region aggregation (per-pixel traffic speeds). As observed, our approach is able to capture nuances of traffic flow. For example, that roundabouts typically have faster traffic in straightaways but slower traffic on on-ramps and exits.

**Table 1:** Micro evaluation of traffic speed estimation.

| Method | Loss | RMSE ↓ | MAE ↓ | $R^2$ ↑ |
|---|---|---|---|---|
| [45] | Pseudo-Huber | 10.64 | 8.09 | 0.46 |
| Ours | Pseudo-Huber | 10.02 | 7.34 | 0.52 |
| Ours | Student's t | **8.84** | **6.64** | **0.63** |

**Table 2:** Macro evaluation of traffic speed estimation.

| | Workman et al. [45] | | | Ours | | |
|---|---|---|---|---|---|---|
| | RMSE | MAE | $R^2$ | RMSE | MAE | $R^2$ |
| Mon (4am) | 13.20 | 9.74 | 0.63 | 10.95 | 7.85 | 0.75 |
| Mon (12pm) | 10.40 | 7.86 | 0.52 | 8.78 | 6.56 | 0.66 |
| Sat (5pm) | 10.35 | 7.96 | 0.43 | 8.68 | 6.60 | 0.60 |
| Sat (8pm) | 10.26 | 7.78 | 0.46 | 8.47 | 6.34 | 0.63 |
| Overall | 11.12 | 8.34 | 0.49 | **9.15** | **6.76** | **0.65** |

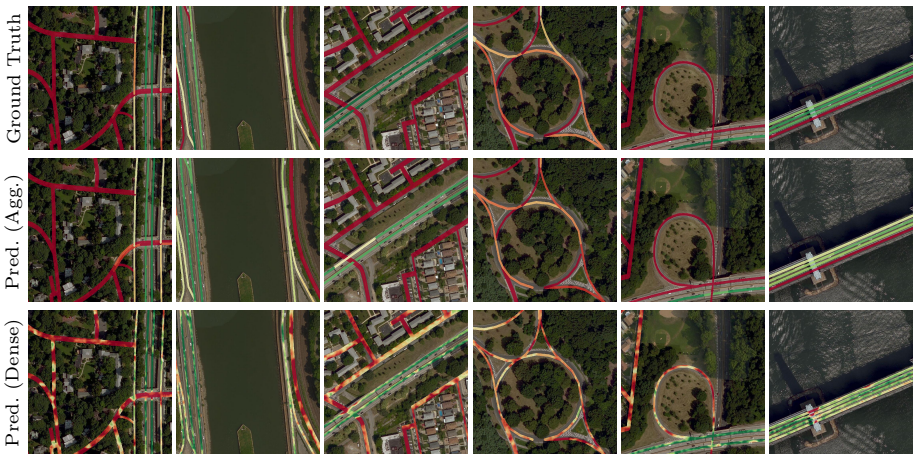


**Fig. 5:** Qualitative results for traffic speed estimation (Monday, 8am). (top) Ground-truth traffic speeds are provided as aggregates for individual road segments. For visualization, we replicate the ground-truth speed across the entire road segment. (middle) Predictions from our approach, after applying region aggregation. (bottom) Our results without region aggregation capture nuances of traffic flow, such as slowing down around curves. Green (red) corresponds to fast (slow) traffic speeds.

**Table 3:** Ablation study evaluating the impact of location and time context.

| Method | Context | RMSE | MAE | $R^2$ |
|---|---|---|---|---|
| [45] | *loc, time* | 13.13 | 9.60 | 0.18 |
| | *image* | 11.33 | 8.68 | 0.39 |
| | *image, loc* | 10.94 | 8.43 | 0.43 |
| | *image, time* | 10.67 | 8.11 | 0.46 |
| | *image, loc, time* | 10.64 | 8.09 | 0.46 |
| Ours | *loc, time* | 12.16 | 9.20 | 0.30 |
| | *image* | 10.28 | 7.70 | 0.50 |
| | *image, loc* | 9.65 | 7.26 | 0.56 |
| | *image, time* | 9.10 | 6.75 | 0.61 |
| | *image, loc, time* | 8.84 | 6.64 | 0.63 |

### 4.3 Ablation Study

We conduct an extensive ablation study to validate components of our approach. First, we consider the choice of objective function. Table 1 shows the results of this experiment. We compare our probabilistic approach (Student's t) versus a variant of our method which directly regresses traffic speeds using the Pseudo-Huber loss. As observed, our probabilistic approach significantly improves performance relative to this baseline. The choice of baseline objective function is consistent with that used in the prior state-of-the-art [45], highlighting that our probabilistic formulation directly leads to an increase in performance.

Next, we evaluate the impact of geo-temporal context on our traffic speed predictions. Table 3 shows the results of this experiment. First, an image-only variant of our approach outperforms a metadata-only variant (*loc, time*), demonstrating the importance of visual features for this task. Further, adding both location and time context leads to additional performance gains, with time context being superior to location context for predicting time-varying traffic speeds. Finally, like-for-like comparisons with prior work show the superior performance of our approach in each setting.

We also perform an ablation study comparing strategies for integrating geo-temporal context. The results are shown in Table 4. For this experiment, we vary

**Table 4:** Ablation study comparing different strategies for integrating geo-temporal context.

| Loc. Rep. | Context Enc. | | Context Fusion | | RMSE | MAE | $R^2$ |
|---|---|---|---|---|---|---|---|
| | *loc* | *time* | *loc* | *time* | | | |
| Center | MLP | MLP | Concat | Concat | 10.64 | 8.09 | 0.46 |
| Center | MLP | MLP | Token | Token | 9.13 | 6.87 | 0.61 |
| Dense | CNN | MLP | PE | Token | 9.02 | 6.71 | 0.61 |
| Dense | GTPE | GTPE | PE | PE | 8.84 | 6.64 | 0.63 |

the location representation, context encoding, and context fusion method. The results show that using a per-pixel representation (Dense) for location is superior to the center coordinate (Center) as in prior work [45]. Additionally, encoding location and time context using GTPE leads to better performance as compared to an MLP or CNN encoder. For fusing context features with visual features, treating location and time as a positional encoding (PE), as in GTPE, performs better than channel-wise concatenating (Concat) these features in the decoder, or adding context features as an additional token (Token) in each transformer stage (e.g., [11]). In summary, GTPE outperforms other strategies for integrating geo-temporal context, which are representative of the prior state-of-the-art.

Finally, we compare our approach, which includes the auxiliary tasks of road segmentation and orientation estimation, to variants of our approach that consider subsets of the tasks (Table 5). Results show that including the auxiliary tasks has a positive impact on performance. This matches recent work in multi-task learning that finds sharing information across tasks can lead to performance improvements when the tasks are synergistic [8].

**Table 5:** Ablation study on the impact of multi-task learning.

| Road | Orientation | RMSE | MAE | $R^2$ |
|---|---|---|---|---|
| ✗ | ✗ | 9.18 | 6.86 | 0.60 |
| ✓ | ✗ | 9.03 | 6.70 | 0.61 |
| ✗ | ✓ | 8.92 | 6.67 | 0.62 |
| ✓ | ✓ | 8.84 | 6.64 | 0.63 |

### 4.4 DTS++: A Dataset for Location Adaptation

Location adaptation, an analog to domain adaptation, aims to address the problem of adapting a model trained on one region to another region. To support mobility-related location adaptation experiments, we introduce the Dynamic Traffic Speeds++ (DTS++) dataset, an extension of DTS to include a new city, Cincinnati, OH. DTS++ is constructed in a similar manner to DTS and includes one year of historical traffic data collected from Uber Movement Speeds [1]. This mobility data is paired with 11,137 overhead images ($1024 \times 1024$) at approximately 0.3 meters per-pixel.

Using DTS++, we conducted an initial experiment to evaluate how our approach, trained on New York City (NYC), performs when adapted to Cincinnati (Cincy). We show the results of this experiment in Table 6. As expected, performance deteriorates as Cincy has vastly different spatiotemporal mobility patterns than NYC. For example, the average speed across all road segments in DTS++ for

**Table 6:** An example of location adaptation using DTS++. When adapting our approach from New York City (NYC) to Cincinnati (Cincy), fine-tuning the geo-temporal positional encoding (GTPE) module dramatically improves performance.

| Train | Test | GTPE | RMSE | MAE | $R^2$ |
|---|---|---|---|---|---|
| NYC | Cincy | *original* | 24.06 | 18.84 | 0.04 |
| NYC | Cincy | *fine-tuned* | 12.29 | 9.18 | 0.75 |

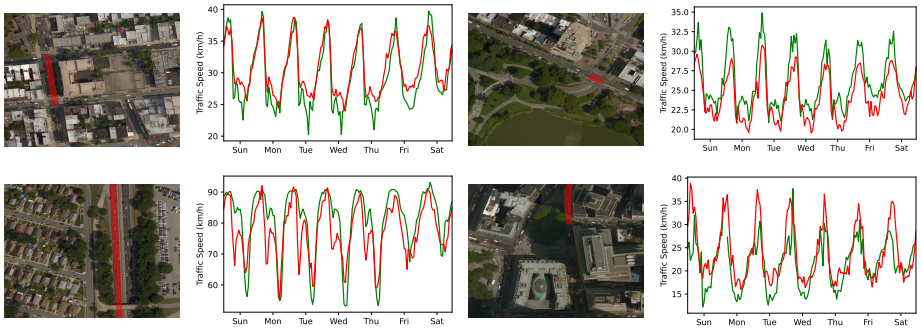


**Fig. 6:** Visualizing traffic speed predictions for individual road segments versus time. (left) A road segment shown in red. (right) Predictions from our approach (red) versus historical traffic data (green).

NYC is 18.93 km/h while Cincy is 31.51 km/h. In addition, we show that fine-tuning our geo-temporal positional encoding module (GTPE) on Cincy dramatically improves performance. This experiment simultaneously highlights the impact of GTPE on traffic speed estimation and shows that fine-tuning only the associated $\sim$ 65k parameters can be an efficient way of adapting models to new locations.

In summary, while location adaptation is not the primary focus of this work, it is a nascent and important research direction and our hope is that DTS++ helps enable future studies related to mobility-related location adaptation.

## 5 Applications

*Creating Dense City-Scale Traffic Models* Though empirical traffic data has become increasingly available, not all roads are traversed at all times. This presents challenges for downstream applications which rely on data from transport modeling, as they are limited to regions/times where data is available, or must collect their own data. Our approach presents an alternative as it can be used to create dense city-scale traffic models. Figure 1 (left) shows available historical traffic data from the Dynamic Traffic Speeds dataset (Brooklyn, Monday 8am). As observed, many roads are missing empirical speed data (white) as they were not traversed at the given time. Figure 1 (right) shows how our approach can be used to generate a dense model of traffic flow due to its ability to generalize across space and time.

*Modeling Traffic Flow* Our approach can be used to model spatiotemporal traffic patterns for individual road segments. To demonstrate this, we analyze how predictions from our approach on unseen road segments compare to historical traffic data. Figure 6 shows the results of this experiment. The x-axis represents time, with each day representing a 24 hour interval. Our approach (red) is able to capture temporal trends in historical traffic data (green).

**Fig. 7:** The output of our approach on unseen locations visualized as a local motion model. We sample from locations predicted as road and visualize corresponding orientation estimates as vectors, colored by the predicted speed. Green (red) corresponds to fast (slow) traffic speeds.

*Generalizing to Novel Locations* Our approach can be thought of as estimating a local motion model from an overhead image which describes how the environment is traversed. Figure 7 visualizes how our method can be applied to novel locations to generate a local motion model. This allows our approach to be applied to locations where transport data is incomplete or unavailable, a common occurrence. For example, road networks are not always accurate for a given location (e.g., unmapped or changed). Further, traffic speed data is sparse and not always available for all roads. The generalizability of our approach is made possible by the inclusion of the auxiliary tasks of road segmentation and orientation estimation, in addition to their performance benefit from multi-task learning.

## 6 Conclusion

Our goal was to understand city-scale mobility patterns using overhead imagery, a task we refer to as image-driven traffic modeling. We proposed a multi-modal, multi-task transformer-based segmentation architecture and showed how it can be used to create dense city-scale traffic models. Our method has several key components, including a probabilistic formulation for naturally modeling variations in traffic speeds, and a geo-temporal positional encoding module for integrating geo-temporal context. Extensive experiments using the Dynamic Traffic Speeds (DTS) dataset demonstrate how our approach significantly improves the state-of-art in traffic speed estimation. Finally, we introduced the DTS++ dataset to support motion-related location adaptation studies across diverse cities. Our hope is that these results continue to demonstrate the real-world utility of image-driven traffic modeling.

## References


1. Uber Movement: Speeds calculation methodology. Tech. rep., Uber Technologies, Inc. (2019)
2. Albert, A., Kaur, J., Gonzalez, M.C.: Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017)
3. Bickel, P.J., Chen, C., Kwon, J., Rice, J., Van Zwet, E., Varaiya, P.: Measuring traffic. Statistical Science pp. 581–597 (2007)
4. Chakrabarti, S., Kushari, T., Mazumder, T.: Does transportation network centrality determine housing price? Journal of Transport Geography **103**, 103397 (2022)
5. Chao, Q., Bi, H., Li, W., Mao, T., Wang, Z., Lin, M.C., Deng, Z.: A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving. Computer Graphics Forum **39**(1), 287–308 (2020)
6. Chen, C., Ma, J., Susilo, Y., Liu, Y., Wang, M.: The promises of big data and small data for travel behavior (aka human mobility) analysis. Transportation Research Part C: Emerging Technologies **68**, 285–299 (2016)
7. Chen, L., Grimstead, I., Bell, D., Karanka, J., Dimond, L., James, P., Smith, L., Edwardes, A.: Estimating vehicle and pedestrian activity from town and city traffic cameras. Sensors **21**(13), 4564 (2021)
8. Crawshaw, M.: Multi-task learning with deep neural networks: A survey. arXiv preprint arXiv:2009.09796 (2020)
9. Cui, H., Radosavljevic, V., Chou, F.C., Lin, T.H., Nguyen, T., Huang, T.K., Schneider, J., Djuric, N.: Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In: International Conference on Robotics and Automation (2019)
10. Dai, Z., Liu, H., Le, Q.V., Tan, M.: CoAtNet: Marrying convolution and attention for all data sizes. In: Advances in Neural Information Processing Systems (2021)
11. Diao, Q., Jiang, Y., Wen, B., Sun, J., Yuan, Z.: Metaformer: A unified meta framework for fine-grained recognition. arXiv preprint arXiv:2203.02751 (2022)
12. Eicher, L., Mommert, M., Borth, D.: Traffic noise estimation from satellite imagery with deep learning. In: International Geoscience and Remote Sensing Symposium (2022)
13. Falcon, W., The PyTorch Lightning team: PyTorch Lightning (2019), https://github.com/Lightning-AI/lightning
14. Feng, Z., Guo, S., Tan, X., Xu, K., Wang, M., Ma, L.: Rethinking efficient lane detection via curve modeling. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)
15. Fernández Llorca, D., Hernández Martínez, A., García Daza, I.: Vision-based vehicle speed estimation: A survey. IET Intelligent Transport Systems **15**(8), 987–1005 (2021)
16. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)
17. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: AAAI Conference on Artificial Intelligence (2019)
18. Hadzic, A., Blanton, H., Song, W., Chen, M., Workman, S., Jacobs, N.: RasterNet: Modeling free-flow speed using lidar and overhead imagery. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2020)

19. Hamilton-Baillie, B., Jones, P.: Improving traffic behaviour and safety through urban design. In: Proceedings of the Institution of Civil Engineers: Civil Engineering. vol. 158, pp. 39–47 (2005)
20. Homayounfar, N., Ma, W.C., Liang, J., Wu, X., Fan, J., Urtasun, R.: DAGMapper: Learning to map by discovering lane topology. In: IEEE International Conference on Computer Vision (2019)
21. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
22. Jacobs, N., Kraft, A., Rafique, M.U., Sharma, R.D.: A weakly supervised approach for estimating spatial density functions from high-resolution satellite imagery. In: ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (2018)
23. Janai, J., Güney, F., Behl, A., Geiger, A., et al.: Computer vision for autonomous vehicles: Problems, datasets and state of the art. Foundations and Trends® in Computer Graphics and Vision **12**(1–3), 1–308 (2020)
24. Ke, G., He, D., Liu, T.Y.: Rethinking positional encoding in language pre-training. In: International Conference on Learning Representations (2020)
25. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2014)
26. Kumar, A., Kashiyama, T., Maeda, H., Omata, H., Sekimoto, Y.: Citywide reconstruction of traffic flow using the vehicle-mounted moving camera in the CARLA driving simulator. In: IEEE International Conference on Intelligent Transportation Systems (2022)
27. Li, M., Tong, P., Li, M., Jin, Z., Huang, J., Hua, X.S.: Traffic flow prediction with vehicle trajectories. In: AAAI Conference on Artificial Intelligence (2021)
28. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al.: Swin transformer v2: Scaling up capacity and resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)
29. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: IEEE Conference on Computer Vision and Pattern Recognition (2021)
30. Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., Johnson, B.A.: Deep learning in remote sensing applications: A meta-analysis and review. ISPRS Journal of Photogrammetry and Remote Sensing **152**, 166–177 (2019)
31. Mac Aodha, O., Cole, E., Perona, P.: Presence-only geographical priors for fine-grained image classification. In: IEEE International Conference on Computer Vision (2019)
32. Mahajan, V., Kuehnel, N., Intzevidou, A., Cantelmo, G., Moeckel, R., Antoniou, C.: Data to the people: a review of public and proprietary data for transport models. Transport Reviews **42**(4), 415–440 (2022)
33. Marshall, W.E., Piatkowski, D.P., Garrick, N.W.: Community design, street networks, and public health. Journal of Transport & Health **1**(4), 326–340 (2014)
34. Máttyus, G., Luo, W., Urtasun, R.: DeepRoadMapper: Extracting road topology from aerial images. In: IEEE International Conference on Computer Vision (2017)
35. Mukherjee, R., Rollend, D., Christie, G., Hadzic, A., Matson, S., Saksena, A., Hughes, M.: Towards indirect top-down road transport emissions estimation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2021)
36. Najjar, A., Kaneko, S., Miyanaga, Y.: Combining satellite imagery and open data to map road safety. In: AAAI Conference on Artificial Intelligence (2017)

37. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (2019)
38. Salem, T., Workman, S., Jacobs, N.: Learning a dynamic map of visual appearance. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
39. Salem, T., Zhai, M., Workman, S., Jacobs, N.: A multimodal approach to mapping soundscapes. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2018)
40. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. In: Advances in Neural Information Processing Systems (2020)
41. Song, W., Salem, T., Blanton, H., Jacobs, N.: Remote estimation of free-flow speeds. In: IEEE International Geoscience and Remote Sensing Symposium (2019)
42. Vargas-Munoz, J.E., Srivastava, S., Tuia, D., Falcao, A.X.: OpenStreetMap: Challenges and opportunities in machine learning and remote sensing. IEEE Geoscience and Remote Sensing Magazine **9**(1), 184–199 (2020)
43. Wang, C., Quddus, M., Ison, S.: A spatio-temporal analysis of the impact of congestion on traffic safety on major roads in the UK. Transportmetrica A: Transport Science **9**(2), 124–148 (2013)
44. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: PVT v2: Improved baselines with pyramid vision transformer. Computational Visual Media (2022)
45. Workman, S., Jacobs, N.: Dynamic traffic modeling from overhead imagery. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
46. Workman, S., Rafique, M.U., Blanton, H., Jacobs, N.: Revisiting near/remote sensing with geospatial attention. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)
47. Workman, S., Souvenir, R., Jacobs, N.: Understanding and mapping natural beauty. In: IEEE International Conference on Computer Vision (2017)
48. Workman, S., Zhai, M., Crandall, D.J., Jacobs, N.: A unified model for near and remote sensing. In: IEEE International Conference on Computer Vision (2017)
49. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and efficient design for semantic segmentation with transformers. In: Advances in Neural Information Processing Systems (2021)
50. Yang, W., Li, Q., Liu, W., Yu, Y., Ma, Y., He, S., Pan, J.: Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation. In: IEEE Conference on Computer Vision and Pattern Recognition (2021)
51. Zhang, Q., Chang, J., Meng, G., Xiang, S., Pan, C.: Spatio-temporal graph structure learning for traffic forecasting. In: AAAI Conference on Artificial Intelligence (2020)
52. Zhou, L., Zhang, C., Wu, M.: D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2018)