

DETERMINING A ROUTE BETWEEN AN ORIGIN AND A DESTINATION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to Singapore Patent Application No. 10202000534V filed on 20 January 2020 and entitled "Spatial Transition Learning With Deep Probabilistic Model," the content of which is hereby incorporated by reference in its entirety.

TECHNICAL FIELD

[0002] The present disclosure generally relates to determining a route between an origin and a destination, more specifically using neural network representations for the route determination.

BACKGROUND

[0003] Route-planning applications can be utilized to provide users with directions between different locations. In an example, a user can provide a route planning application with a beginning point of travel and an end point of travel (e.g., beginning and ending addresses). The route planning application can include or utilize representations of roads and intersections and one or more existing route-prediction methods can output a most likely route of travel.

[0004] Some existing route-prediction methods may use probabilistic models that are trained with sampling-based methods that suffer from slow convergence and are limited to relatively small datasets. Therefore, many existing route-prediction methods may not be able to efficiently scale with large datasets.

[0005] The route of travel can have a sequential property (e.g., since the route of travel may be expressed as a sequence of roads). Some existing route-prediction methods capture the sequential property by modeling the spatial transition patterns using a Hidden Markov Model (HMM), which requires explicit dependency assumptions to make the inferences tractable. However, in many real-world scenarios, the explicit dependency assumptions cannot account for long-range dependencies. Therefore, many existing route-prediction methods may make an erroneous or inaccurate prediction of the route of travel because the sequential property of the route of travel is not fully considered.

[0006] The route of travel can be impacted by the destination, and some existing route-prediction methods consider the impact of destination by treating each destination

separately. However, by treating each destination separately, existing route-prediction methods fail to share statistical strength across trips with nearby destinations. Existing route-prediction methods may also assume that accurate representations of the destinations are available (e.g., exact ending streets of the trips) and may use the corresponding road segments to predict the route of travel. However, in many real-world scenarios, the destinations may only be rough or approximate destination representations, and a driver may not end a trip on the exact street as the user requested (e.g., in a taxi dispatch system). Therefore, many existing route-prediction methods may make an erroneous or inaccurate prediction of the route of travel because deviations or inaccuracies in the destinations' representations are not fully considered.

[0007] The route of travel can be impacted by traffic, and some existing route-prediction methods consider the impact of traffic by assuming that traffic conditions in the same time slot (e.g., 7:00 am to 8:00 am every weekday) are temporally invariant. However, in many real-world scenarios, the temporally invariant assumptions about traffic conditions do not reflect real-time traffic. Therefore, many existing route-prediction methods may make an erroneous or inaccurate prediction of the route of travel because real-time traffic data are not fully considered.

[0008] Therefore, there exists a need for a method that can determine (e.g., predict) the most likely route between a given origin and a given destination in a manner that can scale with large datasets and that fully, and jointly, considers the sequential property of the route of travel, any deviations or inaccuracies in the destinations' representations, and real-time traffic.

SUMMARY

[0009] According to a first aspect of the present disclosure, a method for determining a route is provided. The method may include: obtaining input data from a user, the input data including an indication of an origin and a destination for a trip in a road network; obtaining real-time traffic data from a plurality of probe vehicles spatially distributed across the road network; and obtaining a past traveled route from at least one of the plurality of probe vehicles, each probe vehicle having a respective past traveled route, the past traveled route indicating a sequence of road segments traveled in the trip. The method may further include: determining, using a first neural network, a representation of the past traveled route; determining, using a second neural network, a representation

of the real-time traffic data; determining, using an adjoint generative process, a representation of the destination; and determining a next road segment for the trip based on the representation of the past traveled route, the representation of the real-time traffic data, and the representation of the destination.

[0010] According to a second aspect of the present disclosure, a system for determining a route is provided. The system may include a memory and at least one processor communicatively coupled to the memory and configured to perform operations. The operations may include: obtaining input data from a user, the input data including an indication of an origin and a destination for a trip in a road network; obtaining real-time traffic data from a plurality of probe vehicles spatially distributed across the road network; and obtaining a past traveled route from at least one of the plurality of probe vehicles, each probe vehicle having a respective past traveled route, the past traveled route indicating a sequence of road segments traveled in the trip. The operations may further include: determining, using a first neural network, a representation of the past traveled route; determining, using a second neural network, a representation of the real-time traffic data; determining, using an adjoint generative process, a representation of the destination; and determining a next road segment for the trip based on the representation of the past traveled route, the representation of the real-time traffic data, and the representation of the destination.

[0011] According to a third aspect of the present disclosure a non-transitory computer-readable medium is provided. The non-transitory computer-readable medium may include instructions that are operable, when executed by a data processing apparatus, to perform operations. The operations may include: obtaining input data from a user, the input data including an indication of an origin and a destination for a trip in a road network; obtaining real-time traffic data from a plurality of probe vehicles spatially distributed across the road network; and obtaining a past traveled route from at least one of the plurality of probe vehicles, each probe vehicle having a respective past traveled route, the past traveled route indicating a sequence of road segments traveled in the trip. The operations may further include: determining, using a first neural network, a representation of the past traveled route; determining, using a second neural network, a representation of the real-time traffic data; determining, using an adjoint generative process, a representation of the destination; and determining a next road segment for the

trip based on the representation of the past traveled route, the representation of the real-time traffic data, and the representation of the destination.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] FIG. 1A is a diagram showing an example road network, according to an implementation of the present disclosure.

[0013] FIG. 1B is a diagram illustrating the task of route recovery in an example spatial region having a sparse trajectory, according to an implementation of the present disclosure.

[0014] FIG. 2 is a diagram showing a graphical model of an example generative process, where a route $\mathbf{r}$ between an origin and a destination is determined, according to an implementation of the present disclosure.

[0015] FIG. 3 is a diagram showing a graphical model of an example generative process, including the generative process shown in FIG. 2 and an example adjoint generative process that simultaneously learns multiple destination representations and adaptively allocates trips into one of these destination representations, according to an implementation of the present disclosure.

[0016] FIG. 4 is a diagram showing an example inference framework of an example deep probabilistic spatial transition method, according to an implementation of the present disclosure.

[0017] FIG. 5A is a diagram showing an example spatial distribution of GPS points in a first example road segment space, according to an implementation of the present disclosure.

[0018] FIG. 5B is a diagram showing an example spatial distribution of GPS points in a second example road segment space, according to an implementation of the present disclosure.

[0019] FIG. 6A is a diagram showing a distribution of travel distances for a first dataset, according to an implementation of the present disclosure.

[0020] FIG. 6B is a diagram showing a distribution of the number of road segments used in a trip for the first dataset, according to an implementation of the present disclosure.

[0021] FIG. 7A is a diagram showing a distribution of travel distances for a second dataset, according to an implementation of the present disclosure.

[0022] FIG. 7B is a diagram showing a distribution of the number of road segments used in a trip for the second dataset, according to an implementation of the present disclosure.

[0023] FIG. 8A is a diagram showing a plot illustrating the accuracy of an example deep probabilistic spatial transition method relative to other methods for the first dataset, according to an implementation of the present disclosure.

[0024] FIG. 8B is a diagram showing a plot illustrating the accuracy of an example deep probabilistic spatial transition method relative to other methods for the second dataset, according to an implementation of the present disclosure.

[0025] FIG. 9 is a diagram showing a plot illustrating an example route recovery accuracy of the STRS and STRS+ methods relative to sampling rates for the second dataset, according to an implementation of the present disclosure.

[0026] FIG. 10 is a diagram showing a plot illustrating an example variation of training time with training data size, according to an implementation of the present disclosure.

[0027] FIG. 11 is a diagram showing an example system for determining a route between an origin and a destination, according to an implementation of the present disclosure.

[0028] FIG. 12 is a block diagram showing an example computing platform, according to an implementation of the present disclosure.

[0029] FIG. 13 is a flowchart showing an example process for determining a route between an origin and a destination, according to an implementation of the present disclosure.

DETAILED DESCRIPTION

[0030] In some aspects of what is described here, spatial transition patterns of real-world trips are modeled on a road network. For example, given an origin and a destination, various aspects of the present disclosure present a Deep Probabilistic Spatial Transition method (abbreviated as DeepST in this disclosure for the sake of simplicity) that determines (e.g., generates or predicts) the most likely traveling route on the road network and generates a probability value to indicate the likelihood of the route being traveled. The proposed DeepST method can be used in a variety of applications where spatial transition patterns can be used, examples being a taxi ridesharing service, route recovery from sparse trajectories, and recommendation of popular routes in a road

network. As an example, in a taxi dispatch system, the origin and destination may be stipulated before the start of a trip; thus, predicting the most likely route traveled from the origin to the destination can improve the efficiency and profits of the taxi dispatch service by identifying potential passengers that may be located on or nearby the most likely traveled route.

[0031] FIG. 1A is a diagram showing an example road network 100, according to an implementation of the present disclosure. The example road network 100 includes geolocation measurements (e.g., GPS points) 102, 104, 106, 108, 110, 112, 114, 116, 118 spatially distributed across various locations in the road network 100. In some instances, the GPS points may be provided by probe vehicles distributed across the road network 100, and the GPS points may be described by the latitude and longitude of the respective locations. In some implementations, some of the GPS points may serve as trip origins and trip destinations. For example, in the road network 100 seen in FIG. 1A, GPS points 102, 104, 106 are denoted as destinations A, B, C, respectively. The road network 100 may include road segments $r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}$ that connect one GPS point to an adjacent (e.g., immediately adjacent) GPS point. For example, the road segment r_1 may be a segment of road in the road network 100 that connect GPS point 108 with GPS point 112. Similarly, road segment r_2 may be a segment of road that connects GPS point 112 with GPS point 114. Assuming GPS points 108 and 110 are trip origins, example trips (e.g., expressed as a sequence of road segments) that can be made from GPS point 110 to the destination C and their associated frequency of use are shown below in Table 1. Table 1 also shows example trips that can be made from GPS point 108 to destinations A and B and their associated frequency of use. In the example shown in Table 1, the frequency of use may refer to the number of times a particular route has been traveled based on historical use of the road network 100.

Table 1: Example Trips on a Road Network.

Origin	Route	Destination	Frequency
GPS point 110	$r_3 \rightarrow r_2 \rightarrow r_4$	C	400
GPS point 110	$r_3 \rightarrow r_2 \rightarrow r_5 \rightarrow r_{10}$	C	100
GPS point 108	$r_1 \rightarrow r_2 \rightarrow r_5 \rightarrow r_6$	A	100
GPS point 108	$r_1 \rightarrow r_2 \rightarrow r_7 \rightarrow r_8$	B	100
GPS point 108	$r_1 \rightarrow r_2 \rightarrow r_5 \rightarrow r_6 \rightarrow r_9$	B	100

[0032] As seen in Table 1, the route for a particular trip (e.g., between a given origin and a given destination) can be modeled by spatial transitions of road segments (which may be referred to as transition patterns in this disclosure). Furthermore, the historical transition patterns in the road network 100 can be highly skewed since some routes within the road network 100 may be more likely to be traveled than others. To reliably model such transition patterns, implementations of the present disclosure jointly account for at least the following factors: (1) spatial transition patterns demonstrate a strong sequential property; (2) destinations have a global impact on the transition; and (3) route choices are influenced by real-time traffic conditions.

[0033] To illustrate how spatial transition patterns demonstrate a strong sequential property, suppose the next transition of a vehicle driving on road segment r_2 needs to be predicted. Then, based on the statistical information shown in Table 1 (e.g., the historical trips made in the road network 100), $\mathbb{P}(r_4|r_2) = 4/8$ is greater than $\mathbb{P}(r_5|r_2) = 3/8$ and $\mathbb{P}(r_7|r_2) = 1/8$. Therefore, there can be a high confidence to predict that the vehicle will transit to road segment r_4 from road segment r_2 . However, if there is additional information about the sequence of road segments already traveled in the trip (e.g., long-range dependencies), such as knowledge that the traveled road sequence is $r_1 \rightarrow r_2$, then the prediction can favor road segment r_5 over road segment r_4 , since $\mathbb{P}(r_4|r_1 \rightarrow r_2) = 0$, $\mathbb{P}(r_5|r_1 \rightarrow r_2) = 2/3$, and $\mathbb{P}(r_7|r_1 \rightarrow r_2) = 1/3$.

[0034] To illustrate how destinations have a global impact on the spatial transition patterns, suppose a vehicle is driving on road segment r_5 and the trip's destination is destination C. Then, based on the statistical information shown in Table 1 (e.g., the historical trips made in the road network 100), there is a higher probability of transiting to road segment r_6 than transiting to road segment r_{10} , since $\mathbb{P}(r_6|r_5) = 2/3 > \mathbb{P}(r_{10}|r_5) = 1/3$. However, if there is additional information, such as knowledge that the trip's destination is destination C, then the prediction can favor road segment r_{10} over road segment r_6 , since $\mathbb{P}(r_{10}|r_5, C) = 1$ and $\mathbb{P}(r_6|r_5, C) = 0$.

[0035] To illustrate how route choices can be influenced by real-time traffic conditions, it may be assumed in some instances that vehicle drivers tend to choose those less congested routes rather than the shortest one. Suppose a vehicle is on road segment r_2 and the trip's destination is destination B, then based on the statistical information shown in Table 1 (e.g., the historical trips made in the road network 100), there is an equal probability of transiting to road segment r_5 or road segment r_7 , since $\mathbb{P}(r_5|r_2, B) =$

$\mathbb{P}(r_7|r_2, B) = 1/2$. However, if there is additional information, such as knowledge that traffic on road segments r_7, r_8 is more congested than that on road segments r_5, r_6, r_9 , the driver may be more likely to choose road segment r_5 instead of road segment r_7 even if the route $r_7 \rightarrow r_8$ is shorter.

[0036] According to various aspects of the present disclosure, the proposed DeepST method unifies all the above-mentioned factors (strong sequential property of spatial transitions, the global impact of destination on the transitions, and the influence of real-time traffic conditions) into a single generative model that determines (e.g., learns) the spatial transition patterns on a road network. In some instances, aspects of the systems and techniques that utilize the proposed DeepST method provide technical improvements and advantages over existing approaches.

[0037] The proposed DeepST method determines the most likely route between an origin and a destination by conditioning the determination on a past traveled route (e.g., a sequence of road segments already traveled in the trip), the representation of a destination, and the representation of real-time traffic. In some implementations, the sequence of road segments already traveled in the trip is processed (e.g., compressed) by one or more recurrent neural networks (RNNs) that accounts for long-range dependencies and that, unlike existing methods, does not make an explicit dependency assumption. In some implementations, to account for the impact of destination, instead of treating the destinations separately as in existing methods, the proposed DeepST method determines (e.g., learns) K -destination proxies based on an adjoint generative model. With the adjoint generative model, the destination proxies are determined jointly with the generation of routes, which, unlike existing methods, permits end-to-end training. With the adjoint generative model, the proposed DeepST method also enables effective sharing of statistical strength across trips and allows the proposed DeepST method to be robust to any inaccuracies in the destinations. In some implementations, to account for the influence of real-time traffic, the proposed DeepST method determines (e.g., learns) the real-time traffic representation based on a high-dimensional latent variable whose posterior distribution can then be inferred from the observations. To this end, various aspects of the present disclosure also propose an efficient inference method to learn the posterior distributions of the latent variables of the proposed DeepST method based on observations. According to various aspects of the present disclosure, the inference method can be developed within the Variational Auto-Encoders (VAEs) framework and

may be fully differentiable, thus enabling the proposed DeepST method to scale to large-scale datasets.

[0038] FIG. 1B is a diagram illustrating the task of route recovery in an example spatial region 101 having a sparse trajectory, according to an implementation of the present disclosure. The example spatial region 101 includes GPS points 122, 124, 126, 128, 130, which are also denoted in FIG. 1B as points a_1, a_2, a_3, a_4, a_5 , respectively. The GPS points 122, 124, 126, 128, 130 may be points in a sparse trajectory (e.g., a down-sampled spatial transition pattern). The example spatial region 101 also includes buildings 132, 134, 136, 138. As discussed above, route recovery can arise in real-world trajectories due to low-sampling rates or the turning off of location-acquisition devices (e.g., GPS devices). The task of route recovery attempts to infer the most likely route that was traversed between GPS points in the sparse trajectory. For example, route recovery may attempt to infer the most likely route between two road segments that are not adjacent to each other on the road network. In the illustration in FIG. 1B, there may be two possible routes $\mathbf{r}_A$ (denoted by trajectory 140) and $\mathbf{r}_B$ (denoted by trajectory 142) from GPS point a_3 to GPS point a_4 in an observed sparse trajectory $T_a = [a_1, a_2, \dots, a_5]$. For the task of route recovery, if GPS points a_3, a_4 are treated as the origin and destination, respectively, various aspects of the present disclosure score the likelihood of the two candidate routes $\mathbf{r}_A$ and $\mathbf{r}_B$ to infer the most likely route from origin GPS point a_3 to destination GPS point a_4 .

[0039] According to various aspects of the present disclosure, the proposed DeepST method can also be used as a spatial transition inference module of a sparse route recovery method. For example, in some implementations, the proposed DeepST method may be used to score the likelihood of a route being traveled, and the likelihood score can be used to recover the underlying route from sparse trajectories, thereby increasing the accuracy of the sparse route recovery method in inferring the most likely route that was traversed between GPS points in the sparse trajectory.

[0040] For a better understanding of the present disclosure and for ease of reference, the example notations used in present disclosure are shown below in Table 2.

Table 2: Notations.

Symbol	Definition
T	GPS Trajectory
$\mathbf{T}$	Trip

$\mathbf{r}$	Route
r_i	The i -th road segment in route $\mathbf{r}$
$\mathbf{x}$	Proxy destination coordinate
$\boldsymbol{\pi}$	Destination allocation indicator
$\mathbf{c} \in \mathbb{R}^{ \mathcal{C} }$	The real-time traffic representation
$\mathcal{C}$	Real-time traffic indicator

[0041] Referring to Table 2, in some instances, a road network can be represented as a directed graph $G(V, E)$, in which V, E represent the vertices (crossroads) and edges (road segments) respectively.

[0042] Referring to Table 2, in some instances, a route $\mathbf{r} = [r_i]_{i=1}^n$ can be a sequence of adjacent road segments, where $r_i \in E$ represents the i -th road segment in the route.

[0043] Referring to Table 2, in some instances, a GPS trajectory T can be a sequence of sample points $\langle p_i, \tau_i \rangle_{i=1}^{|T|}$ from the underlying route $\mathbf{r}$ of a moving object, where p_i, τ_i represent the i -th GPS location and timestamp, respectively.

[0044] Referring to Table 2, in some instances, a trip $\mathbf{T}$ can be a travel along a route $\mathbf{r}$ on the road network starting at time s . The notations $\mathbf{T}.\mathbf{r}$ and $\mathbf{T}.s$, respectively, can be used to denote the traveled route and starting time of trip $\mathbf{T}$.

[0045] Various implementations of the present disclosure can be applied to the following general scenario: Given a road network $G(V, E)$ and a historical trajectory dataset $\mathcal{D} = \{T^m\}_{m=1}^M$, as well as the starting time, origin, and destination of a trip $\mathbf{T}$, various implementations of the present disclosure determine (e.g., predict using spatial transition learning with deep probabilistic models) the most likely traveled route of the trip $\mathbf{T}$ and score the likelihood of any route being traveled by conditioning the determination of the most likely traveled route on a past traveled route (e.g., a sequence of road segments already traveled in the trip), the representation of a destination, and the representation of real-time traffic. To facilitate the determination of the most likely traveled route, it may be assumed that the initial road segment $\mathbf{T}.r_1$ of the trip $\mathbf{T}$ is given. However, for the destination, various implementations of the present disclosure assume that a rough or approximate coordinate $\mathbf{x}$ (also denoted as $\mathbf{T}.\mathbf{x}$) of the destination is available (e.g., expressed as a latitude and longitude).

[0046] For a better understanding of the present disclosure, the present disclosure is also separated into sections, and various concepts that are relevant to the various aspects of the present disclosure are now discussed.

Variational Inference and Variational Auto-Encoders

[0047] Probabilistic methods or Bayesian generative methods provide a way to explain the generation of observed data. Specifically, probabilistic or Bayesian generative methods incorporate prior knowledge regarding data into a model design by: (1) introducing appropriate latent variables $\mathbf{z}$ to serve as explanatory factors; and (2) describing the generation of observed data by specifying proper generative process based on domain knowledge.

[0048] Generally, the probabilistic methods can be formulated as follows: (1) draw a latent variable $\mathbf{z}$ from a prior distribution $p(\mathbf{z})$; (2) relate $\mathbf{z}$ to observation $\mathbf{x}$ through a conditional distribution $p(\mathbf{x}|\mathbf{z})$; and (3) infer the posterior distribution $p(\mathbf{z}|\mathbf{x})$, which can be used in a prediction stage.

[0049] One of the challenges in adopting probabilistic methods lies in inferring the posterior distribution $p(\mathbf{z}|\mathbf{x})$. By Bayes' theorem, the following holds:

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{z})p(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})} \quad (1)$$

The challenge in Equation (1) is the intractable computation of marginal distribution $p(\mathbf{x}) = \int p(\mathbf{z}, \mathbf{x}) d\mathbf{z}$ in a high-dimensional space, and existing inference methods (e.g., Markov Chain Monte Carlo (MCMC)) are usually only applicable to small datasets and simple models.

[0050] According to various aspects of the present disclosure, variational inference (VI) may be used in place of existing inference (e.g., MCMC) methods. Specifically, VI can turn the inference of the posterior distribution $p(\mathbf{z}|\mathbf{x})$ into an optimization problem. In comparison to MCMC in existing methods, VI is more computationally efficient by taking advantage of numerical optimization methods.

[0051] In VI, a family of probability densities $\mathcal{Q}$ is posited. Subsequently, a search is made for the optimal posterior approximation $q^*(\mathbf{z}) \in \mathcal{Q}$ that is closest to the posterior distribution $p(\mathbf{z}|\mathbf{x})$, where the closeness is measured by the Kullback-Leibler (KL)-

divergence. The search for the optimal posterior approximation $q^*(\mathbf{z})$ can be expressed as follows:

$$q^*(\mathbf{z}) = \operatorname{argmin}_{q \in \mathcal{Q}} \text{KL}(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x})), \quad (2)$$

where $\text{KL}(\cdot)$ is the KL-divergence term.

[0052] Extending the KL-divergence term produces the following:

$$\begin{aligned} \text{KL}(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z}, \mathbf{x})] \\ &\quad + \log p(\mathbf{x}), \end{aligned} \quad (3)$$

where $\mathbb{E}_{q(\mathbf{z})}(f(\mathbf{z}))$ denotes the expectation of function $f(\mathbf{z})$ under the probability distribution $q(\mathbf{z})$. From Equation (3), it can be concluded that the term $(\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z}, \mathbf{x})] - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})])$ is a lower bound of the log-likelihood $\log p(\mathbf{x})$ since the KL-divergence term is non-negative. This lower bound can be referred to as the Evidence Lower Bound (*ELBO*), which may be expressed as:

$$ELBO = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z}, \mathbf{x})] - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})] \quad (4)$$

From Equations (3) and (4), maximizing *ELBO* is equivalent to minimizing $\text{KL}(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x}))$ or maximizing the log-likelihood $\log p(\mathbf{x})$. In some instances, *ELBO* may also be referred to as the variational lower bound or negative variational free energy.

[0053] Some existing prediction methods use mean-field VI for the optimization problem shown in Equation (2). Mean-field VI assumes that the approximated posterior distribution $q(\mathbf{z})$ can be expressed as a factorized form $q(\mathbf{z}) = \prod_{j=1}^{|\mathbf{z}|} q(z_j)$, i.e., all z_j are mutually independent and each z_j is governed by its own factor distribution $q(z_j)$, whose parameters are referred to as variational parameters. Mean-field VI requires specification of the parametric form for each factor distribution $q(z_j)$ and derivation of parameter iterative equations by hand. One of the disadvantages of mean-field VI is that it is constrained to build models within only a small fraction of probability distributions; otherwise, no parameter iterative equation exists. This implies that the resulting models of existing prediction methods that use mean-field VI may lack the flexibility to explain the observed data. Moreover, the optimization of mean-field VI often relies on the Coordinate Ascent method which also struggles when the datasets are very large.

[0054] To address the disadvantages of mean-field VI, particularly its inflexibility to explain all observed data and its unsuitability to large-scale datasets, various implementations of the present disclosure use Variational Auto-Encoders (VAEs), which combine the automatic-differentiation (which is an ingredient of deep learning) with variational inference, thus yielding a flexible and efficient inference framework for probabilistic generative methods.

[0055] VAEs replace $q(\mathbf{z})$ in Equations (2) to (4) with $q(\mathbf{z}|\mathbf{x})$. Consequently, using VAEs, Equations (3) and (4) can be re-expressed as:

$$ELBO = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \text{KL}(q(\mathbf{z}|\mathbf{x})||\log p(\mathbf{z})), \quad (5)$$

where the terms $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{x}|\mathbf{z})$ can be parameterized with neural networks. The terms $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{x}|\mathbf{z})$ can be referred to as an inference network and a generative network, respectively. In some instances, VAEs assume $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{x}|\mathbf{z})$ follow two parametric distributions. Furthermore, VAEs fit the parameters of the two distributions with two neural networks. The inference network $q(\mathbf{z}|\mathbf{x})$ can take a datapoint $\mathbf{x}$ as an input and can produce a corresponding latent variable $\mathbf{z}$. The generative network $p(\mathbf{x}|\mathbf{z})$ can take latent variable $\mathbf{z}$ as an input and decodes datapoint $\mathbf{x}$.

[0056] The $ELBO$ can serve as a loss function for both the inference network $q(\mathbf{z}|\mathbf{x})$ and generative network $p(\mathbf{x}|\mathbf{z})$, which can be estimated with a Monte Carlo method by drawing L samples, $\mathbf{z}^{(l)}$, from $q(\mathbf{z}|\mathbf{x})$. This may be expressed as follows:

$$ELBO \approx \frac{1}{L} \sum_{l=1}^L \log p(\mathbf{x}|\mathbf{z}^{(l)}) - \text{KL}(q(\mathbf{z}|\mathbf{x})||\log p(\mathbf{z})), \quad (6)$$

where the KL-divergence term has an analytic solution for the Gaussian prior and posterior. The parameters of the inference network $q(\mathbf{z}|\mathbf{x})$ and generative network $p(\mathbf{x}|\mathbf{z})$ can be optimized by stochastic gradient descent methods, which easily scale to large-scale datasets.

Deep Probabilistic Spatial Transition (DeepST)

[0057] This section discusses various aspects of the proposed DeepST method. Among other aspects of the proposed DeepST method, this section discusses a generative process of the proposed DeepST method where the most likely route $\mathbf{r}$ between an origin and a destination is determined. This section also discusses a representation of a sequence of

road segments already traveled in the trip (also referred to as a past traveled route) and a representation of a destination. This section further discusses an inference framework (e.g., using VAEs) for inferring posterior distributions of latent variables and a prediction framework for the generative process that determines the most likely route $\mathbf{r}$ between an origin and a destination based on the posterior distributions of the latent variables. A discussion of the complexity of the proposed DeepST method is also presented in this section.

[0058] With regards to the generative process of the proposed DeepST method, according to various aspects of the present disclosure, the proposed DeepST determines (e.g., predicts or generates) the most likely route $\mathbf{r}$ (e.g., the road segments in the route $\mathbf{r}$) by conditioning the determination on a past traveled route, the representation of a destination, and the representation of real-time traffic. In such a manner, the proposed DeepST method simultaneously takes into consideration the aforementioned factors – the sequential property of transition and the global impact of destination and real-time traffic – in determining the most likely route.

[0059] FIG. 2 is a diagram showing a graphical model of an example generative process 200, where a route $\mathbf{r}$ between an origin and a destination is determined and conditioned on a representation of a past traveled route, a representation of a destination, and a representation of real-time traffic, according to an implementation of the present disclosure. The generative process 200 generates a road segment 202 (e.g., the $(i + 1)^{\text{th}}$ road segment, r_{i+1}) conditioned on a latent variable 204 (e.g., variable $\mathbf{c} \in \mathbb{R}^{|\mathbf{c}|}$) that can represent real-time traffic, a destination 206 (e.g., vector $\mathbf{x}$) that represents the trip destination, and a representation of the past traveled route 208 (e.g., vector $\mathbf{r}_{1:i}$). Intuitively, once the generative process 200 is trained on observed trajectories, its parameters may be tuned to capture the spatial transition patterns hidden in the dataset, and thus the proposed DeepST method can make predictions, in the future, based on the learned patterns.

[0060] In some examples, the generative process 200 shown in FIG. 2 can be described by the following operations:

Operation 1: Draw latent variable $\mathbf{c} \sim \text{Normal}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))$.

Operation 2: For $i + 1$ -th road segment, r_{i+1} , where $i \geq 1$:

- Draw $r_{i+1} \sim \mathbb{P}(r_{i+1} | \mathbf{r}_{1:i}, \mathbf{x}, \mathbf{c})$.
- Draw ending indicator $s \sim \text{Bernoulli}(f_s(r_{i+1}, \mathbf{x}))$.

- If $s = 0$ then continue, else end the generation.

[0061] Operation 1 may denote selecting (e.g., drawing) the latent variable 202 (e.g., latent variable $\mathbf{c} \in \mathbb{R}^{|\mathbf{c}|}$) from a distribution. In the example shown in FIG. 2, $\mathbf{c} \sim \text{Normal}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))$. Stated differently, in some instances, the latent variable 204 (e.g., latent variable $\mathbf{c} \in \mathbb{R}^{|\mathbf{c}|}$, which represents the real-time traffic) can be drawn from an isotropic Gaussian prior distribution.

[0062] Operation 2 may include selecting (e.g., drawing) the $i + 1$ -th road segment 202 (e.g., r_{i+1} , where $i \geq 1$) from the distribution $\mathbb{P}(r_{i+1} | \mathbf{r}_{1:i}, \mathbf{x}, \mathbf{c})$. In Operation 2, a Bernoulli distribution (e.g., whose parameter is the output of function f_s taking $r_{i+1}, \mathbf{x}$ as input) can be used to decide whether to terminate the generation of the $i + 1$ -th road segment 202. The Euclidean distance between the projection point of $\mathbf{x}$ on r_{i+1} and $\mathbf{x}$ can be used to determine the termination of the generation, namely:

$$f_s(r_{i+1}, \mathbf{x}) = \frac{1}{1 + \|\mathbf{p}(\mathbf{x}, r_{i+1}) - \mathbf{x}\|_2}, \quad (7)$$

where $\mathbf{p}(\mathbf{x}, r_{i+1})$ denotes the projection point of $\mathbf{x}$ on r_{i+1} . The Bernoulli distribution may be chosen since s is a binary variable. The function f_s can be any differentiable function.

[0063] According to various aspects of the present disclosure, the distribution $\mathbb{P}(r_{i+1} | \mathbf{r}_{1:i}, \mathbf{x}, \mathbf{c})$ can be modeled as a categorical distribution, which can be a discrete probability distribution that describes the possible results of a random variable (e.g., r_{i+1}) that can take on one of R possible categories, with the probability of each category separately specified. In the proposed categorical distribution, the R possible categories are all road segments adjacent (e.g., immediately adjacent) to road segment r_i whose parameter is the output of a function f taking the past traveled route sequence $\mathbf{r}_{1:i}$, destination $\mathbf{x}$, and real-time traffic $\mathbf{c}$ as input.

[0064] With regards to the distribution $\mathbb{P}(r_{i+1} | \mathbf{r}_{1:i}, \mathbf{x}, \mathbf{c})$, the additive function can be leveraged due to its simplicity to express the distribution $\mathbb{P}(r_{i+1} | \mathbf{r}_{1:i}, \mathbf{x}, \mathbf{c})$ as follows:

$$\mathbb{P}(r_{i+1} | \mathbf{r}_{1:i}, \mathbf{x}, \mathbf{c}) = \text{softmax}(\alpha^\top f_r(\mathbf{r}_{1:i}) + \beta^\top f_x(\mathbf{x}) + \gamma^\top \mathbf{c}), \quad (8)$$

where $f_r(\mathbf{r}_{1:i}) \in \mathbb{R}^{n_r}$, $f_x(\mathbf{x}) \in \mathbb{R}^{n_x}$ are representations of past traveled route $\mathbf{r}_{1:i}$ and destination $\mathbf{x}$, respectively. Furthermore, $\alpha \in \mathbb{R}^{n_r \times \mathcal{N}(r_i)}$, $\beta \in \mathbb{R}^{n_x \times \mathcal{N}(r_i)}$, $\gamma \in \mathbb{R}^{|\mathbf{c}| \times \mathcal{N}(r_i)}$ are projection matrices that map the corresponding representations into the road segment space, and $\mathcal{N}(r_i)$ is the number of adjacent, subsequent road segments of the road

segment r_i . Illustratively, $\mathcal{N}(r_2) = 3$ in FIG. 1A since road segments r_4, r_5 , and r_7 (e.g., a total of three subsequent road segments) are adjacent to road segment r_2 .

[0065] The projection matrices can be shared across all road segments. Since different road segments r_i may have different number of adjacent road segments, the term $\mathcal{N}(r_i)$ can be substituted with $\max_{r \in E} \mathcal{N}(r)$, which is the maximum number of neighboring or adjacent road segments on the road network. Since the generative process 200 is data-driven, its parameters can be tuned to push the probability mass of distribution $\mathbb{P}(r_{i+1} | \mathbf{r}_{1:i}, \mathbf{x}, \mathbf{c})$ towards the road segments that are truly adjacent to road segment r_i .

[0066] With regards to the representation of the past traveled route 208 (e.g., vector $\mathbf{r}_{1:i}$ shown in FIG. 2), some existing methods model the transition relationship (e.g., the transition from one road segment to the next) with a Markov model, which requires explicit dependency assumptions to make an inference tractable. In contrast, the proposed DeepST method uses one or more RNNs to represent the past traveled route 208 (e.g., vector $\mathbf{r}_{1:i}$ shown in FIG. 2). RNNs can model the long-range dependency by embedding a sequence of tokens into a vector. Since the transition patterns of a vehicle on a road network may demonstrate strong sequential properties, it may be desirable to capture such long-range dependencies when modeling the spatial transition (e.g., the transition from one road segment to the next in a route). In some instances, representing the past traveled route 208 (e.g., vector $\mathbf{r}_{1:i}$ shown in FIG. 2) using one or more RNNs includes updating the i -th hidden state, which can be expressed as follows:

$$\mathbf{h}_i = \begin{cases} \mathbf{0} & i = 1; \\ \text{GRU}(\mathbf{h}_{i-1}, r_{i-1}) & i \geq 2, \end{cases} \quad (9)$$

where $\mathbf{h}_1$ is initialized as a zero vector and $\text{GRU}(\cdot, \cdot)$ represents the Gated Recurrent Unit updating function. In some instances, the i -th hidden state $\mathbf{h}_i$ is chosen as the representation of the past traveled route and $f_r(\mathbf{r}_{1:i-1}) = \mathbf{h}_i$.

[0067] With regards to the representation of the destination 206 (e.g., vector $\mathbf{x}$ in FIG. 2), trip destinations can have a global impact on the spatial transition (e.g., the transition from one road segment to the next in a route). Therefore, the proposed DeepST method efficiently and effectively learns the representations $f_x(\mathbf{x}) \in \mathbb{R}^{n_x}$ of the destinations to help the route decision process. In many real-world scenarios, the exact road segment of the destination may not be available at the start of a trip. Furthermore, as discussed above in relation to existing route-prediction methods, learning the representation of each

destination road segment separately prevents effective sharing of the statistical strength across trips with spatially close destination. As an example, referring to FIG. 1A and Table 1 above, if the destinations are treated separately, the following probabilities are obtained:

$$\mathbb{P}(r_{10}|r_5, C) = 1/3; \mathbb{P}(r_6|r_5, A) = 1/3; \mathbb{P}(r_6|r_5, B) = 1/3 \quad (10)$$

[0068] However, the spatial proximity of destinations A and B is small, and if there is a single representation AB used for all trips driving towards destination A or destination B, then the probabilities can be obtained:

$$\mathbb{P}(r_{10}|r_5, C) = 1/3; \mathbb{P}(r_6|r_5, AB) = 2/3 \quad (11)$$

As seen in Equation (11), the transition patters can be shared to mutually reinforce the transition probability between road segment r_5 and road segment r_6 across all trips towards destination A or destination B.

[0069] Intuitively, trips whose destinations are spatially close to each other can share similar destination representations. The destination representations $f_x(\mathbf{x}) \in \mathbb{R}^{n_x}$ that are learned by the proposed DeepST method can be referred to as learned destination representations. The learned destination representations can effectively guide the spatial transition of their corresponding trips. Separating the learning of the representations of destinations and the learning of spatial transitions into two stages (e.g., as in existing methods) can prevent end-to-end training, which can lead to a suboptimal solution. Therefore, to achieve end-to-end training (and in contrast to existing methods), the proposed DeepST method jointly learns the representations of the destination 206 (e.g., vector $\mathbf{x}$ in FIG. 2) and models the spatial transitions such that the statistical strength can be effectively shared across different trips. Consequently, in the proposed DeepST method, if the spatial proximity between the destinations of two trips is small, the learned destination representations may be similar, and the learned destination representations can effectively guide the spatial transition of their respective trips.

[0070] FIG. 3 is a diagram showing a graphical model of an example generative process 300 of the proposed DeepST method including the generative process 200 shown in FIG. 2 and an example adjoint generative process that simultaneously learns multiple destination representations and adaptively allocates trips into one of these destination

representations, according to an implementation of the present disclosure. The generative process 300 is a more detailed graphical model of the generative process 200 shown in FIG. 2. Specifically, the generative process 300 accounts for the adjoint learning of the multiple destination representations 206 (vector $\mathbf{x}$) and the adaptive allocation of trips into one of these destination representations 206.

[0071] As discussed above, intuitively, trips whose destinations are spatially close to each other can share similar destination representations. Therefore, the proposed DeepST method learns representations for K -destination proxies that are shared by all trips, instead of learning a separate destination representation for each trip. Specifically, the proposed DeepST method simultaneously learns representations of K -spatial locations, which can be referred to as K -destination proxies, and adaptively allocates the trips into one of these destination proxies. The destination proxy representation can be used to guide the spatial transition of trips that are allocated to this proxy. This simultaneous learning of the K -destination proxies and the adaptive allocation of the trips into one of these destination proxies can be achieved by an adjoint generative model.

[0072] The adjoint generative model explains the generation of the destination proxy's coordinates with another latent variable 302 (e.g., latent variable $\boldsymbol{\pi}$) and can be described by the following operations:

Operation 1: Draw $\boldsymbol{\pi} \sim \text{Categorical}(\boldsymbol{\eta})$.

Operation 2: Draw $\mathbf{x} \sim \text{Normal}(M\boldsymbol{\pi}, \text{diag}(S\boldsymbol{\pi}))$,

where $\boldsymbol{\eta} \in \mathbb{R}^K$ is a hyperparameter 304, and $\boldsymbol{\pi} \in \mathbb{N}^K$ is a one-hot vector indicating which proxy the trip is allocated to. In some instances, the latent variable 302 (e.g., latent variable $\boldsymbol{\pi}$) can be considered a destination allocation indicator, and the nonzero dimension of $\boldsymbol{\pi}$ indicates the proxy to which the trip is allocated.

[0073] In some instances of the proposed DeepST method, the latent variable $\boldsymbol{\pi}$ and matrices M, S can be used to generate the destination coordinate $\mathbf{x}$, where $M \in \mathbb{R}^{2 \times K}$ is a mean-value matrix 306 and $S \in \mathbb{R}_+^{2 \times K}$ is a variance matrix 308 of the K -destination proxies. In Operation 2, the sub-operation $M\boldsymbol{\pi}$ selects one column of M , which corresponds to the mean of the allocated proxy. Furthermore, in Operation 2, the sub-operation $S\boldsymbol{\pi}$ selects one column of S , which corresponds to the variance of the allocated proxy. The proposed DeepST method interprets the observation $\mathbf{x}$ (e.g., in Operation 2 of the adjoint generative process) via conditioning on $M\boldsymbol{\pi}$ and $S\boldsymbol{\pi}$ with a Gaussian distribution, which can tolerate the small deviation of $\mathbf{x}$ from the proxy mean value $M\boldsymbol{\pi}$.

Therefore, adding some small noise to the observation $\mathbf{x}$ does not change its proxy. Hence, the proposed DeepST method is robust to inaccuracies in the observation $\mathbf{x}$. As discussed above, each trip destination $\mathbf{x}$ can be allocated to a proxy, and the trips that are allocated to the same proxy can share the same proxy representation to effectively share the statistical strength across these trips.

[0074] Since the purpose of learning the representation of a destination is to obtain the proxy destination representation, an embedding matrix 310 (e.g., matrix $W \in \mathbb{R}^{n_x \times K}$) is introduced, and the representations $f_x(\mathbf{x})$ can be expressed as $f_x(\mathbf{x}) = W\boldsymbol{\pi}$. The introduction of the latent variable $\boldsymbol{\pi}$ may allow the simultaneous learning of the K -destination proxies and the adaptive allocation of the trips into one of these destination proxies. Furthermore, $M\boldsymbol{\pi}$ and $S\boldsymbol{\pi}$ can explain the observations $\mathbf{x}$ to satisfy the spatial constraint. Additionally, $W\boldsymbol{\pi}$ can yield a useful destination representation that can effectively guide the spatial transition of the trips, since this representation is learned by maximizing the probability of observed routes in $\mathbb{P}(r_{i+1}|\mathbf{r}_{1:i}, W\boldsymbol{\pi}, \mathbf{c})$.

[0075] With regards to the inference framework for inferring posterior distributions, the generative process 300 shown in FIG. 3 implies the following log-likelihood function for one trip:

$$\begin{aligned} \mathcal{L}(\theta) = & \log \sum_{\boldsymbol{\pi}} \int_{\mathbf{c}} \mathbb{P}(\mathbf{c}|\boldsymbol{\mu}, \boldsymbol{\sigma}^2) \mathbb{P}(\boldsymbol{\pi}|\boldsymbol{\eta}) \\ & \times \prod_{i=1}^{n-1} \mathbb{P}(r_{i+1}|\mathbf{r}_{1:i}, W\boldsymbol{\pi}, \mathbf{c}) d\mathbf{c} + \log \sum_{\boldsymbol{\pi}} \mathbb{P}(\boldsymbol{\pi}|\boldsymbol{\eta}) \mathbb{P}(\mathbf{x}|\boldsymbol{\pi}, M, S) \end{aligned} \quad (12)$$

where θ is the collection of all parameters involved. The first term describes the log-likelihood of route generation, while the second term represents the log-likelihood of destination generation.

[0076] As seen in Equation (12), an inference of the posterior distributions $P(\mathbf{c}|\mathbf{r}, \mathbf{x})$ and $P(\boldsymbol{\pi}|\mathbf{r}, \mathbf{x})$ and fitting of the parameters θ by maximizing the log-likelihood given the observations may be needed. The exact computation of the log-likelihood function can be intractable since it may require an integration (e.g., a sum) of two latent variables $\mathbf{c}, \boldsymbol{\pi}$ in a high-dimensional space. Furthermore, existing sampling-based methods can suffer from slow convergence and may not be suitable for the proposed DeepST method.

[0077] To circumvent the drawbacks of intractability in exact computation and slow convergence in sampling-based methods, an approximate inference method is used for the proposed DeepST method, within the Variational Auto-Encoders (VAEs) framework.

The general idea of VAEs is to fit the intractable posterior distributions with appropriate inference neural nets. Since the exact computation of log-likelihood is difficult, VAEs resort to maximize the *ELBO* of the log-likelihood and propagate the gradients backwards to optimize the inference neural nets.

[0078] With regards to Equation (12), the *ELBO* of $\mathcal{L}(\theta)$ can be obtained using the Jensen's Inequality as follows:

$$\begin{aligned} \mathcal{L}(\theta) \geq \mathbb{E}_{q(\mathbf{c}, \boldsymbol{\pi}|\mathbf{r}, \mathbf{x})} \left[\log \frac{\mathbb{P}(\mathbf{c})\mathbb{P}(\boldsymbol{\pi}) \prod_{i=1}^{n-1} \mathbb{P}(r_{i+1}|\mathbf{r}_{1:i}, W\boldsymbol{\pi}, \mathbf{c})}{q(\mathbf{c}, \boldsymbol{\pi}|\mathbf{r}, \mathbf{x})} \right] \\ + \mathbb{E}_{q(\boldsymbol{\pi}|\mathbf{r}, \mathbf{x})} \left[\log \frac{\mathbb{P}(\boldsymbol{\pi})\mathbb{P}(\mathbf{x}|\boldsymbol{\pi}, M, S)}{q(\boldsymbol{\pi}|\mathbf{r}, \mathbf{x})} \right], \end{aligned} \quad (13)$$

where $q(\mathbf{c}, \boldsymbol{\pi}|\mathbf{r}, \mathbf{x})$ and $q(\boldsymbol{\pi}|\mathbf{r}, \mathbf{x})$ are the approximated posterior distributions to be optimized, and $q(\mathbf{c}, \boldsymbol{\pi}|\mathbf{r}, \mathbf{x})$ represents the joint posterior distribution of $\mathbf{c}, \boldsymbol{\pi}$ given the route $\mathbf{r}$ and destination $\mathbf{x}$.

[0079] Since $\boldsymbol{\pi}$ denotes the proxy that destination $\mathbf{x}$ is allocated to, and this allocation can be invariant to the traveled route $\mathbf{r}$, the joint distribution can be factorized as follows:

$$q(\mathbf{c}, \boldsymbol{\pi}|\mathbf{r}, \mathbf{x}) = q(\mathbf{c}|\mathbf{r})q(\boldsymbol{\pi}|\mathbf{x}) \quad (14)$$

Similarly, $q(\boldsymbol{\pi}|\mathbf{x})$ in Equation (14) can be factorized as $q(\boldsymbol{\pi}|\mathbf{x}) = q(\boldsymbol{\pi}|\mathbf{r}, \mathbf{x})$.

[0080] The approximate inference method used for the proposed DeepST method varies slightly from typical VAEs. For example, in the typical usage of VAEs when performing inference, the data to be generated is observable. In the approximate inference method used for the proposed DeepST method, however, the goal may be to predict the most likely route $\mathbf{r}$, which may not be available at the time of prediction. To address this difference between the approximate inference method used for the proposed DeepST method and inference in the typical usage of VAEs, the expression $q(\mathbf{c}|\mathbf{r})$ in Equation (14) can infer the posterior distribution of traffic condition $\mathbf{c}$ from the transition patterns of the route $\mathbf{r}$. Intuitively, the real-time traffic at the start of trip $\mathbf{T}$ can also be measured by the average speed of trajectories (or sub-trajectories) in the time window $[\mathbf{T}.s - \Delta, \mathbf{T}.s)$, where Δ is a specified time parameter (e.g., measured in minutes, an example being 20 minutes, or measured in some other timescale). These trajectories may be denoted as $\mathcal{C}$ (or $\mathbf{T}.C$), and the real-time traffic representation $\mathbf{c}$ can be extracted from

C . Consequently, the road segment space can be partitioned into cells, and the average vehicle speed in each cell can be determined based on the real-time trajectories.

[0081] However, the raw cell representation can be sensitive to the spatial distribution of vehicles within the cell. For example, if there is no sensing vehicle traveling in the cell at a given moment in time, the cell value can be zero for the given moment in time. Furthermore, the raw cell representation cannot reveal the similarity between two analogous traffic conditions, and even if only one cell value changes, the representation can be considered a different cell representation from the original one. Consequently, to circumvent these sensitivities in the raw cell representation, in the proposed DeepST method, high-level features are extracted from the raw cell representation. Specifically, the proposed DeepST method uses a Convolutional Neural Net (CNN) to extract the high-level features from the raw cell representation. More formally, this can be expressed as follows:

$$\begin{aligned} \mathbf{f} &= \text{CNN}(C); \\ q(\mathbf{c}|C) &= \text{Normal}\left(\mu(\mathbf{f}), \text{diag}(\sigma^2(\mathbf{f}))\right), \end{aligned} \quad (15)$$

where $\mu(\mathbf{f}), \sigma^2(\mathbf{f})$ are parameterized by two multi-layer perceptrons (MLPs) with a shared hidden layer.

[0082] FIG. 4 is a diagram showing an example inference framework 400 of the proposed DeepST method, according to an implementation of the present disclosure. The inference framework 400 includes an inference network 402 and a generation network 404. In the inference network 402, an encoder 406 (denoted in FIG. 4 as Encoder NN_1) may be a stack of CNN+MLPs, which approximates the posterior distribution $q(\mathbf{c}|C)$. An encoder 408 (denoted in FIG. 4 as Encoder NN_2) approximates the distribution $q(\boldsymbol{\pi}|\mathbf{x})$ and may be parameterized by an MLP. The encoders 408, 406 can be referred to as first and second inference neural networks, respectively, and the parameters set involved can be denoted as Φ . For a given trip $\mathbf{T}$, the inputs $\mathbf{T}.C$ (indicated in FIG. 4 as reference numeral 405) and $\mathbf{T}.\mathbf{x}$ (indicated in FIG. 4 as destination (e.g., exact) coordinate 206') are respectively provided to the inference neural networks 406, 408 to draw the instances of random variables $\mathbf{c}$ and $\boldsymbol{\pi}$, respectively (denoted in FIG. 4 as variables 204, 302, respectively). The sampled instances 204, 302 are generated as outputs of the inference operation 402, and provided as inputs to the generation network 404. In particular, the

sampled instances 204, 302 are provided as inputs to generative processes 410, 412, respectively, (denoted in FIG. 4 as Generative NN₁ and Generative NN₂, respectively) to compute the subsequent loss. Specifically, with the generated L samples $\{\boldsymbol{\pi}^{(l)}, \mathbf{c}^{(l)}\}_{l=1}^L$, the *ELBO* can be estimated using a Monte Carlo method as follows:

$$\begin{aligned} ELBO \approx & \frac{1}{L} \sum_{l=1}^L \sum_{i=1}^{n-1} \log \mathbb{P}(r_{i+1} | \mathbf{r}_{1:i}, W\boldsymbol{\pi}^{(l)}, \mathbf{c}^{(l)}) \\ & + \frac{1}{L} \sum_{l=1}^L \sum_{i=1}^{n-1} \log \mathbb{P}(\mathbf{x} | \boldsymbol{\pi}^{(l)}, M, S) \\ & - \text{KL}(q(\mathbf{c}|C) || \mathbb{P}(\mathbf{c})) - 2\text{KL}(q(\boldsymbol{\pi}|\mathbf{x}) || \mathbb{P}(\boldsymbol{\pi})), \end{aligned} \quad (16)$$

where $\text{KL}(\cdot || \cdot)$ denotes the KL-divergence.

[0083] Equation (16) involves the evaluation of log-probability of two distributions, $\mathbb{P}(r_{i+1} | \mathbf{r}_{1:i}, \mathbf{x}, \mathbf{c})$ and $\mathbb{P}(\mathbf{x} | \boldsymbol{\pi}, M, S)$, where the superscript (l) for these expressions are ignored for clarity. The distribution $\mathbb{P}(r_{i+1} | \mathbf{r}_{1:i}, \mathbf{x}, \mathbf{c})$ can characterizes the probability of the next possible road r_{i+1} with a softmax distribution, and the log-probability can be calculated as follows,

$$\log \mathbb{P}(r_{i+1} | \mathbf{r}_{1:i}, \mathbf{x}, \mathbf{c}) = \log \frac{\exp(\alpha_j^T \mathbf{h}_i + \beta_j^T W\boldsymbol{\pi} + \gamma_j^T \mathbf{c})}{\sum_{j'} (\alpha_{j'}^T \mathbf{h}_i + \beta_{j'}^T W\boldsymbol{\pi} + \gamma_{j'}^T \mathbf{c})}, \quad (17)$$

where α_j is the j -th column of α . The distribution $\mathbb{P}(\mathbf{x} | \boldsymbol{\pi}, M, S)$ is the Normal distribution with mean $M\boldsymbol{\pi}$ and variance $S\boldsymbol{\pi}$, whose log-probability calculation is straightforward.

[0084] When the approximated posterior $q(\mathbf{c}|C)$ and prior $p(\mathbf{c})$ are both Gaussian distributions, the KL-divergence $\text{KL}(q(\mathbf{c}|C) || p(\mathbf{c}))$ has a closed-form solution, which can be expressed as follows:

$$\text{KL}(q(\mathbf{c}|C) || p(\mathbf{c})) = \frac{1}{2} \sum_{i=1}^{|\mathbf{c}|} (1 - \mu_i^2 - \sigma_i^2 + \log \sigma_i^2). \quad (18)$$

[0085] To enable the gradients flowing back into the inference nets, the random variables $\mathbf{c}$ and $\boldsymbol{\pi}$ can be re-parameterized when sampling. For example, for the Gaussian random variable $\mathbf{c}$, this random variable $\mathbf{c}$ can be re-parameterized as follows:

$$\mathbf{c} \sim \text{Normal}(\mathbf{c} | \mu, \sigma^2) \Leftrightarrow \mathbf{c} = \mu + \sigma \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \text{Normal}(0, 1). \quad (19)$$

[0086] Since the random variable $\boldsymbol{\pi}$ is a discrete random variable that is not amenable to the re-parameterization shown in Equation (19). Consequently, in various

implementations, the random variable π can be re-parameterized using Gumbel-Softmax relaxation.

[0087] Based on the description above, the learning process (also referred to as the training stage) of the proposed DeepST method can be summarized as follows:

Learning Process

Input: Training dataset: $(\mathbf{r}_m, C_m, \mathbf{x}_m)_{m=1}^M$

Output: Parameter sets: Θ and Φ

```

1 while training is true do
2    $B \leftarrow$  A random minibatch of data;
3   for  $b = 1, \dots, |B|$  do
4      $\mathbf{c}^{(b)} \sim q(\mathbf{c}^{(b)} | C^{(b)})$ ;
5      $\pi^{(b)} \sim q(\pi^{(b)} | \mathbf{x}^{(b)})$ ;
6   ELBO  $\leftarrow$  Calculating the ELBO using Equation (16)
7   for  $\theta \in \Theta \cup \Phi$  do
8      $\mathbf{g}_\theta \leftarrow \nabla_\theta \text{ELBO}$ ;
9      $\theta \leftarrow \theta + \Gamma(\mathbf{g}_\theta)$ ;
10 return  $\Theta$  and  $\Phi$ ;

```

[0088] In Operation 4 of the learning process, it may be unnecessary to compute C for each trip since the temporal dimension can be discretized into slots, and the trips whose start times fall into the same slot can share one C . Furthermore, since a mini-batch of data is used to compute the *ELBO* in Operation 6 of the learning process, the number of random variables L can be set to 1 in Equation (16) and low-variance gradient estimators are still achievable in Operation 8 of the learning process. In Operation 9 of the learning process, the parameters of the model are updated with gradient descent. Consequently, a stochastic gradient descent (SGD) method can be used to train the model used in the proposed DeepST method.

[0089] There might be multiple routes between a pair of origin and destination in the training dataset. In such a case, the model can learn to assign different likelihood scores to these routes since all of them are used in the training stage, and in the prediction stage only the one with the highest likelihood score may be returned.

[0090] In route prediction, given the trained network parameter sets θ and ϕ (e.g., generated by the learning process above), the initial road segment $\mathbf{T}.r_1$, the set of trajectories $\mathbf{T}.C$ for representing real-time traffic, and destination $\mathbf{T}.\mathbf{x}$ of a trip $\mathbf{T}$, the proposed DeepST method executes the complete generative process 300 shown in FIG. 3

to generate the most probable route for the trip $\mathbf{T}$. This process, which may be referred to as the prediction process of the proposed DeepST method, can be summarized as follows:

Prediction Process

Input: $(\Theta, \Phi, \mathbf{T}.r_1, \mathbf{T}.C, \mathbf{T}.\mathbf{x})$
Output: Generated route: $\mathbf{T}.\mathbf{r}$

```

1  $r_1 \leftarrow \mathbf{T}.r_1, C \leftarrow \mathbf{T}.C, \mathbf{x} \leftarrow \mathbf{T}.\mathbf{x};$ 
2 Draw  $\mathbf{c} \sim q(\mathbf{c}|C);$ 
3 Draw  $\boldsymbol{\pi} \sim q(\boldsymbol{\pi}|\mathbf{x});$ 
4 for  $i \geq 1$  do
5   Draw  $r_{i+1} \sim \mathbb{P}(r_{i+1}|\mathbf{r}_{1:i}, W\boldsymbol{\pi}, \mathbf{c});$ 
6   Draw ending indicator  $s \sim \text{Bernoulli}(f_s(r_{i+1}, \mathbf{x}));$ 
7   if  $s = 0$  then
8     continue;
9   else
10    break;
11 return  $\mathbf{r};$ 

```

[0091] Unlike the training stage of the proposed DeepST method, the prediction process samples latent random variables $\mathbf{c}$, $\boldsymbol{\pi}$ from the learned posterior distributions (e.g. in Operations 1 to 3). The prediction process uses the sampled random variables $\mathbf{c}$, $\boldsymbol{\pi}$ and the learned parameter W to generate the next road segment r_{i+1} (e.g., in Operation 4) which can be defined as follows:

$$\mathbb{P}(r_{i+1}|\mathbf{r}_{1:i}, W\boldsymbol{\pi}, \mathbf{c}) = \text{softmax}(\alpha^\top \mathbf{h}_i + \beta^\top W\boldsymbol{\pi} + \gamma^\top \mathbf{c}). \quad (20)$$

[0092] With regards to the route likelihood score, scoring the likelihood of a given route $\mathbf{r}$ may be similar to the route prediction, except that the route is fixed. Consequently, for determining the route likelihood score, after drawing the random variables $\mathbf{c}$ and $\boldsymbol{\pi}$ from the posterior distribution, the likelihood of a route $\mathbf{r}$ can be calculated as $\prod_{i=1}^{|\mathbf{r}|-1} \mathbb{P}(r_{i+1}|\mathbf{r}_{1:i}, W\boldsymbol{\pi}, \mathbf{c})$.

[0093] With regards to the complexity of the proposed DeepST method, since the model is trained with the SGD method, the convergence time is linearly to the number of trajectories in the training dataset. Consequently, given a trained model, the complexity of the route prediction and the complexity of the route likelihood score are each $O(|\mathbf{r}|)$. The scalability and the complexity of the proposed DeepST method are investigated in experiments and discussed in further detail below.

Experimental Setup

[0094] The effectiveness of the proposed DeepST method on two real-world large scale trajectory datasets was investigated through experiments and compared to existing methods. FIG. 5A is a diagram showing an example spatial distribution of GPS points in a first example road segment space 500, according to an implementation of the present disclosure. FIG. 5B is a diagram showing an example spatial distribution of GPS points in a second example road segment space 501, according to an implementation of the present disclosure. The GPS points shown in FIGS. 5A and 5B can serve as origin or destination points. The example road segment space 500 shown in FIG. 5A is an example network of roads located in a 10 km x 10 km area in a provincial capital city, Chengdu, in China. The example road segment space 501 shown in FIG. 5B is an example network of roads located in a 28 km x 30 km area in another provincial capital city, Harbin, in China. In the experiments, a route $\mathbf{r}$ was determined between an origin and a destination, and the route $\mathbf{r}$ included road segments that can, as an example, be part of the first example road network shown in FIG. 5A (e.g., in cases where the origin and destination are GPS points in the 10 km x 10 km area of Chengdu shown in FIG. 5A) or part of the second example road network shown in FIG. 5B (e.g., in cases where the origin and destination are GPS points in the 28 km x 30 km area of Harbin shown in FIG. 5B).

[0095] As discussed above, two real-world large scale trajectory datasets were used in the experiments. The first trajectory dataset is associated with the spatial distribution of GPS points in the first example road network shown in FIG. 5A. For the sake of simplicity, the first trajectory dataset is referred to in this disclosure as the Chengdu dataset. The Chengdu dataset is part of a publicly available dataset released by the DiDi Company as part of the GAIA initiative. The publicly available dataset was collected by 33,000 taxi cabs in Chengdu, China with a sampling rate of about 3 seconds. In the experiments, the first 15 days of data from the publicly available dataset were used to generate the Chengdu dataset. The Chengdu dataset includes over 3 million trajectories and covers 3,185 road segments (e.g., the road segments shown in FIG. 5A). The second trajectory dataset is associated with the spatial distribution of GPS points in the second example road network shown in FIG. 5B. For the sake of simplicity, the first trajectory dataset is referred to in this disclosure as the Harbin dataset. The Harbin dataset was collected by 13,000 taxi during a one-month in Harbin, China with a sample rate of about 30 seconds. The Harbin

dataset includes over 2.9 million trajectories and covers 12,497 road segments (e.g., the road segments shown in FIG. 5B).

[0096] FIG. 6A is a diagram showing a distribution 600 of travel distances for the Chengdu dataset, according to an implementation of the present disclosure. The horizontal axis of distribution 600 denotes travel distance in kilometers (km), and the vertical axis of distribution 600 denotes the frequency of occurrence of the respective travel distances. FIG. 6B is a diagram showing a distribution 601 of the number of road segments used in a trip for the Chengdu dataset, according to an implementation of the present disclosure. The horizontal axis of distribution 601 denotes the number of road segments used in a trip, and the vertical axis of distribution 601 denotes the frequency of occurrence of the number of road segments. Table 3, shown below, reports the basic statistics of the distributions seen in FIGS. 6A and 6B for the Chengdu dataset.

Table 3: Dataset Statistics.

City	Chengdu			Harbin		
Measures	Min	Max	Mean	Min	Max	Mean
Distance (km)	1.0	40	4.9	1.1	60	11.4
# Road Segments	5	85	14	5	102	24

As seen in Table 3, the mean travel distance and the mean number of road segments are 4.8 km and 14, respectively for the Chengdu dataset.

[0097] FIG. 7A is a diagram showing a distribution 700 of travel distances for the Harbin dataset, according to an implementation of the present disclosure. The horizontal axis of distribution 700 denotes travel distance in kilometers (km), and the vertical axis of distribution 700 denotes the frequency of occurrence of the respective travel distances. FIG. 7B is a diagram showing a distribution 701 of the number of road segments used in a trip for the Harbin dataset, according to an implementation of the present disclosure. The horizontal axis of distribution 701 denotes the number of road segments used in a trip, and the vertical axis of distribution 701 denotes the frequency of occurrence of the number of road segments. Table 3, shown above, reports the basic statistics of the distributions seen in FIGS. 7A and 7B for the Harbin dataset. As seen in Table 3, the mean travel distance and the mean number of road segments are 11.4 km and 24, respectively for the Harbin dataset.

[0098] In evaluating the efficacy of the proposed DeepST method for the Chengdu dataset, the first 8 days of trajectories were used as a training dataset, the next 2 days of trajectories were used for validation, and the remaining ones were used for testing. Therefore, the dataset size for training, validation, and testing using the Chengdu dataset was 1.6 million, 0.4 million, and 1 million, respectively. Furthermore, in evaluating the efficacy of the proposed DeepST method for the Harbin dataset, the first 18 days of trajectories were used as a training dataset, the next 3 days of trajectories were used for validation, and the remaining ones were used for testing. Therefore, the dataset size for training, validation, and testing using the Harbin dataset was 1.7 million, 0.3 million, and 0.9 million, respectively.

[0099] In the experiments, the proposed DeepST method was evaluated against existing methods on two tasks: (1) route prediction (e.g., predicting the most likely route); and (2) route recovery from sparse trajectories. For the task of the most likely route prediction, the proposed DeepST method was compared to DeepST-C, RNN, the first-order Markov Model (MMI), Weighted Shortest Path (WSP), and an existing route decision model called Constrained State Space RNN (CSSRNN). The DeepST-C mentioned above is a simplified version of the proposed DeepST method without considering the impact of real-time traffic. The RNN mentioned above is a simplified RNN that only takes the initial road segment as input and ignores the impact of both the destination and real-time traffic. The CSSRNN mentioned above assumes the last road segments of the trips are known in advance and learns their representations to help model the spatial transition. The MMI mentioned above models the spatial transition by calculating the first-order conditional probability between adjacent road segments from the historical trips. The WSP mentioned above always returns the shortest path from the origin road segment to the destination road segment on the weighted road network. The edge weight equals to the mean travel time of corresponding road segment, and the mean travel time is estimated using the entire historical dataset.

[00100] For the task of route recovery from sparse trajectories, the STRS method proposed by H. Wu, J. Mao, W. Sun, B. Zheng, H. Zhang, Z. Chen, and W. Wang in “Probabilistic robust route recovery with spatio-temporal dynamics,” in SIGKDD, 2016 was considered. The STRS mentioned above includes a travel time inference module and a spatial transition inference module. Some experiments substituted the spatial transition inference module of the STRS method with the proposed DeepST method to demonstrate

how the proposed DeepST method can be used to enhance the performance of the STRS method. The STRS method with the substituted spatial transition inference module is referred to in this disclosure as the STRS+ method.

[00101] With regards to the experimental platform and parameter settings, the experiment platform used was Ubuntu 16.04 OS, and the proposed DeepST method was implemented with PyTorch 1.0, and trained with one Tesla P100 GPU for about 6 hours for the Chengdu dataset and for about 10 hours for the Harbin dataset. For the Chengdu dataset, the number of destination proxies K was set to 500, and the road segment space was partitioned into a 87×98 matrix with a cell size of about $100 \text{ m} \times 100 \text{ m}$. For the Harbin dataset, the number of destination proxies K was set to 1000, and the road segment space was partitioned into a 138×148 matrix with a cell size of about $200 \text{ m} \times 200 \text{ m}$. In the experiments, the CNN in Equation (15) included three connected convolution blocks followed by an average pooling layer, and each convolution block included the following three layers: Conv2d $\rightarrow$ BatchNorm2d $\rightarrow$ LeakyReLU. In the experiments, the dimension of real-time traffic representation $|\mathbf{c}|$ and the hidden size of all MLPs used were set to 256, and n_r and n_x were set to 128. Furthermore, a three-layer stacking GRU with hidden size 256 was utilized for all RNNs used in the experiments. In the experiments, the time window size $\Delta = 30$ minutes, and the temporal dimension was discretized into slots, each slot having a size of 20 minutes. In the experiments, two trips share the same C if their start times fall into the same slot. The batch size $|\mathcal{B}|$ was 128, and the model was trained with the Adam method, discussed in D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," CoRR, 2014, for 15 epochs, and early stopping was used on the validation dataset.

[00102] With regards to the prediction of the most likely route, different existing methods were also evaluated on the most likely traveled route prediction task in terms of two measurements: recall@ n and accuracy. For example, denoting the ground truth route as $\mathbf{r}$, the experiments generate the most likely route $\hat{\mathbf{r}}$ using different existing methods. Since the generated route $\hat{\mathbf{r}}$ can be arbitrarily long, to give a fair comparison, the generated route $\hat{\mathbf{r}}$ is truncated by preserving the first $|\mathbf{r}|$ road segments, denoting the result as $\hat{\mathbf{r}}_t$. The measurement recall@ n may be expressed as the ratio of the length of correctly predicted road segments over the length of ground truth $\mathbf{r}$, and may be expressed as the following:

$$\text{recall}@n = \frac{|r \cap \hat{r}_t|}{|\hat{r}|} \quad (21)$$

The accuracy measurement may be the ratio of the length of correctly predicted road segments over the maximum value of the length of ground truth $\mathbf{r}$ and the length of predicted route $\hat{\mathbf{r}}$, and may be expressed as the following:

$$\text{accuracy} = \frac{|\mathbf{r} \cap \hat{\mathbf{r}}|}{\max(|\mathbf{r}|, |\hat{\mathbf{r}}|)}. \quad (22)$$

Experimental Results

[00103] As discussed above, in the experiments, the proposed DeepST method was evaluated against existing methods on two tasks: (1) route prediction (e.g., predicting the most likely route); and (2) route recovery from sparse trajectories. As discussed in further detail below, the proposed DeepST method can surpass the best existing method by almost 50% on the most likely route prediction task and up to about 15% on the route recovery task in terms of accuracy.

[00104] Table 4A shows the performance of the proposed DeepST method for the Chengdu dataset relative to other methods in terms of the $\text{recall}@n$ and accuracy measurements shown in Equations (21) and (22). Table 4B shows the performance of the proposed DeepST method for the Harbin dataset relative to other methods in terms of the $\text{recall}@n$ and accuracy measurements shown in Equations (21) and (22).

Table 4A: Overall Performance for Chengdu Dataset.

Method	DeepST	DeepST-C	CSSRNN	RNN	MMI	WSP
$\text{recall}@n$	0.637	0.626	0.577	0.409	0.318	0.431
accuracy	0.612	0.601	0.556	0.389	0.281	0.431

Table 4B: Overall Performance for Harbin Dataset.

Method	DeepST	DeepST-C	CSSRNN	RNN	MMI	WSP
$\text{recall}@n$	0.397	0.385	0.336	0.261	0.202	0.267
accuracy	0.374	0.366	0.313	0.172	0.154	0.267

[00105] As seen in Tables 4A and 4B, RNN outperforms MMI since RNN can capture the long-range dependency in the spatial transition patterns. Furthermore, both RNN and

MMI are worse than WSP in terms of recall@ n and accuracy. A reason for this may be because without considering the destinations, the RNN and MMI methods make identical predictions for all trips that start from the same initial road segment. Tables 4A and 4B also show that CSSRNN performs much better than RNN, MMI, and WSP since CSSRNN explicitly incorporates the influence of destinations by learning their distributed representations. This indicates that the destinations play an important role in the route decision. In Table 4A, it is seen that for the Chengdu dataset, the proposed DeepST method surpasses CSSRNN by about 10.4% for the recall@ n measurement and by about 10.1% for the accuracy measurement. In Table 4B, it is seen that for the Harbin dataset, the proposed DeepST method surpasses CSSRNN by about 18.2% for the recall@ n measurement and by about 19.5% for the accuracy measurement. A reason for the superiority of the proposed DeepST method over CSSRNN may be because the proposed DeepST method simultaneously models the spatial transition sequential property, the impact of destinations and real-time traffic. All methods show better performance on the Chengdu dataset than on the Harbin dataset. This could be because the mean length of trips in the Harbin dataset is much longer and the road network topological structure of Harbin is more complex as shown in FIGS. 5A and 5B. Thus, tasks on the Harbin dataset may be more challenging than tasks on the Chengdu dataset.

[00106] With regards to the effectiveness of the K -destination proxies, as seen in Tables 4A and 4B, even without considering the real-time traffic, the DeepST-C method outperforms the CSSRNN method, which learns destination representations separately. This result verifies the conjecture discussed above that learning representations for the destinations separately cannot effectively share the statistical strength across trips and also demonstrates the effectiveness of the K -destination proxies in the proposed DeepST method.

[00107] In the experiments, the impact of travel distance on the performance of different methods was investigated. In the experiments, the trips in the Chengdu and Harbin datasets were partitioned by their distances (e.g., measured in km) into the following eight buckets: [1,3), [3,5), [5,10), [10,15), [15,20), [20,25), [25,30), [30, ∞). The accuracy of different methods on the buckets was also determined (e.g., using Equation (22)). FIG. 8A is a diagram showing a plot 800 illustrating the accuracy of the proposed DeepST method relative to other methods for the Chengdu dataset, according to an implementation of the present disclosure. The horizontal axis of plot 800 denotes travel

distance ranges (in km), and the vertical axis of plot 800 denotes accuracy. FIG. 8B is a diagram showing a plot 801 illustrating the accuracy of the proposed DeepST method relative to other methods for the Harbin dataset, according to an implementation of the present disclosure. The horizontal axis of plot 801 denotes travel distance ranges (in km), and the vertical axis of plot 801 denotes accuracy. The travel distance ranges shown in FIGS. 8A and 8B correspond to the eight buckets discussed above.

[00108] As seen in FIGS. 8A and 8B, for short trips (e.g., trips less than 3 km), both the proposed DeepST method and CSSRNN show much better performance than the other methods. As the travel distance grows, the performance of all methods decreases. This is because as the travel distance increases, the number of possible routes between the origin and destination grows exponentially, and the task of predicting the most likely route from them becomes more difficult. Nevertheless, the proposed DeepST method is able to outperform the accuracy of other methods on all buckets. As the travel distance grows up to 10 km, the performance gap between the proposed DeepST method and the other methods becomes more evident; in particular, the proposed DeepST method surpasses the best existing method for the given distance bucket by almost 50% in terms of accuracy on the Chengdu dataset.

[00109] Experiments were also conducted to investigate route recovery using the proposed DeepST method. The proposed DeepST method can score the spatial transition likelihood of any given route, and thus can be used to boost any existing route recovery method. As discussed above, a route recovery method attempts to infer the most likely route between two GPS points in a sparse trajectory. In the experiments, the STRS method was used as a sparse route recovery method. Furthermore, the performance of STRS+ (where the spatial transition inference module of the STRS method was substituted with the proposed DeepST method) was also investigated to show how the proposed DeepST method can be used to enhance existing route recovery methods. As the travel time t during the two points is often available, the problem of route recovery can be expressed as $\arg\max_{\mathbf{r}} \mathbb{P}(\mathbf{r}|t)$. By Bayes' theorem, the following holds:

$$\arg\max_{\mathbf{r}} \mathbb{P}(\mathbf{r}|t) = \arg\max_{\mathbf{r}} \mathbb{P}(t|\mathbf{r})\mathbb{P}(\mathbf{r}), \mathbf{r} \in \{\text{candidate routes}\}. \quad (23)$$

[00110] The first term $\mathbb{P}(t|\mathbf{r})$ measures the likelihood of a route $\mathbf{r}$ with an observed travel time t , and is, namely, the temporal inference module of the STRS method. The

second term $\mathbb{P}(\mathbf{r})$ scores the spatial transition likelihood of a route $\mathbf{r}$, and is, namely, the spatial inference module of the STRS method. In the experiments, the spatial inference module $\mathbb{P}(\mathbf{r})$ of the STRS method was substituted with the proposed DeepST method to yield a new model (referred to as the STRS+ method). In the experiments, the route recovery accuracy of the STRS and STRS+ methods were compared to determine whether the proposed DeepST method can enhance the STRS method. In these experiments, 10,000 trajectories were randomly selected from the Chengdu and Harbin datasets, and these trajectories were mapped into the respective road networks as the ground truth with an existing map-matching method discussed in P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in SIGSPATIAL, 2009. The trajectories were subsequently downsampled with different sampling rates and the underlying routes were inferred with the STRS and STRS+ methods.

[00111] Table 5A shows the route recovery accuracy of the STRS and STRS+ methods for the Chengdu dataset for different sampling rates. Table 5B shows the route recovery accuracy of the STRS and STRS+ methods for the Harbin dataset for different sampling rates. In both Table 5A and Table 5B, the accuracy can be determined using Equation (22). In Tables 5A and 5B, the increase in accuracy of the STRS+ method over the STRS method is denoted by δ , which is expressed as a percentage. Furthermore, FIG. 9 is a diagram showing a plot 900 illustrating an example route recovery accuracy of the STRS and STRS+ methods relative to sampling rates for the Harbin dataset, according to an implementation of the present disclosure. The horizontal axis of plot 900 denotes sampling rate (measured in minutes), and the vertical axis of plot 900 denotes accuracy (e.g., determined using Equation (22)).

Table 5A: Route Recovery Accuracy vs. Sampling Rate for Chengdu Dataset.

Sampling Rate (mins)	1	2	3	4	5	6	7	8	9
STRS	0.98	0.96	0.93	0.90	0.86	0.82	0.79	0.73	0.69
STRS+	0.99	0.98	0.97	0.95	0.93	0.91	0.88	0.83	0.81
δ (%)	1.02	2.08	4.30	5.56	8.14	11.0	11.4	13.7	15.9

Table 5B: Route Recovery Accuracy vs. Sampling Rate for Harbin Dataset.

Sampling Rate (mins)	1	2	3	4	5	6	7	8	9
STRS	0.98	0.95	0.93	0.89	0.85	0.82	0.76	0.70	0.65
STRS+	0.98	0.96	0.95	0.92	0.89	0.86	0.82	0.77	0.75
δ (%)	0.00	1.05	2.15	3.37	4.71	4.88	7.89	10.0	15.4

[00112] As seen in Tables 5A and 5B, the superiority of the STRS+ method over the STRS method becomes more pronounced as sampling rate increases and the trajectory becomes sparser, and the need for route recovery is more critical when the trajectory is sparse. As the δ row shows, as the sampling rate grows up to 9 minutes, the STRS+ method outperforms the STRS method by about 15% for both datasets in terms of accuracy.

[00113] Experiments were also conducted to investigate the scalability of the proposed DeepST method. FIG. 10 is a diagram showing a plot 1000 illustrating an example variation of training time with training data size, according to an implementation of the present disclosure. The horizontal axis of plot 1000 denotes training data size (in millions), and the vertical axis of plot 1000 denotes training time (in hours). The plot 1000 was obtained by training the proposed DeepST method using varying training data sizes from the Harbin dataset. As seen in FIG. 10, the training time grows linearly with the size of the training dataset. A similar behavior was observed for the Chengdu dataset.

[00114] Experiments were also conducted to investigate parameter sensitivity. As discussed above, the proposed DeepST method proposes the joint learning of K -destination proxies to effectively share statistical strength across trips, rather than treating each destination separately. The K -destination proxies can have an impact on the performance of the proposed DeepST method, and experiments were conducted to investigate the impact of K on the performance of the proposed DeepST method. Table 6, shown below, illustrates the impact of K on the performance of the proposed DeepST method for the Harbin dataset, as measured using the recall@ n and accuracy measurements presented in Equations (21) and (22).

Table 6: Sensitivity of DeepST to K for Harbin Dataset.

K	500	1000	1500	2000	2500	3000
recall@ n	0.362	0.397	0.398	0.386	0.383	0.375
accuracy	0.339	0.374	0.373	0.368	0.362	0.353

[00115] As seen in Table 6, the performance of the proposed DeepST method significantly improves when K increases from 500 to 1000. This may be because a small K may not be able to provide sufficient number of proxies to guide the spatial transition of vehicles. Both recall@ n and accuracy decrease when increasing K to 2000 and beyond. This may be because with a large K , no sufficient number of trips is allocated to a proxy, and thus different proxies cannot effectively share the desired statistical strength.

[00116] Therefore, the proposed DeepST method can predict the most likely traveling route on the road network between two given locations. proposed DeepST method unifies three key explanatory factors – past traveled route, destination and real-time traffic – for the spatial transition modeling. The proposed DeepST method achieves this by explaining the generation of next route conditioned on the representations of the three explanatory factors in a principled way. For example, the past traveled route may be compressed with a RNN, and thus can account for long-range dependencies. To enable effectively sharing statistical strength across trips, an adjoint generative process is used to learn representations of K -destination proxies rather than learning the destination representations separately. The introduction of the latent variable $\mathbf{c}$ incorporates the impact of real-time traffic by inferring its posterior distribution. Lastly, an efficient inference algorithm is developed within the VAEs framework to scale to large-scale datasets. The experiments conducted on two real-world large-scale trajectory datasets to demonstrate the superiority of proposed DeepST method existing methods on two tasks: the most likely route prediction; and route recovery from sparse trajectories. As an example, on the Chengdu trajectory dataset, proposed DeepST method surpasses the best existing method by almost 50% on the most likely route prediction task and up to 15% on the route recovery task in terms of accuracy.

[00117] FIG. 11 is a diagram showing an example system 1100 for determining a route between an origin and a destination, according to an implementation of the present disclosure. As an example, the system 1100 may be used to predict the most likely route between an origin and a destination and score the likelihood of any route being traveled

by conditioning the determination of the most likely traveled route on a past traveled route (e.g., a sequence of road segments already traveled in the trip), the representation of a destination, and the representation of real-time traffic.

[00118] The system 1100 includes probe vehicles 1102A to 1102N (collectively referred to as probe vehicles 1102) spatially distributed across a road network (e.g., the road networks 100, 500, 501 shown in FIGS. 1A, 5A, 5B, respectively). Each of the probe vehicles 1102A to 1102N may include or be associated with respective sensors 1104A to 1104N (collectively referred to as sensors 1104) and respective applications 1106A to 1106N (collectively referred to as applications 1106). The probe vehicles 1102 having the sensors 1104 and applications 1106 can provide the sequence of road segments already traveled in the trip (e.g., the past traveled route) and the real-time traffic data for the proposed DeepST method.

[00119] The probe vehicles 1102 can be any type of motorized or non-motorized modes of transportation, examples being bicycles, motorcycles, cars, trucks, vans, mopeds, etc. In some implementations, the probe vehicles 1102 may include Global Positioning System (GPS) receivers to obtain geographic coordinates from satellites 1108 for determining current location and time associated with the probe vehicles 1102. Further, the location can be determined by an Assisted Global Positioning (A-GPS), Cell of Origin, a wireless signal triangulation system, or other location extrapolation technologies.

[00120] The sensors 1104 may enable determination, for example, position, destination, speed, type and identification, or any combination thereof, for the probe vehicles 1102. Additionally or optionally, the sensors 1104 may enable determination the real-time status situation in one or more road segments, such as traffic conditions or weather. The sensors 1104 may be embedded in user devices located in the probe vehicles 1102 or may be embedded in the probe vehicles 1102 themselves. In some implementations, the sensors 1104 may include one or more of the following example sensors: a global positioning sensor for gathering location data, such as a Global Navigation Satellite System (GNSS) sensor; Light Detection And Ranging (LIDAR) for gathering distance data; a network detection sensor for detecting wireless signals or receivers for different short-range communications (e.g., Bluetooth, Wi-Fi, Li-Fi, Near Field Communication (NFC) etc.); temporal information sensors; or a camera/imaging sensor for gathering image data (e.g., the camera sensors may automatically capture traffic flow information and/or traffic light information). Additionally or optionally, the

sensors 1104 may include one or more of the following example sensors: light sensors; oriental sensors augmented with height sensor and acceleration sensor (e.g., an accelerometer can measure acceleration and can be used to determine orientation of the probe vehicles 1102); tilt sensors (e.g., gyroscopes) to detect the degree of incline or decline of the vehicle along a path of travel; an electronic compass to detect a compass direction, moisture sensors, and pressure sensors. Additionally or optionally, the sensors 1104 may be distributed around the perimeter of the probe vehicles 1102 and may detect the relative distance of the probe vehicles 1102 from road segments, the presence of other vehicles, pedestrians, traffic lights, potholes, and any other objects, or a combination thereof. Additionally or optionally, the sensors 1104 may detect weather data, road condition, traffic information, or a combination thereof. In some instances, the sensors 1104 may provide in-vehicle navigation services, where one or more location-based services may be provided to the probe vehicles 1102 or user devices located in the probe vehicles 1102.

[00121] The applications 1106 may include one or more of the following example applications: location-based service application; navigation application; content provisioning application; camera/imaging application; media player application; social networking application; calendar applications; multimedia application; and the like. The applications 1106 may be installed within the probe vehicles 1102 or user devices located in the probe vehicles 1102. In some instances, a location-based service application installed in probe vehicles 1102 enables a computation platform 1110 to determine, for example, position, destination, heading, speed, context, identification, type, or any combination thereof, for one or more of the probe vehicles 1102. In some instances, applications 1106 enable the computation platform 1110 to process location information, traffic information, and sensor information to determine (e.g., predict) a route (e.g., the most likely route) between the origin and the destination.

[00122] The computation platform 1110 may be a platform with multiple interconnected components. In some implementations, the computation platform 1110 may perform one or more operations associated with determining a route between an origin and a destination (e.g., determining the most likely route between the origin and the destination). For example, the computation platform 1110 may include one or more servers, intelligent networking devices, computing devices, components and corresponding software for executing the proposed DeepST method, which determines

the most likely route between an origin and a destination and scores the likelihood of any route being traveled by conditioning the determination of the most likely traveled route on a past traveled route, the representation of a destination, and the representation of real-time traffic. In some instances, the computation platform 1110 can provide a timely notification to at least one device based on the results of the proposed DeepST method.

[00123] The probe vehicles 1102 are communicatively coupled to the computation platform 1110 via a communication network 1112. The communication network 1112 may include one or more networks such as a data network, a wireless network, a telephony network, or any combination thereof. It is contemplated that the data network may be any local area network (LAN), metropolitan area network (MAN), wide area network (WAN), a public data network (e.g., the Internet), short range wireless network, or any other suitable packet-switched network, such as a commercially owned, proprietary packet-switched network, e.g., a proprietary cable or fiber-optic network, and the like, or any combination thereof. In some instances, the wireless network may be, for example, a cellular communication network and may employ various technologies including enhanced data rates for global evolution (EDGE), general packet radio service (GPRS), global system for mobile communications (GSM), Internet protocol multimedia subsystem (IMS), universal mobile telecommunications system (UMTS), etc., as well as any other suitable wireless medium, e.g., worldwide interoperability for microwave access (WiMAX), Long Term Evolution (LTE) networks, code division multiple access (CDMA), wideband code division multiple access (WCDMA), wireless fidelity (Wi-Fi), wireless LAN (WLAN), Bluetooth®, Internet Protocol (IP) data casting, satellite, mobile ad-hoc network (MANET), vehicle controller area network (CAN bus), and the like, or any combination thereof.

[00124] The system 1100 may include a database 1114, which may be communicatively coupled to the computation platform 1110. The database 1114 may store data associated with the road network (e.g., the locations of the road segments in the road network, the locations of the probe vehicles 1102, training data for the proposed DeepST method, and the like. In general, the database 1114 may store any data proposed or generated by the proposed DeepST method.

[00125] The system 1100 may include a services platform 1116 including one or more services 1118. The services platform 1116 may be communicatively coupled to the computation platform 1110 via the communication network 1112. The services platform

1116 may include any type of service provided to a user. By way of example, the one or more services 1118 may include mapping services, navigation services, travel planning services, route calculation services, notification services, social networking services, content (e.g., audio, video, images, etc.) provisioning services, application services, storage services, contextual information determination services, location-based services, information (e.g., weather, news, etc.) based services, etc., or any combination thereof. In some instances, the services platform 1116 may be embodied in a user equipment (UE) that includes a user interface that accepts an input from a user. In some implementations, the user may provide the origin and destination information for the proposed DeepST method via the services platform 1116. In some instances, the UE that embodies the services platform 1116 may include any type of a mobile terminal, wireless terminal, fixed terminal, or portable terminal. Examples of the UE may include a mobile handset, a wireless communication device, a station, a unit, a device, a multimedia computer, a multimedia tablet, an Internet node, a communicator, a desktop computer, a laptop computer, a notebook computer, a netbook computer, a tablet computer, a Personal Communication System (PCS) device, a personal navigation device, a Personal Digital Assistant (PDA), a digital camera/camcorder, an infotainment system, a dashboard computer, a television device, or any combination thereof, including the accessories and peripherals of these devices, or any combination thereof. In some instances, the computation platform 1110 may provide a notification to the UE that is indicative of the results of the proposed DeepST method.

[00126] FIG. 12 is a block diagram showing an example computing platform 1200, according to an implementation of the present disclosure. The computing platform 1200 in FIG. 12 may be an example implementation of the computing platform 1110 shown in FIG. 11. As shown in FIG. 12, the example computing platform 1200 includes an interface 1202, a processor 1204, a memory 1206, and a power unit 1208. A computing platform may include additional or different components, and the computing platform 1200 may be configured to operate as described with respect to the examples above. In some implementations, the interface 1202, processor 1204, memory 1206, and power unit 1208 of a computing platform are housed together in a common housing or other assembly. In some implementations, one or more of the components of a computing platform can be housed separately, for example, in a separate housing or other assembly.

[00127] The example interface 1202 can communicate (receive, transmit, or both) wireless signals. For example, the interface 1202 may be configured to communicate radio frequency (RF) signals formatted according to a wireless communication standard (e.g., Wi-Fi, 4G, 5G, Bluetooth, etc.). In some implementations, the example interface 1202 includes a radio subsystem and a baseband subsystem. The radio subsystem may include, for example, one or more antennas and radio frequency circuitry. The radio subsystem can be configured to communicate radio frequency wireless signals on the wireless communication channels. As an example, the radio subsystem may include a radio chip, an RF front end, and one or more antennas. The baseband subsystem may include, for example, digital electronics configured to process digital baseband data. In some cases, the baseband subsystem may include a digital signal processor (DSP) device or another type of processor device. In some cases, the baseband system includes digital processing logic to operate the radio subsystem, to communicate wireless network traffic through the radio subsystem or to perform other types of processes.

[00128] The example processor 1204 can execute instructions, for example, to generate output data based on data inputs. The instructions can include programs, codes, scripts, modules, or other types of data stored in memory 1206. Additionally or alternatively, the instructions can be encoded as pre-programmed or re-programmable logic circuits, logic gates, or other types of hardware or firmware components or modules. The processor 1204 may be or include a general-purpose microprocessor, as a specialized co-processor or another type of data processing apparatus. In some cases, the processor 1204 performs high level operation of the computing platform 1200. For example, the processor 1204 may be configured to execute or interpret software, scripts, programs, functions, executables, or other instructions stored in the memory 1206. In some implementations, the processor 1204 may be included in the interface 1202 or another component of the computing platform 1200.

[00129] The example memory 1206 may include computer-readable storage media, for example, a volatile memory device, a non-volatile memory device, or both. The memory 1206 may include one or more read-only memory devices, random-access memory devices, buffer memory devices, or a combination of these and other types of memory devices. In some instances, one or more components of the memory can be integrated or otherwise associated with another component of the computing platform 1200. The memory 1206 may store instructions that are executable by the processor 1204. For

example, the instructions may include instructions to perform one or more of the operations of the proposed DeepST method (e.g., the example process 1300 shown in FIG. 13).

[00130] The example power unit 1208 provides power to the other components of the computing platform 1200. For example, the other components may operate based on electrical power provided by the power unit 1208 through a voltage bus or other connection. In some implementations, the power unit 1208 includes a battery or a battery system, for example, a rechargeable battery. In some implementations, the power unit 1208 includes an adapter (e.g., an AC adapter) that receives an external power signal (from an external source) and converts the external power signal to an internal power signal conditioned for a component of the computing platform 1200. The power unit 1208 may include other components or operate in another manner.

[00131] FIG. 13 is a flowchart showing an example process 1300 for determining a route between an origin and a destination, according to an implementation of the present disclosure. As an example, the process 1300 may implement one or more aspects of the proposed DeepST method and may be used to predict the most likely route between an origin and a destination and score the likelihood of any route being traveled by conditioning the determination of the most likely traveled route on a past traveled route (e.g., a sequence of road segments already traveled in the trip), the representation of a destination, and the representation of real-time traffic. The process 1300 may include additional or different operations, and the operations shown in FIG. 13 may be performed in the order shown or in another order. In some cases, one or more of the operations shown in FIG. 13 are implemented as processes that include multiple operations, sub-processes for other types of routines. In some cases, operations can be combined, performed in another order, performed in parallel, iterated or otherwise repeated or performed in another manner. The process 1300 may be performed by the system 1100 in FIG. 11 (e.g., the computation platform 1110) or the computation platform 1200 in FIG. 12, or by another type of device.

[00132] At 1302, input data may be obtained from a user. The input data may include an indication of an origin and a destination for a trip in a road network (e.g., the road networks 100, 500, 501 shown in FIGS. 1A, 5A, 5B, respectively). At 1304, real-time traffic data (e.g., input **T.C.**, indicated in FIG. 4 as reference numeral 405) may be obtained from

a plurality of probe vehicles (e.g., probe vehicles 1102 in FIG. 11) spatially distributed across the road network.

[00133] At 1306, a past traveled route (e.g., vector $\mathbf{r}_{1:i}$ shown in FIG. 2) may be obtained from at least one of the plurality of probe vehicles. Each probe vehicle may have a respective past traveled route, and the past traveled route may indicate a sequence of road segments traveled in the trip. At 1308, a first neural network may be used to determine a representation of the past traveled route (e.g., see Equation (9) showing an example of the representation of the past traveled route).

[00134] At 1310, a second neural network may be used to determine a representation of the real-time traffic data (e.g., see latent variable 204 shown in FIGS. 2, 3, 4 as an example of the representation of the real-time traffic data). At 1312, an adjoint generative process may be used to determine a representation of the destination (e.g., see destination proxy 206 for destination coordinate 206' in FIG. 4). At 1314, a next road segment (e.g., road segment 202) may be determined based on the representation of the past traveled route, the representation of the real-time traffic data, and the representation of the destination.

[00135] Some of the subject matter and operations described in this specification can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Some of the subject matter described in this specification can be implemented as one or more computer programs, i.e., one or more modules of computer program instructions, encoded on a computer storage medium for execution by, or to control the operation of, data-processing apparatus. A computer storage medium can be, or can be included in, a computer-readable storage device, a computer-readable storage substrate, a random or serial access memory array or device, or a combination of one or more of them. Moreover, while a computer storage medium is not a propagated signal, a computer storage medium can be a source or destination of computer program instructions encoded in an artificially generated propagated signal. The computer storage medium can also be, or be included in, one or more separate physical components or media (e.g., multiple CDs, disks, or other storage devices).

[00136] Some of the operations described in this specification can be implemented as operations performed by a data processing apparatus on data stored on one or more computer-readable storage devices or received from other sources.

[00137] The term “data processing apparatus” encompasses all kinds of apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, a system on a chip, or multiple ones, or combinations, of the foregoing. The apparatus can include special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit). The apparatus can also include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, a cross-platform runtime environment, a virtual machine, or a combination of one or more of them.

[00138] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, declarative or procedural languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, object, or other unit suitable for use in a computing environment. A computer program may, but need not, correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[00139] Some of the processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform actions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

[00140] To provide for interaction with a user, operations can be implemented on a computer having a display device (e.g., a monitor, or another type of display device) for

displaying information to the user and a keyboard and a pointing device (e.g., a mouse, a trackball, a tablet, a touch sensitive screen, or another type of pointing device) by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input. In addition, a computer can interact with a user by sending documents to and receiving documents from a device that is used by the user; for example, by sending web pages to a web browser on a user's client device in response to requests received from the web browser.

[00141] In a general aspect, a route is determined between an origin and a destination. As an example, the most likely route between the origin and the destination is predicted.

[00142] In a first example, a method for determining a route may include: obtaining input data from a user, the input data including an indication of an origin and a destination for a trip in a road network; obtaining real-time traffic data from a plurality of probe vehicles spatially distributed across the road network; and obtaining a past traveled route from at least one of the plurality of probe vehicles, each probe vehicle having a respective past traveled route, the past traveled route indicating a sequence of road segments traveled in the trip. The method may further include determining, using a first neural network, a representation of the past traveled route; determining, using a second neural network, a representation of the real-time traffic data; determining, using an adjoint generative process, a representation of the destination; and determining a next road segment for the trip based on the representation of the past traveled route, the representation of the real-time traffic data, and the representation of the destination.

[00143] Implementations of the first example may include one or more of the following features. The first neural network (e.g., encoder 408) may include one or more recurrent neural networks. The second neural network (e.g., encoder 406) may include a convolutional neural network and multi-layer perceptrons. Determining, using the second neural network, the representation of the real-time traffic data may include: providing the real-time traffic data (e.g., input **T.C**, indicated in FIG. 4 as reference numeral 405) as an input to the second neural network; and generating a first latent variable (e.g., latent variable 204 shown in FIGS. 2, 3, 4) as an output of the second neural network, the first latent variable being the representation of the real-time traffic data.

[00144] The first example may further include inferring, using a first variational auto-encoder, a first posterior distribution conditioned on the real-time traffic data (e.g., posterior distribution $q(\mathbf{c}|C)$ shown in FIG. 4). Generating the first latent variable as the output of the second neural network may include selecting a first random variable from the first posterior distribution conditioned on the real-time traffic data, the first random variable being the first latent variable. Determining, using the adjoint generative process, the representation of the destination may include: simultaneously determining, based on a second latent variable (e.g., latent variable 302 shown in FIGS. 3 and 4), a plurality of destination proxies in the road network; and allocating the trip to a selected destination proxy of the plurality of destination proxies, the selected destination proxy (e.g., destination proxy 206 shown in FIGS. 3 and 4) being the representation of the destination. The first example may further include generating the second latent variable based on a second posterior distribution conditioned on spatial coordinates of the destination (e.g., posterior distribution $q(\boldsymbol{\pi}|\mathbf{x})$ shown in FIG. 4). The first example may further include inferring, using a second variational auto-encoder, the second posterior distribution conditioned on the spatial coordinates of the destination (e.g., coordinates 206' shown in FIG. 4). Generating the second latent variable based on the second posterior distribution conditioned on the spatial coordinates of the destination may include selecting a second random variable from the second posterior distribution conditioned on the spatial coordinates of the destination, the second random variable being the second latent variable.

[00145] In a second example, a non-transitory computer-readable medium stores instructions that are operable when executed by data processing apparatus to perform one or more operations of the first example. In a third example, a system includes a memory and at least one processor communicatively coupled to the memory and configured to perform operations of the first example.

[00146] While this specification contains many details, these should not be understood as limitations on the scope of what may be claimed, but rather as descriptions of features specific to particular examples. Certain features that are described in this specification or shown in the drawings in the context of separate implementations can also be combined. Conversely, various features that are described or shown in the context of a single implementation can also be implemented in multiple implementations separately or in any suitable subcombination.

[00147] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system components in the implementations described above should not be understood as requiring such separation in all implementations, and it should be understood that the described program components and systems can generally be integrated together in a single product or packaged into multiple products.

[00148] A number of implementations have been described. Nevertheless, it will be understood that various modifications can be made. Accordingly, other implementations are within the scope of the following claims.

CLAIMS

What is claimed is:

1. A method for determining a route, the method comprising:
 - obtaining input data from a user, the input data comprising an indication of an origin and a destination for a trip in a road network;
 - obtaining real-time traffic data from a plurality of probe vehicles spatially distributed across the road network;
 - obtaining a past traveled route from at least one of the plurality of probe vehicles, each probe vehicle having a respective past traveled route, the past traveled route indicating a sequence of road segments traveled in the trip;
 - determining, using a first neural network, a representation of the past traveled route;
 - determining, using a second neural network, a representation of the real-time traffic data;
 - determining, using an adjoint generative process, a representation of the destination; and
 - determining a next road segment for the trip based on the representation of the past traveled route, the representation of the real-time traffic data, and the representation of the destination.
2. The method of claim 1, wherein the first neural network comprises one or more recurrent neural networks.
3. The method of claim 1, wherein the second neural network comprises a convolutional neural network and multi-layer perceptrons.
4. The method of any one of claims 1 to 3, wherein determining, using the second neural network, the representation of the real-time traffic data comprises:
 - providing the real-time traffic data as an input to the second neural network; and
 - generating a first latent variable as an output of the second neural network, the first latent variable being the representation of the real-time traffic data.
5. The method of claim 4, further comprising inferring, using a first variational auto-encoder, a first posterior distribution conditioned on the real-time traffic data.

6. The method of claim 5, wherein generating the first latent variable as the output of the second neural network comprises:
 - selecting a first random variable from the first posterior distribution conditioned on the real-time traffic data, the first random variable being the first latent variable.
7. The method of claim 6, wherein determining, using the adjoint generative process, the representation of the destination comprises:
 - simultaneously determining, based on a second latent variable, a plurality of destination proxies in the road network; and
 - allocating the trip to a selected destination proxy of the plurality of destination proxies, the selected destination proxy being the representation of the destination.
8. The method of claim 7, further comprising generating the second latent variable based on a second posterior distribution conditioned on spatial coordinates of the destination.
9. The method of claim 8, further comprising inferring, using a second variational auto-encoder, the second posterior distribution conditioned on the spatial coordinates of the destination.
10. The method of claim 8, wherein generating the second latent variable based on the second posterior distribution conditioned on the spatial coordinates of the destination comprises:
 - selecting a second random variable from the second posterior distribution conditioned on the spatial coordinates of the destination, the second random variable being the second latent variable.
11. A system for determining a route, the system comprising:
 - a memory; and
 - at least one processor communicatively coupled to the memory and configured to perform operations comprising:
 - obtaining input data from a user, the input data comprising an indication of an origin and a destination for a trip in a road network;
 - obtaining real-time traffic data from a plurality of probe vehicles spatially distributed across the road network;

obtaining a past traveled route from at least one of the plurality of probe vehicles, each probe vehicle having a respective past traveled route, the past traveled route indicating a sequence of road segments traveled in the trip;

determining, using a first neural network, a representation of the past traveled route;

determining, using a second neural network, a representation of the real-time traffic data;

determining, using an adjoint generative process, a representation of the destination; and

determining a next road segment for the trip based on the representation of the past traveled route, the representation of the real-time traffic data, and the representation of the destination.

12. The system of claim 11, wherein the first neural network comprises one or more recurrent neural networks.

13. The system of claim 11, wherein the second neural network comprises a convolutional neural network and multi-layer perceptrons.

14. The system of any one of claims 11 to 13, wherein determining, using the second neural network, the representation of the real-time traffic data comprises:

providing the real-time traffic data as an input to the second neural network; and

generating a first latent variable as an output of the second neural network, the first latent variable being the representation of the real-time traffic data.

15. The system of claim 14, the operations further comprising inferring, using a first variational auto-encoder, a first posterior distribution conditioned on the real-time traffic data.

16. The system of claim 15, wherein generating the first latent variable as the output of the second neural network comprises:

selecting a first random variable from the first posterior distribution conditioned on the real-time traffic data, the first random variable being the first latent variable.

17. The system of claim 16, wherein determining, using the adjoint generative process, the representation of the destination comprises:

simultaneously determining, based on a second latent variable, a plurality of destination proxies in the road network; and

allocating the trip to a selected destination proxy of the plurality of destination proxies, the selected destination proxy being the representation of the destination.

18. The system of claim 17, the operations further comprising generating the second latent variable based on a second posterior distribution conditioned on spatial coordinates of the destination.

19. The system of claim 18, the operations further comprising inferring, using a second variational auto-encoder, the second posterior distribution conditioned on the spatial coordinates of the destination.

20. The system of claim 18, wherein generating the second latent variable based on the second posterior distribution conditioned on the spatial coordinates of the destination comprises:

selecting a second random variable from the second posterior distribution conditioned on the spatial coordinates of the destination, the second random variable being the second latent variable.

21. A non-transitory computer-readable medium comprising instructions that are operable, when executed by a data processing apparatus, to perform operations comprising:

obtaining input data from a user, the input data comprising an indication of an origin and a destination for a trip in a road network;

obtaining real-time traffic data from a plurality of probe vehicles spatially distributed across the road network;

obtaining a past traveled route from at least one of the plurality of probe vehicles, each probe vehicle having a respective past traveled route, the past traveled route indicating a sequence of road segments traveled in the trip;

determining, using a first neural network, a representation of the past traveled route;

determining, using a second neural network, a representation of the real-time traffic data;

determining, using an adjoint generative process, a representation of the destination; and

determining a next road segment for the trip based on the representation of the past traveled route, the representation of the real-time traffic data, and the representation of the destination.