Factorized Deep Generative Models for Trajectory Generation with Spatiotemporal-Validity Constraints

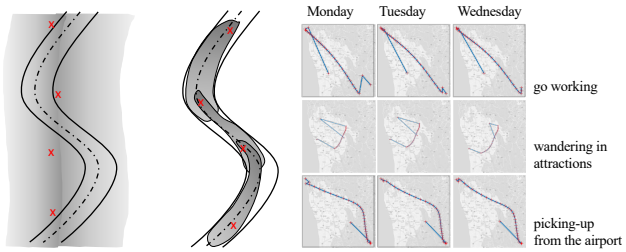
Liming Zhang[1], **Liang Zhao**[1], **Dieter Pfoser**[1]
[1] George Mason University
4400 University Dr,
Fairfax, Virginia 22030
{lzhang22, lzhao9, dpfoser}@gmu.edu

Abstract

Trajectory data generation is an important domain that characterizes the generative process of mobility data. Traditional methods heavily rely on predefined heuristics and distributions and are weak in learning unknown mechanisms. Inspired by the success of deep generative neural networks for images and texts, a fast-developing research topic is deep generative models for trajectory data which can learn expressively explanatory models for sophisticated latent patterns. This is a nascent yet promising domain for many applications. We first propose novel deep generative models factorizing time-variant and time-invariant latent variables that characterize global and local semantics, respectively. We then develop new inference strategies based on variational inference and constrained optimization to encapsulate the spatiotemporal validity. New deep neural network architectures have been developed to implement the inference and generation models with newly-generalized latent variable priors. The proposed methods achieved significant improvements in quantitative and qualitative evaluations in extensive experiments.

Introduction

Recent advances in Global Positional System (GPS), traffic surveillance cameras, unmanned aerial vehicles (UAV), and Radio-frequency identification (RFID) sensors embedded in devices and cities have enabled an unprecedented increase in the amount of location records of moving objects on earth, such as taxi GPS traces and tourist check-ins. Such a series of temporally-ordered location points of an object represents a trajectory. Mining trajectory data is important to a broad range of applications such as location-based social networks, intelligent transportation systems, and urban computing (Zheng 2015). Trajectory data mining involves two important tasks: 1) Trajectory representation learning, which aims at encoding trajectory data into (low-dimensional) vector space; and 2) Trajectory generation, which reversely aims at constructing a trajectory-structured data from low-dimensional space containing the trajectory generation rules or distribution. Different from trajectory representation learning which benefits the downstream tasks such as discriminative learning and clustering, trajectory generation focuses on learning and interpreting the underlying distribution and mechanism of the trajectory generative process, which is crucial for tasks such as human mobility simulation and privacy preservation of individual traces (Wang et al. 2020). In this paper, we focus on the topic of trajectory data generation.



(a) Without constraint (b) With constraint (c) Trajectory patterns

Figure 1: (a) red crosses are points of a trajectory, while gray zone are the probability density envelop that a deep learning model without constraint has; (b) the probability density envelop (grey zone) with roads as constraints; (c) for different time of a day, there could be multiple choices of routing for the same start and end point in a city. Start and end is determined by users which is static patterns, but routing choices are dynamic patterns.

Traditional models for trajectory generation typically rely on are hand-crafted rules or prescribed distributions (Giannotti et al. 2005; Pelekis et al. 2013) which require extensive human labors and domain knowledge yet still suffer from the bias and limited knowledge of the sophisticated mechanisms in trajectory generation. To address these issues, a fast-developing research topic is to extend deep generative models toward trajectory data, which enables to learn expressively generative models that could learn the sophisticated distributions based on a large amount of historical data. This is inspired by the success of deep generative neural networks in images and texts. Although deep learning has been widely used for other trajectory data mining tasks, such as trajectory representation learning and prediction, deep generative models for trajectory generation have not been well explored as indicated in a recent survey (Wang et al. 2020). This is a fast-growing domain, where existing few relevant works for trajectory generation are based on image-based Generative Adversarial Network (GAN) models (Ouyang et al. 2018; Smolyak et al. 2020) and sequential

Variational Autoencoder (VAE) (Huang et al. 2019).

Despite the progress in this promising domain in recent years, there are still several important challenges yet to be addressed: **Challenge 1: the necessity and difficulty in factorizing semantic and spatiotemporal patterns in trajectory generative modeling.** Each trajectory typically comes with an underlying purpose, such as "go working", "wandering in attractions", or "picking-up people from the airport". This type of patterns is namely global semantic meaning that do not change across different location points inside the trajectory. In addition, trajectory naturally also comes with local patterns that characterize the information for each location inside it as well as their spatiotemporal dependencies. Explicitly differentiating them has not been well explored by the existing deep generative models, which has limited the model interpretability and generalizability. **Challenge 2: Difficulty in jointly ensuring spatiotemporal-validity of the generated trajectories.** A generated trajectory is reasonable only when it satisfies necessary geometrical, physical, social principles. For example, all the location points should be on the roads and the movement speed should be limited to a reasonable range. Although deep generative models are good at learning expressive distributions from data, the learned distributions are still smoothed distribution over observations. Therefore, it is difficult yet imperative to diminish the probabilistic density for the invalid patterns. **Challenge 3: More reasonable inductive bias upon the prior distributions is needed.** Existing deep trajectory generative models usually follow the conventional priors used in deep generative models, which is to assume the independency among the latent variables corresponding to different locations. This, however, may not be ideal for trajectory generation because of the inherent dependence between the consecutive location points. How to design a new prior that goes beyond the conventional priors (e.g., isotropic Gaussians) is preferable yet challenging for trajectory generation.

To address the above issues, we proposes a new framework of factorized deep generative models for trajectory generation with spatiotemporal-validity constraints. Through factorized latent variables, it speparates global semantics as well as local spatiotemporal semantic. Newly-generalized dependent priors for latent sequential variables are proposed contrast to conventional independent priors in sequential models. With a novel constrained optimization solution, it reduces the probability of generating invalid samples. Extensive experiments with ablation study and qualitative study showed the effectiveness of different latent variables and this constrained optimization.

Related Work

Trajectory Generation/Synthesis: This domain has a long history, where the representative methods include Oporto (Giannotti et al. 2005) based physical movement estimation, or Hermoupolis (Pelekis et al. 2013) based on urban points of interests. See (Wang et al. 2020) for a comprehensive survey. Such conventional methods are hard to replicate since it uses many ground features of a specific city, and requires extensive programming efforts and domain knowledge to implement. The current emerging trend for trajectory generation is to use deep generative models in a data-driven end-to-end fashion. Deep generative models for trajectory generation are not widely explored until now (Wang et al. 2020). One type of works converted trajectories to images first and applied GANs for generation tasks (Ouyang et al. 2018; Smolyak et al. 2020). Such an approach loses many aspects of information including time, speed, and directions. Another work (Huang et al. 2019) utilize vanilla variational autoencoder scenarios by generating a whole trajectory via variables from unit Gaussian, which cannot jointly encode time-variant and -invariant information (Kingma and Welling 2013). Deep generative models that can comprehensively take care of static and dynamic patterns in trajectory while ensuring the spatiotemporal validity are seriously under-explored yet imperative.

Spatially-valid constraints in trajectory: other studies on trajectories consider spatial-temporal-validity constraints, such as trajectory generation of vehicles (Choe et al. 2015), collisions avoidance (Mehdi, Choe, and Hovakimyan 2017), monitoring with turning constraints (Stephens et al. 2019), Trajectory tracking with velocity and heading rate constraints (Ren and Beard 2004), bounded zoning constraints (Jorris 2007). Such constraints are not trivial to be considered in deep generative models and raise a major obstacle to generate realistic trajectory by neural networks.

Disentangled and factorized deep generative models: Disentangled deep generative models are promising research topic recently, especially for applications on image data (Bang et al. 2019; Chen et al. 2018; Higgins et al. 2016; Kim and Mnih 2018). The notion of disentanglement and factorization is to separate out the underlying explanatory factors responsible for variations of data. The generative representation learned in this way have been relatively resilient to the complex variants involved (Bengio, Courville, and Vincent 2013), and can be used to enhance interpretability, generalizability, and robustness against adversarial attack (Bang et al. 2019). Additional inductive bias could be considered to further factorize by leveraging particular data properties, such as factorizing graph data into the node and edge patterns (Guo et al. 2020) and factorizing video data into the object and motion patterns (Li and Mandt 2018).

Spatiotemporal-valid Trajectory Generation

We first introduce the Bayesian network of the proposed generative model, followed by new model inference methods. Then spatiotemporal-validity constraints are described and induced to the training objective. Finally, the model architectures for the encoder and decoder are elaborated.

Generative model

First, we define a trajectory as a sequence of location points $\{s_1, s_2, \cdots, s_T\}$ at time points $1, 2, \cdots, T$. The proposed method focuses on a new generative process of trajectories, which factorizes the whole semantic meaning of a trajectory into two parts: 1) the global semantics of the whole trajectory as well as 2) local semantics that characterize the dependencies among the neighborhood. The global semantics cover the overall meaning of the trajectory including com-

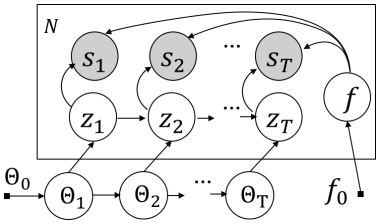


Figure 2: Plate notation of the proposed deep generative models for generating N trajectories. The index of the variables for each sample has been omitted for simplicity.

muting from suburb to downtown, wandering inside downtown, jogging in the trails, and so on. The local semantics cover spatiotemporal autoregressive patterns such as how the next location is dependent on the current locations. Moreover, instead of assuming that the local semantic variable $\{z_1, \cdots, z_T\}$ must be all from identical and independent prior distributions, here we allow their priors $\{\Theta_1, \cdots, \Theta_T\}$ to be conditional dependent with each other. When z_t follows Gaussian distribution, $\Theta_t = \{\mu_t, \sigma_t\}$ which are mean and standard deviation. These conditional distributions introduce more reasonable inductive bias and model expressiveness, by considering the dependencies among spatial neighbors. Specifically, as also shown in Figure 2, our model characterizes the following generative process:

- Draw a sequence of the priors $\{\Theta_1, \cdots, \Theta_T\}$ for local semantic status, based on conditional probability: $\Theta_t \sim p(\Theta_t|\Theta_{t-1})$, where $p(\Theta_0)$ is a predefined distribution such as a unit Gaussian.
- For each trajectory, draw a time-invariant variable f as the global semantic from $p_\theta(f)$ such as a unit Gaussian;
- For each trajectory, draw the local semantic variable z_1 for the first time point $t = 1$ from Θ_1.
  - For each time point $t \geq 1$, draw the underlying local semantic variable z_t with the conditional probability $z_t \sim p(z_t|z_{t-1}, \Theta_t)$.
  - For each t, draw the observed variable $s_t \sim p(s_t|z_t, f)$.

A natural question is what the connection of our work and previous works in trajectory generation is. It is found that if $\mu_{1:T}, \sigma_{1:T}$ are generated from f instead of $\mathbf{0}$, our models collapse to the baseline SVAE model in (Huang et al. 2019) and z_t, μ_t, σ_t would become internal parameters and states that have no significant meanings. We also provide such ablation study in later experiment sections to support the usage of dynamic factors $z_{1:T}$ with its priors $\mu_{1:T}, \sigma_{1:T}$.

Model Inference

Since the proposed generative model is intractable to infer, we proposed to solve it based on variational inference used for training variational autoencoder. This is achieved by first establish an approximate posterior $q_\phi(z_{1:T}, f|s_{1:T})$ in order to approximate the original distribution $p(z_{1:T}, f|s_{1:T})$, we investigate two possible choices of q_ϕ:

$$q_\phi(f, z_{1:T}|s_{1:T}) = \begin{cases} q_\phi(f|s_{1:T})q_\phi(z_{1:T}, \Theta_{1:T}|s_{1:T}) & (\textit{factorized}) \\ q_\phi(f|s_{1:T})q_\phi(z_{1:T}, \Theta_{1:T}|f, s_{1:T}) & (\textit{full}) \end{cases}$$

where the level of variance of $z_{1:T}$ could change depending on f in full model, for example, if most roads between home and work are highways, then there is almost no variance for routing choice, while the level of noise of $z_{1:T}$ in the factorized model do not depend on f. Such modeling could reflect on different road network layout of different cities.

Following β-VAE, the objective is as follows:

$$\min_{\psi,\phi} \mathcal{L}(p_\psi, q_\phi) = -\mathbb{E}_{q_\phi}\left[\log p_\psi(s_{1:T}|z_{1:T}, f, \Theta_{1:T})\right] \quad (1)$$
$$+\beta KL(q_\phi(z_{1:T}, f, \Theta_{1:T}|s_{1:T})||p_\psi(z_{1:T}, f, \Theta_{1:T}))$$

where β is hyper-parameter to control disentanglement in β-VAE, KL is shorten for KullbackLeibler divergence (Higgins et al. 2016), ψ and ϕ are sets of parameters in neural networks. They could be the parameters of a predefined distribution or deep generative neural networks. The first term is typically used for minimizing the reconstruction loss while the second one helps regularize the learned posterior close to the prior distributions. More specifically, the second term can be expanded as follows:

$$KL(q_\phi(z_{1:T}, f, \Theta_{1:T}|s_{1:T})||p_\psi(z_{1:T}, f, \Theta_{1:T})) \quad (2)$$
$$= KL\left(q_\phi(z_{1:T}, f, \Theta_{1:T}|s_{1:T})||p(f)\prod_{t=1}^{T} p(z_t|z_{t-1}, \Theta_t)p(\Theta_t|\Theta_{<t})\right)$$

where the prior $p(\Theta_0)$ follows an unit Gaussian distribution.

Spatiotemporal-validity constraints

Although the generative model learned by the Equation 2 could effectively characterize the underlying process of trajectory generation, the trajectories sampled from the learned generative model may not guarantee its validity and physical meaning in the real world. For example, the probabilistic density of the trajectory usually is continuous in the whole geographical space, leaving any location with nonzero probability to be passed by the trajectory. However, a trajectory needs to strictly follow spatial constraints. For example, the trajectory of vehicles needs to be on the roads, and hence its shapes and patterns should be constraints by the geometry of the roads. This requires to diminish the probabilistic density for the unfeasible trajectory patterns such as "out of road" or "constantly back and forth". Embedding in such an inductive bias can effectively increase the model generalizability and possibly strengthen the robustness against noise in training data due to the inaccuracy of the sensing data (e.g., those from GPS). The notion of spatial validity constraints can be leveraged our Equation 2, by the newly extended objective:

The central contribution is imposing spatial validity constrains in optimizing generic VAE loss function $\mathcal{L}$ that we have developed in Equation 2 as follows:

$$\min_{\psi,\phi} \mathcal{L}(p_\psi, q_\phi),\ s.t. \forall s_{1:T} \notin \mathcal{C} : p_\psi(s_{1:T}|z_{1:T}, f, \Theta_{1:T}) = 0 \quad (3)$$

where $\mathcal{C}$ denotes the set of all the trajectory patterns that satisfy the spatial validity constraint. The spatial validity constraint can be specified by the user based on the practical need. For example, if the constraint says all the

locations in the trajectory must be on the roads, then $\mathcal{C}_1 = \{[x_1, \cdots, x_T] | x_t \in \mathcal{R}\}$, where $\mathcal{R}$ denotes the spatial regions of the roads. The constraint could also be on the first-order phenomena such as speed limit, meaning the trajectory's moving speed must be physically feasible for the moving object. This could be denoted as $\mathcal{C}_2 = \{[x_1, \cdots, x_T] || \Delta x_t| \leq \mathcal{S}\}$, where $|\Delta x_t|$ denotes the object's speed at time t while $\mathcal{S}$ is the speed limit that this object's speed cannot exceed. Another pattern could be the turning angles between two consecutive segments in the trajectory, in many situations, it is unlikely to have many consecutive sharp turnings. To constrain this, we could have $\mathcal{C}_3 = \{[x_1, \cdots, x_T] | \sum_t cos(x_{t-1} - x_t, x_{t+1} - x_t) < \lambda\}$, where $cos(x_{t-1} - x_t, x_{t+1} - x_t)$ denotes the cosine similarity of the two vectors each of which is the movement in the 2D Euclidean surface. The spatial constraint $\mathcal{C}$ can also be composed of the logical combinations among multiple rules. For example, $\mathcal{C} = \mathcal{C}_2 \bigcap (\mathcal{C}_1 \bigcup \mathcal{C}_3)$.

Directly solving complex constrained problems using conventional ways such as Lagrangian has been demonstrated to be inefficient for deep neural networks. Here we extend a recent deep constrained optimization framework (Ma, Chen, and Xiao 2018) to handle our problem in Equation 3, which is reformulated as follows:

$$\tilde{\mathcal{L}}(p_\psi, q_\phi, \gamma) = \mathcal{L}(p_\psi, q_\phi) + \gamma \, [\int \mathbb{1}(g(z_{1:T}, f, \Theta_{1:T}) \notin \mathcal{C}) \cdot p_\psi(z_{1:T}, f, \Theta_{1:T}) \, \mathrm{d}z_{1:T} \, \mathrm{d}f \, \mathrm{d}\Theta_{1:T} \,]^{\frac{1}{2}} \quad (4)$$

where $\mathcal{C}$ is the set of validity functions, and $\mathbb{1}(\cdot)$ is an indicator function that output 1 if a generated trajectory is invalid, otherwise 0. We can reduce the integral term with a common approach of Monte Carlo Sampling in VAE (Kingma and Welling 2013). To allow gradient-flow over the regularization term, constraint functions in $\mathcal{C}$ must have gradients.

Deep neural network architectures

In this section, we introduce the detailed architectures for our proposed STG. Let a trajectory $s_{1:T}$ in our database $\mathcal{S}$, and $s_t = \langle x_1^t, x_2^t \rangle$ denotes the tth coordinate at time step t. The abstracted operations are shown in Figure 3 with sub-modules. Our encoder is $q_\phi(z_{1:T}, f, \Theta_{1:T} | s_{1:T}) = q(z_{1:T}\Theta_{1:T} | f, s_{1:T}) q(f | s_{1:T})$, which can be decomposed into two sub-encoders. 1) *time-invariant encoder* $q(f|s_{1:T})$, which consumes the whole sequence that capture the stochastic whole-sequence representation f detailed in upper left corner of Figure 3; 2) *time-variant encoder*, with factorized modeling alternative $q(z_{1:T}\Theta_{1:T}|s_{1:T})$ and full modeling alternative $q(z_{1:T}\Theta_{1:T}|f, s_{1:T})$, which takes each coordinates step by step to generates a stochastic posterior representation z_t for each step detailed in lower left corner of Figure 3. Blue lines are for full modeling alternative; 3) *joint-factor decoder for training* $p_\psi(s_{1:T}|z_{1:T}, f, \Theta_{1:T})$ during training phrase that combine sampled y and z_t to stochastically generate each coordinates s_t step by step, and minimize our training loss, which is detailed in right part of Figure 3. For *joint-factor decoder for synthesis*, joint-factor decoder relies only on the sequential network to generate prior means and variances of z_t first without the need to use

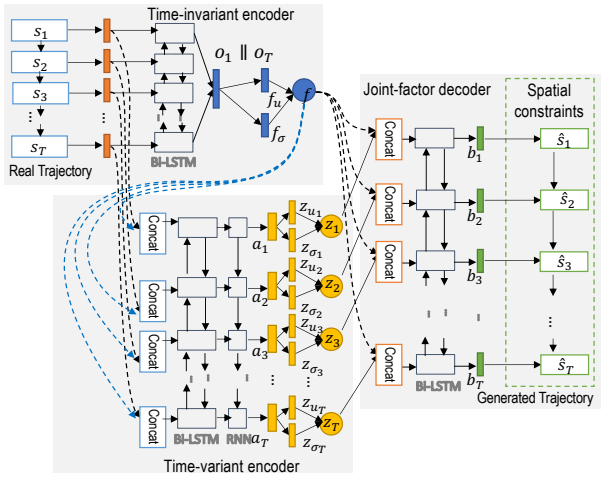


Figure 3: spatiotemporal-valid Trajectory Generative Architect.

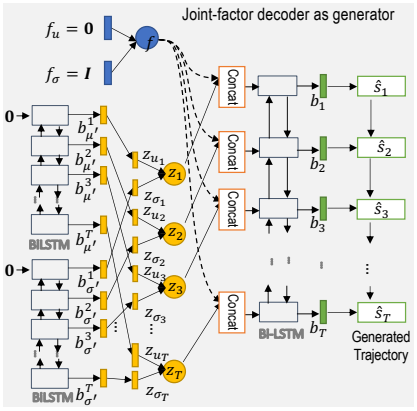


Figure 4: Joint-factor decoder for synthesis: a dynamic sequential generator to sample from sequential meta-priors, which make $z_{1:T}$ not i.i.d. samples but with recurrent dependence.

encoder. In the following, we use $MLP_*(\cdot)$ to denote different multi-layer perceptron, use $BiLSTM_*$ for different bi-directional LSTM (Graves and Schmidhuber 2005), use RNN_* for vanilla RNN networks. In general, all the operations are as follows:

Time-invariant encoder:

$$m_f^{t+1}; o^{t+1} = BiLSTM_f(MLP_s(s_t), m_f^t, o^t)$$

$$\mu_f = MLP_{\mu_f}(o^1 || o^T), \; \sigma_f = MLP_{\sigma_f}(o^1 || o^T), \; y \sim \mathcal{N}(\mu_f, \sigma_f)$$

Time-variant encoders:

$$\textit{(1) factorized:} \quad \tilde{m}_z^{t+1}; \tilde{a}^{t+1} = BiLSTM_z(MLP_s(s_t), \tilde{m}_z^t, \tilde{a}^t)$$

$$\textit{(2) full:} \quad \tilde{m}_z^{t+1}; \tilde{a}^{t+1} = BiLSTM_z(MLP_s(s_t) || y, \tilde{m}_z^t, \tilde{a}^t)$$

$$m_z^{t+1}; a^{t+1} = RNN_z(\tilde{m}_z^{t+1}, m_z^t; a^t)$$

$$\mu_{z_t} = MLP_{\mu_{z_t}}(a^t), \sigma_{z_t} = MLP_{\sigma_{z_t}}(a^t), \quad z_t \sim \mathcal{N}(\mu_{z_t}, \sigma_{z_t})$$

Joint-factor decoder for training:

$$m_s^{t+1}; b^{t+1} = BiLSTM_s(y || z_t, m_s^t, b^t), \; \widehat{s}_t = MLP_s(b^t)$$

Joint-factor decoder for synthesis:

$$m_\mu^{t+1}; b_\mu^{t+1} = BiLSTM_\mu(\mathbf{0}, m_\mu^t, b_\mu^t), \; \mu_t = MLP_\mu(b_\mu^t)$$

$$m_\sigma^{t+1}; b_\sigma^{t+1} = BiLSTM_\sigma(\mathbf{0}, m_\sigma^t, b_\sigma^t),\ \sigma_t = MLP_\sigma(b_\sigma^t)$$
$$z_t \sim \mathcal{N}(\mu_t, \sigma_t)$$
$$m_s^{t+1}; b^{t+1} = BiLSTM_s(y||z_t, m_s^t, b^t),\ \widehat{s}_t = MLP_b(b^t)$$

where $||$ is the concatenation operation of vectors, $\sim$ is the sampling operation which use the re-parameterization trick (Kingma and Welling 2013) to allow gradient back-propagation. m_* are different latent states vectors, and o_*, a_*, b_* are outputs for either $BiLSTM_*$ or RNN_* modules.

Experiments

In this section, both quantitative and qualitative results are reported to show the performance of STG with ablation study and comparisons to previous methods over four datasets. All experiments are conducted on a 64-bit machine with a NVIDIA 1080ti GPU.

Datasets

Real-world datasets The first dataset is collected from 442 taxi at Porto, Portugal describing a complete year (from 01/07/2013 to 30/06/2014) [1]. Data do not have time-stamps but with a fixed 15 second sampling interval. The second dataset is T-Drive data that collect continuous GPS points of 10,357 taxis in one week with real timestamps [2]. Preprocessing steps are used to clean the data, including Noise Filtering and Stay Point Detection (Zheng 2015). The third real-world dataset is human check-ins collected from a location-based website Gowalla [3], for which only the dense region at Dallas metropolitan from original global data is selected. All points are projected to a local geographic coordinate system in meters and convert to a 1000-meter unit.

Synthetic dataset We generated a synthetic dataset for 10,000 students living on a university campus from a location-based simulator (Kim et al. 2020). The student agents mimic real-world contacting and check-ins patterns based on predefined living and social preference settings in agent-based simulation.

We convert all datasets to Euclidean space using geographically projection based on the original earth projection system used in data. We split raw data with 0.9/0.1 ratios for training and testing subsets.

Constraints settings

Physics-induced constraint: Many constraints can be developed from physics law and engineering of a car. Since sampling time intervale e is small (15 seconds), a constraint is that if the average speed $\gamma_t = \frac{||s_t - s_{t-1}||_2}{e}$ is higher than a threshold $\bar{\gamma} = 60Km/h$, it is impossible for a car to make an sharp turn, so we can not observe a preceding angle η_t (in cosine value) smaller than a threshold $\bar{\eta}$. And, $\eta_t = \frac{(s_t - s_{t-1})\cdot(s_{t-1} - s_{t-2})}{||s_t - s_{t-1}||_2||s_{t-1} - s_{t-2}||_2}$. This regularization only impose penalties over a case that the angle is smaller than threshold $\eta_t > \bar{\eta}$ when a segment is larger than threshold $\gamma_t > \bar{\gamma}$ at the same time. We show such patterns in Porto dataset (red dashed region in second row) in Figure 5 which is formulated as follows:

$$\frac{\lambda}{N}\sum^J\sum_{t=2}^T c(t, s_{1:T}) = \frac{\lambda}{N}\sum^J\sum_{t=2}^T(\gamma_t - \bar{\gamma})_+(\eta_t - \bar{\eta})_+$$

where s_t is treated as a vector, and λ is a hyper-parameter because there is only one constraint function. Notice that total $T-2$ constraints for each trajectory are possible.

Behavior-induced constraints: Behavioral constraints come from behaviors which are abnormal to human, even though these behaviors did not validate the physics laws. For example, in two consecutive segments with a high GPS sampling rate like 5 second, it is abnormal to have two U-Turns (turn to opposite direction), in other words, two consecutive angles could not both be very sharp (less than 30 degrees). Such constraint can be shown in the Beijing dataset (red dashed regions in first row) in Figure 5. This regularization penalizes the case that first angle is smaller than a sharp threshold $\eta_t > \bar{\eta}$ when its preceding angle is also very sharp $\eta_{t-1} > \bar{\eta}$. Its formula is as follows:

$$\frac{\lambda}{N}\sum^J\sum_{t=3}^T c(s_{1:T}) = \frac{\lambda}{N}\sum^J\sum_{t=3}^T(\eta_t - \bar{\eta})_+(\eta_t - \bar{\eta})_+$$

where η is also the cosine value of angles. Notice that total $T-3$ constraints for each trajectory are possible.

Competing methods and Ablation study

Here, we introduce competing methods. Since some competing methods also belong to the ablation study, we also introduce the ablation study models simultaneously.

LSTM: a basic LSTM model that can take any start point as input and output a sequence of points.

IGMM-GAN: a GAN-based model with a new Dirichlet Process Mixture Model for latent noise input. It processes a trajectory as an image not a sequence (Smolyak et al. 2020).

SVAE-y: it used a static latent variable for a whole sequence, firstly developed in (Huang et al. 2019). This method can be also treated as an ablation study for our model that we use y without $z_{1:T}$.

SVAE-z: this ablation model uses only $z_{1:T}$ without y.

Disentangled SVAE (DSVAE): this is for the full model with both y and $z_{1:T}$ and each z_t is dependent on y.

Factorized Disentangled SVAE (FDSVAE): this is for the factorized alternative that each z_t is independent on y.

Also, to provide ablation study to our other contribution, for each of the variantional based model (namely SVAE-y, SVAE-z, DSVAE, FDSVAE), we implement another version with spatial constraints namely SVAE-y-S, SVAE-z-S, DSVAE-S, FDSVAE-S. For human check-in trajectories, we do not do experiments with spatial constraints since such constraints do not hold for our datasets. Any other potential constraints are left to future works.

Evaluation metrics:

Mean Distance Error (MDE) in Haversine Distance or Euclidean distance is the most used metric in current works (Ouyang et al. 2018; Huang et al. 2019; Smolyak et al. 2020), defined as $\frac{1}{n}\sum_i^N ||s_i - \hat{s}_i||_2$, where $||\cdot||_2$ denotes L2 norm. Since MDE only compute a reconstruction loss, we proposed to directly evaluate spatial feature distributions

[1] www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data

[2] www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/

[3] snap.stanford.edu/data/loc-Gowalla.html

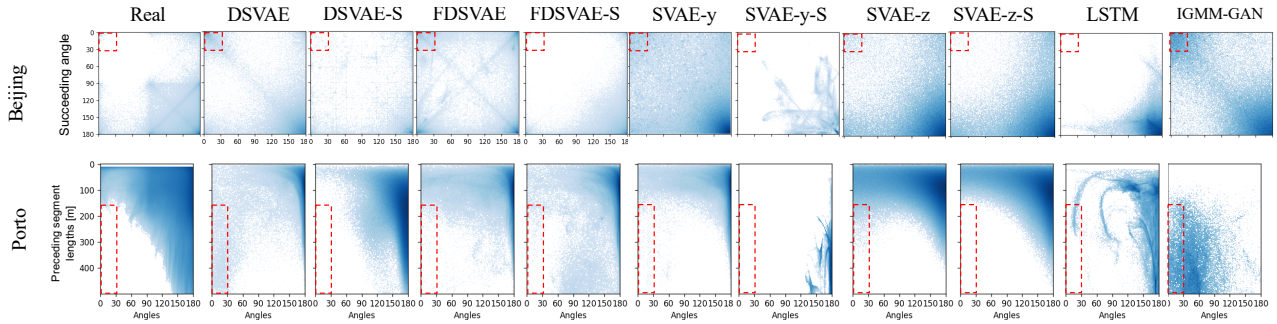


Figure 5: Spatial constraint results of Beijing and Porto datasets with red-dashed area for violation zone.

in "Maximum Mean Discrepancy (MMD)", a sophisticated distance function used in recent approaches in image (Zhang et al. 2020), text (Guo, Pasunuru, and Bansal 2020), and graph (You et al. 2018). A trajectory could have a feature vector $d_{1:T-1}$, and an original feature set is represented as $D \in \mathbf{R}^{N \times T}$, and $\tilde{D}$ for a generated feature set. Function $MMD(D, \tilde{D}) \mapsto [0, 1]$ has output value between 1 for least similarity and 0 for two exactly same distributions. The chosen spatial features include angles, segment lengths, total lengths, point counts of cells in a grid.

Violation Score (VS) is used to evaluate spatial constraint results. It's a ratio of violation case number over the total number of all cases. The lower the VS value is, the better a model outputs spatial-temporal-valid results. The formula is as follow:

$$VS = \frac{\sum_i^N \sum_{t=t_*}^T \mathbf{1}_c(t, s_{1:T}^{(i)})}{N \times (T - t_*)}, \mathbf{1}_c(\cdot) := \begin{cases} 1 & \text{if } c_*(\cdot) \wedge \dots \\ 0 & \text{else} \end{cases}$$

where t_* is the start step defined for $c_*(\cdot)$, for example, $t_* = 2$ for segment length, while $t_* = 3$ for the angle of consecutive segments.

Evaluation results

General performances: The performances of previous methods, our proposed STG methods, and ablated methods are presented in Table 1. It gives the MDE score between a real trajectory and its reconstructed trajectory and compares angle distribution, segment length distribution, total length distribution, and grid point count distribution in MMD between real trajectory sets and synthetic sets.

For the two taxi trajectory datasets, our STG outperformed other competing and ablated methods in most metrics. The margin of improvement of VAE-based models and IGMM-GAN compared to LSTM is huge. It is caused by the lack of randomness in vanilla LSTM. VAE-based models are overall preferred in MMDs. Specifically, comparing simple SVAE-y and SVAE-z to our proposed DSVAE and FDSVAE in Table 1, DSVAE is the best in MDE for both taxi dataset. The spatial constraint versions normally improve over non-constraint ones, so DSVAE-S and FDSVAE-S gained the best performances in most metrics except SVAE-z-S method achieve slightly better in angles for Porto and in total lengths and grid points in MMD for Beijing. The overall grid point distributions in Figure 6 also shows that all VAE-like models with/without constraints are generally better than LSTM and IGMM-GAN.

dataset	Metrics / Method	MDE	Angles in MMD	Segment lengths in MMD	Total lengths in MMD	Grid point counts in MMD
Porto	LSTM	13.6525	0.5243	0.4976	0.4136	0.1135
	IGMM-GAN	11.1488	0.0772	1.0	0.3779	0.0429
	SVAE-y	1.2422	0.0430	0.0034	0.0655	0.0244
	SVAE-z	1.73782	0.0081	0.0069	0.3303	0.0279
	DSVAE	0.9018	0.0649	0.0041	0.2439	0.0208
	FDSVAE	1.7415	0.0380	0.0019	0.0497	0.0294
	SVAE-y-S	1.9755	0.0795	0.1009	0.6947	0.0647
	SVAE-z-S	1.1896	**0.0054**	0.0055	0.2593	0.0106
	DSVAE-S	**0.8059**	0.0628	0.0032	0.2133	0.0208
	FDSVAE-S	1.2281	0.0273	**0.0004**	**0.0495**	**0.0105**
Beijing	LSTM	16.0594	0.4005	0.6447	0.4080	0.3717
	IGMM-GAN	0.9577	0.0556	0.7280	0.1052	0.003
	SVAE-y	0.6073	0.3844	0.3563	0.1409	0.1360
	SVAE-z	0.9849	0.0178	0.0074	0.0437	0.0005
	DSVAE	0.5916	0.1310	0.0788	0.1448	0.1040
	FDSVAE	1.1136	0.3635	0.0076	0.7308	0.1594
	SVAE-y-S	1.5179	0.4316	0.3456	0.1288	0.1207
	SVAE-z-S	0.9138	0.0200	0.0079	**0.0436**	**0.0002**
	DSVAE-S	**0.5130**	**0.0047**	0.0087	0.0852	0.0088
	FDSVAE-S	0.6118	0.0088	**0.0018**	0.0610	0.0660
POL	LSTM	34.5681	0.5921	0.9292	0.9915	0.9835
	IGMM-GAN	**0.8872**	0.1957	**0.0003**	0.0004	**0.0023**
	SVAE-y	10.6391	0.2107	0.6874	1.0067	0.4426
	SVAE-z	6.1959	0.1629	0.0044	0.0012	0.0480
	DSVAE	8.5842	0.1012	0.0030	**0.0003**	0.0463
	FDSVAE	7.2282	**0.0935**	0.0094	0.0009	0.0456
Gowalla	LSTM	629.07	0.0325	NaN	NaN	NaN
	IGMM-GAN	101.32	0.0351	0.1015	0.0431	0.0163
	SVAE-y	90.292	0.0111	0.0430	0.0017	0.0030
	SVAE-z	40.2241	**0.0067**	0.0328	0.0002	0.0026
	DSVAE	2.7645	0.0065	0.0196	0.0002	0.0036
	FDSVAE	**1.7714**	0.0102	**0.0144**	**0.0001**	**0.0025**

Table 1: Experiment results.

For two check-in datasets in Table 1, FDSVAE is preferred, except that IGMM-GAN is the lowest in MDE, segment length, and grid point of POL dataset. This might result from the relatively small sample size of POL. For the Gowalla dataset, FDSVAE won in almost all metrics, except

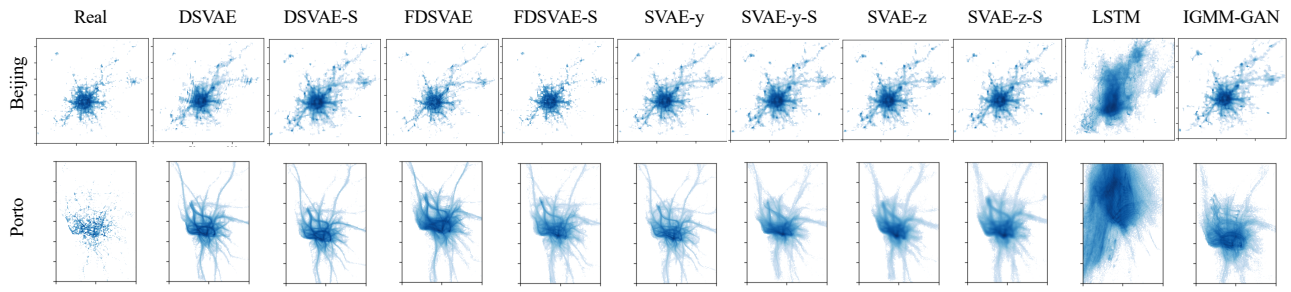


Figure 6: Point count distributions over grids for synthetic trajectories and real taxi GPS trajectories comparisons in cities.

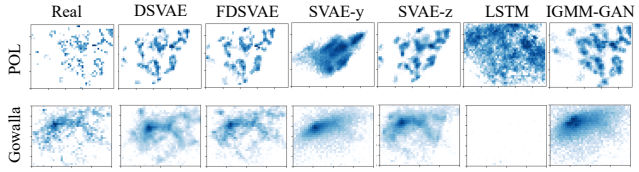


Figure 7: Point count distributions over grids for synthetic trajectories and real check-ins comparisons.

dataset	Method	VS	Method	VS
Porto	plain LSTM	0.045219	-	-
	IGMM-GAN	0.02624	-	-
	SVAE-y	0.034881	SVAE-y-S	0.004960
	SVAE-z	0.018214	SVAE-z-S	0.002749
	DSVAE	0.027971	DSVAE-S	0.003682
	FDSVAE	0.021269	FDSVAE-S	0.001180
	Raw data	0.001718	-	-
Beijing	plain LSTM	0.004753	-	-
	IGMM-GAN	0.055332	-	-
	SVAE-y	0.008197	SVAE-y-S	0.033250
	SVAE-z	0.009581	SVAE-z-S	0.006895
	DSVAE	0.010779	DSVAE-S	0.003263
	FDSVAE	0.054197	FDSVAE-S	0.007847
	Raw data	0.003395	-	-

Table 2: Violation score Experiment results.

SVAE-z won in angles. By showing grid point density in Figure 7, it is also confirmed that DSVAE, FDSVAE, and SVAE-z captured a similar pattern to real datasets.

Constraint performances: By comparing models without constraints with models with constraints in Table 2, proposed spatial regularization terms help to generate much fewer violation cases for all models since VS consistently decrease after adding constraints. In the plotted distribution of related features in Figure 5, there is much more white space (indicating zero number of samples) in red dashed regions after adding constraints.

Disentanglement analysis

In this part, we demonstrate a qualitative analysis to show that the factorization of time-variant and time-invariant factors achieved better interpretability in Figure 8. We use FDVAE-S model with Porto dataset as an example. Along x-axis, the sampled $z_{1:T}$ vectors' second dimension is replaced with values from 1 to 9. Along y-axis, we randomly

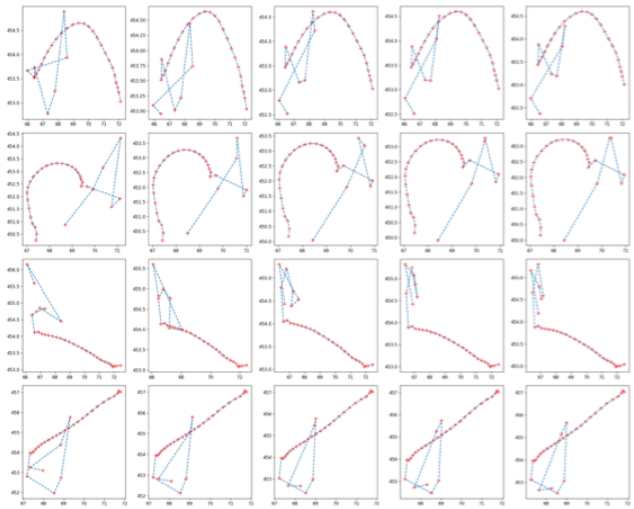


Figure 8: Disentangled factors studies. x axis is to change time-variant z_t factors, y axis is to change time-invariant y factor. This experiment used Porto dataset with FDVAE-S model.

sampled 9 different y vectors from distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$. We can see that y controls the overall trend of trajectories since trajectories in the same row present highly similar patterns. The z_t injected different noises to each step of trajectories, since trajectories in the same row show small variances. For the same column, it shows that z_t might control some high-dimensional geometric dynamics, though it is hard to visually conclude any specific geometric factor that z_t controls.

Conclusion

We develop a novel STG framework for deep generative models with spatiotemporal-validity constraints, which achieved better performance not only in the conventional MDE metric but also over feature distribution in MMD metrics and violation score. It shows that the effectiveness of factorizing time-variant and time-invariant factors, sequential priors over each time step, and constrained optimization. Even though taxi GPS trajectories and check-ins trajectory are selected, our STG framework can be also applied for other similar data mining tasks, and in other domains like animal migration trajectory, ant movement trajectory, and sport trajectory, which are left to the future works.

References

Bang, S.; Xie, P.; Lee, H.; Wu, W.; and Xing, E. 2019. Explaining a black-box using deep variational information bottleneck approach. *arXiv preprint arXiv:1902.06918* .

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35(8): 1798–1828.

Chen, R. T.; Li, X.; Grosse, R. B.; and Duvenaud, D. K. 2018. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2610–2620.

Choe, R.; Puig, J.; Cichella, V.; Xargay, E.; and Hovakimyan, N. 2015. Trajectory generation using spatial Pythagorean Hodograph Bézier curves. In *AIAA Guidance, Navigation, and Control Conference*, 0597.

Giannotti, F.; Mazzoni, A.; Puntoni, S.; and Renso, C. 2005. Synthetic generation of cellular network positioning data. In *Proceedings of the 13th annual ACM international workshop on Geographic information systems*, 12–20.

Graves, A.; and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* 18(5-6): 602–610.

Guo, H.; Pasunuru, R.; and Bansal, M. 2020. Multi-Source Domain Adaptation for Text Classification via DistanceNet-Bandits. In *AAAI*, 7830–7838.

Guo, X.; Zhao, L.; Qin, Z.; Wu, L.; Shehu, A.; and Ye, Y. 2020. Interpretable Deep Graph Generation with Node-Edge Co-Disentanglement. *arXiv preprint arXiv:2006.05385* .

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A. 2016. beta-vae: Learning basic visual concepts with a constrained variational framework .

Huang, D.; Song, X.; Fan, Z.; Jiang, R.; Shibasaki, R.; Zhang, Y.; Wang, H.; and Kato, Y. 2019. A Variational Autoencoder Based Generative Model of Urban Human Mobility. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 425–430. IEEE.

Jorris, T. R. 2007. Common aero vehicle autonomous reentry trajectory optimization satisfying waypoint and no-fly zone constraints. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND .

Kim, H.; and Mnih, A. 2018. Disentangling by factorising. *arXiv preprint arXiv:1802.05983* .

Kim, J.-S.; Jin, H.; Kavak, H.; Rouly, O. C.; Crooks, A.; Pfoser, D.; Wenk, C.; and Züfle, A. 2020. Location-based Social Network Data Generation Based on Patterns of Life. In *IEEE International Conference on Mobile Data Management (MDM20)(to appear). IEEE*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* .

Li, Y.; and Mandt, S. 2018. Disentangled sequential autoencoder. *arXiv preprint arXiv:1803.02991* .

Ma, T.; Chen, J.; and Xiao, C. 2018. Constrained generation of semantically valid graphs via regularizing variational autoencoders. In *Advances in Neural Information Processing Systems*, 7113–7124.

Mehdi, S. B.; Choe, R.; and Hovakimyan, N. 2017. Piecewise Bézier Curves for Avoiding Collisions During Multivehicle Coordinated Missions. *Journal of Guidance, Control, and Dynamics* 40(7): 1567–1578.

Ouyang, K.; Shokri, R.; Rosenblum, D. S.; and Yang, W. 2018. A Non-Parametric Generative Model for Human Trajectories. In *IJCAI*, 3812–3817.

Pelekis, N.; Ntrigkogias, C.; Tampakis, P.; Sideridis, S.; and Theodoridis, Y. 2013. Hermoupolis: a trajectory generator for simulating generalized mobility patterns. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 659–662. Springer.

Ren, W.; and Beard, R. W. 2004. Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints. *IEEE Transactions on Control Systems Technology* 12(5): 706–716.

Smolyak, D.; Gray, K.; Badirli, S.; and Mohler, G. 2020. Coupled IGMM-GANs with Applications to Anomaly Detection in Human Mobility Data. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 6(4): 1–14.

Stephens, S.; Manyam, S. G.; Casbeer, D. W.; Cichella, V.; and Kunz, D. L. 2019. Randomized Continuous Monitoring of a Target by Agents with Turn Radius Constraints. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 588–595. IEEE.

Wang, S.; Bao, Z.; Culpepper, J. S.; and Cong, G. 2020. A Survey on Trajectory Data Management, Analytics, and Learning. *arXiv preprint arXiv:2003.11547* .

You, J.; Ying, R.; Ren, X.; Hamilton, W. L.; and Leskovec, J. 2018. Graphrnn: Generating realistic graphs with deep auto-regressive models. *arXiv preprint arXiv:1802.08773* .

Zhang, W.; Zhang, X.; Lan, L.; and Luo, Z. 2020. Maximum Mean and Covariance Discrepancy for Unsupervised Domain Adaptation. *Neural Processing Letters* 51(1): 347–366.

Zheng, Y. 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6(3): 1–41.

Neural network details

We introduce details of neural networks, especially number of neurons. For Porto and Beijing dataset, the $MLP_s(\cdot)$ is multiply layers with $[48, 16]$. For all $BiLSTM_*$ and RNN_* modules, the dimensions of the hidden states (not shown in our paper) are set to 512. For the $BiLSTM_f$ of static pattern, the dimension of recurrent input o^t is 16. The MLP_{μ_f} and MLP_{σ_f} with inputs of 512×2 dimension, and with output dimension of 256. For the $BiLSTM_z$ of dynamic patterns, the dimension of input $\tilde{a}^t$ is 16. The second RNN_z module takes forward and backward hidden states as an input, whose dimension is 512×2. The hidden state

of RNN is with 512 dimension. $MLP_{\mu_{z_t}}$ and $MLP_{\sigma_{z_t}}$ is set to have one layer of 64 neurons. The priors Θ includes μ_t and σ_t, whose decoding modules $BiLSTM_\mu$ and $BiLSTM_\sigma$ have the same design of $BiLSTM_z$. The difference is that its input is a $\mathbf{0}$ vector with dimension of 16. The MLP_μ and MLP_σ have the same number of 64 neurons. The $f||z_t$ input's dimension is $256+64$ for the decoder module $BiLSTM_s$. The MLP_s has one internal layer of 128 neurons and a last layer of two neurons for two coordinate values in s_t.

There are a few differences for POL and Gowalla data. The $MLP_s(\cdot)$ is multiply layers with $[48, 32]$. For all $BiLSTM_*$ and RNN_* modules, the dimensions of the hidden states (not shown in our paper) are set to 512. For the $BiLSTM_f$ of static pattern, the dimension of recurrent input o^t is 32. The MLP_{μ_f} and MLP_{σ_f} with inputs of 512×2 dimension, and with output dimension of 256. For the $BiLSTM_z$ of dynamic patterns, the dimension of input $\tilde{a}^t$ is 32. The second RNN_z module takes forward and backward hidden states as an input, whose dimension is 512×2. The hidden state of RNN is with 512 dimension. $MLP_{\mu_{z_t}}$ and $MLP_{\sigma_{z_t}}$ is set to have one layer of 32 neurons. The priors Θ includes μ_t and σ_t, whose decoding modules $BiLSTM_\mu$ and $BiLSTM_\sigma$ have the same design of $BiLSTM_z$. The difference is that its input is a $\mathbf{0}$ vector with dimension of 32. The MLP_μ and MLP_σ have the same number of 32 neurons. The $f||z_t$ input's dimension is $256 + 32$ for the decoder module $BiLSTM_s$. The MLP_s has two internal layer of $[64, 32]$ neurons and a last layer of two neurons for two coordinate values in s_t.

Other parameter tuning

Except changing different neural networks architectures, there are several hyper-parameters to be tuned, including: 1) β parameter for β-VAE to enhance disentangling. We test values in $[1, 10, 100]$. 100 is chose for the model in our paper; 2) γ parameter for regularization of constraints. We test values in $[1, 10, 100]$. 1 is chose for the presented model in the paper; 3) other conventional parameters. Learning rate is set to be 0.0002 for Porto, 0.0002 for Beijing, 0.0002 for POL, and 0.002 for Gowalla. The training epoch are all set to be 100. The batch size is set to 128 for all datasets and models.

Additional experiment results

In this part, we did extensive case studies for different datasets so as to illustrate the effectiveness of our factorization approaches. Each row is generated with a fixed f, and each column is generated with a fixed $z_{1:T}$ sequence. We can see that for both taxi trajectories and check-in trajectories f controls a static pattern (similar patterns in each row), while z_t control the variances for each trajectory in such a row.

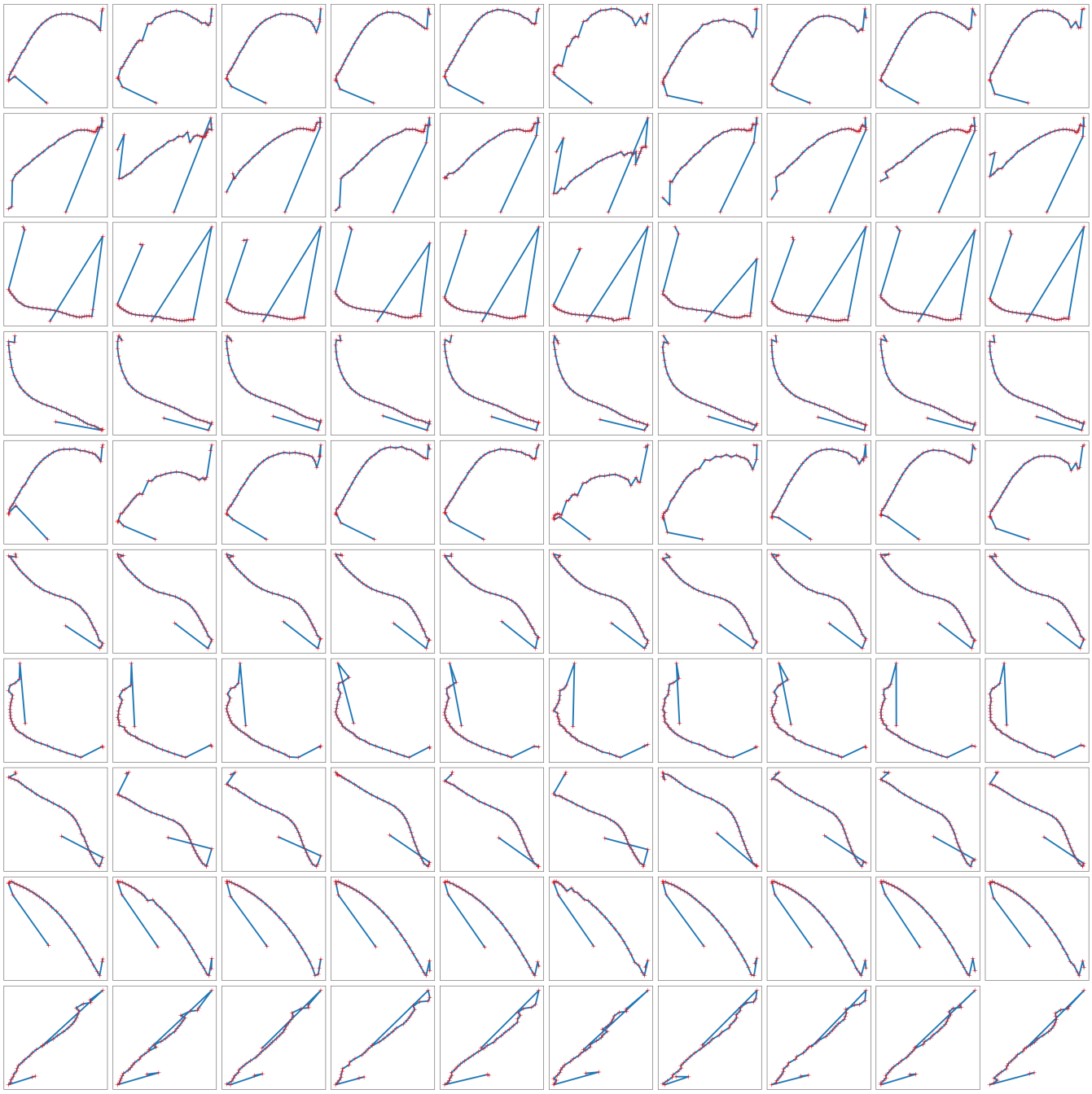


Figure 9: Case studies of f and z_t over Porto dataset

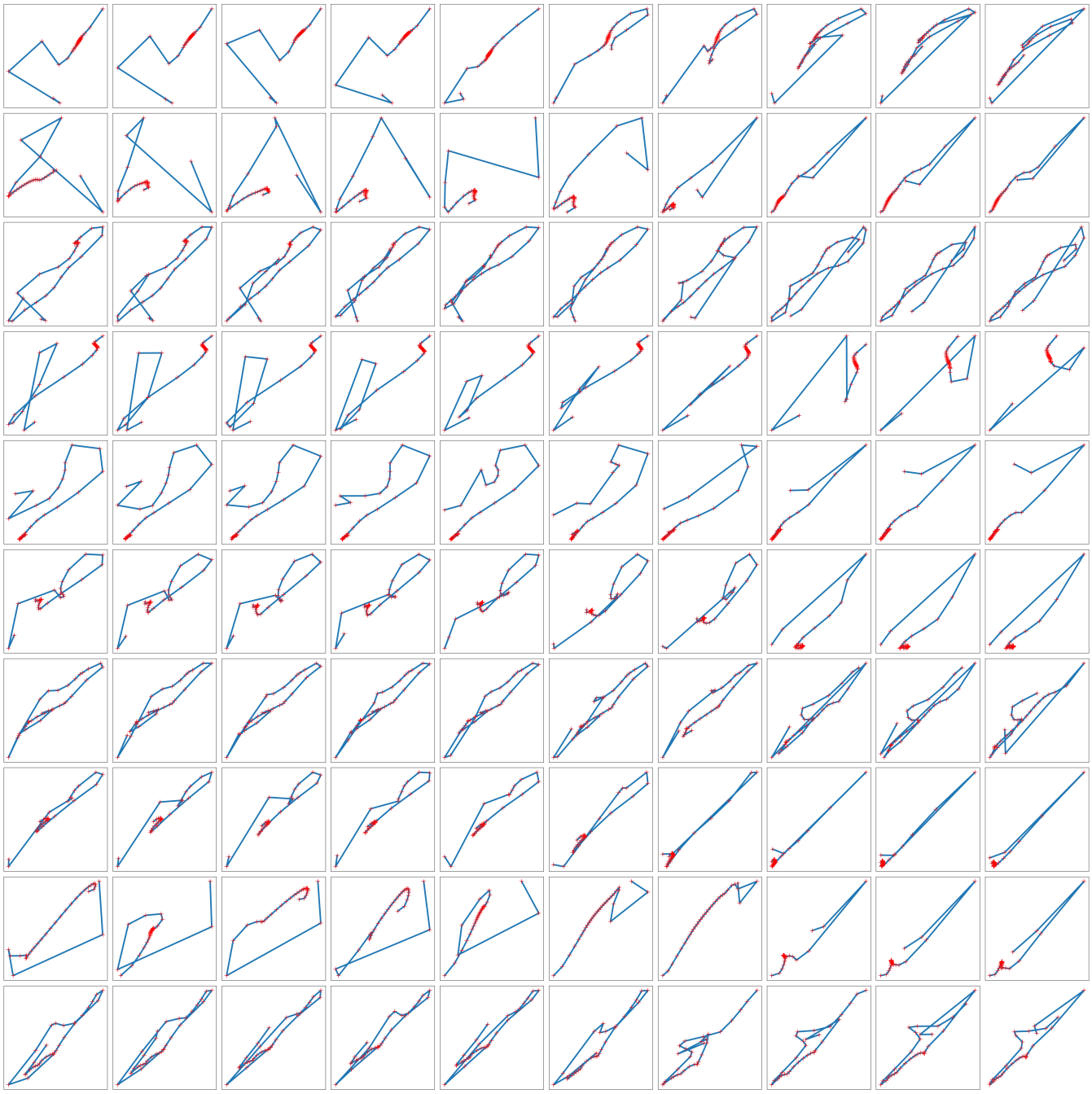


Figure 10: Case studies of f and z_t over Beijing dataset

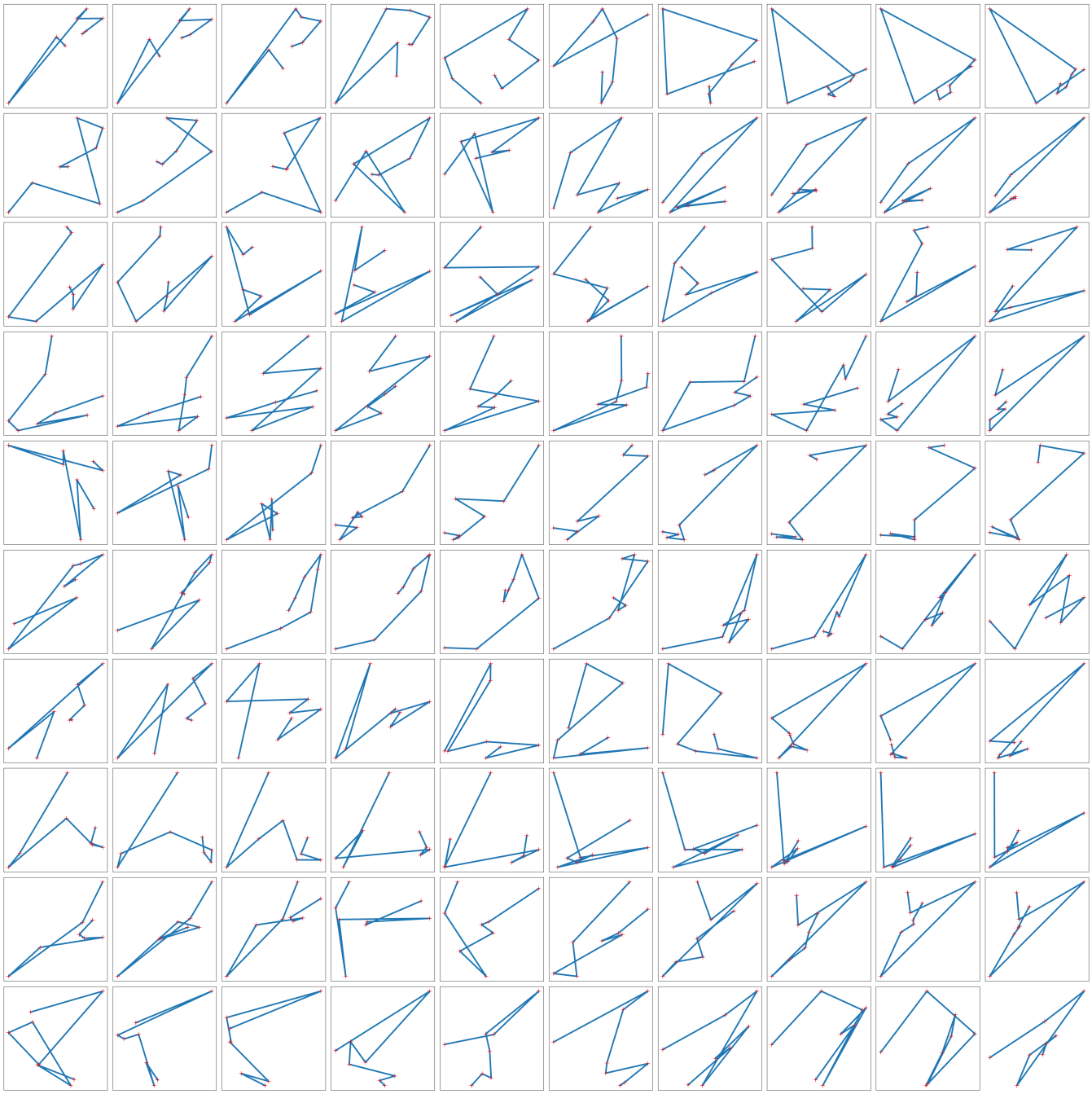


Figure 11: Case studies of f and z_t over POL dataset

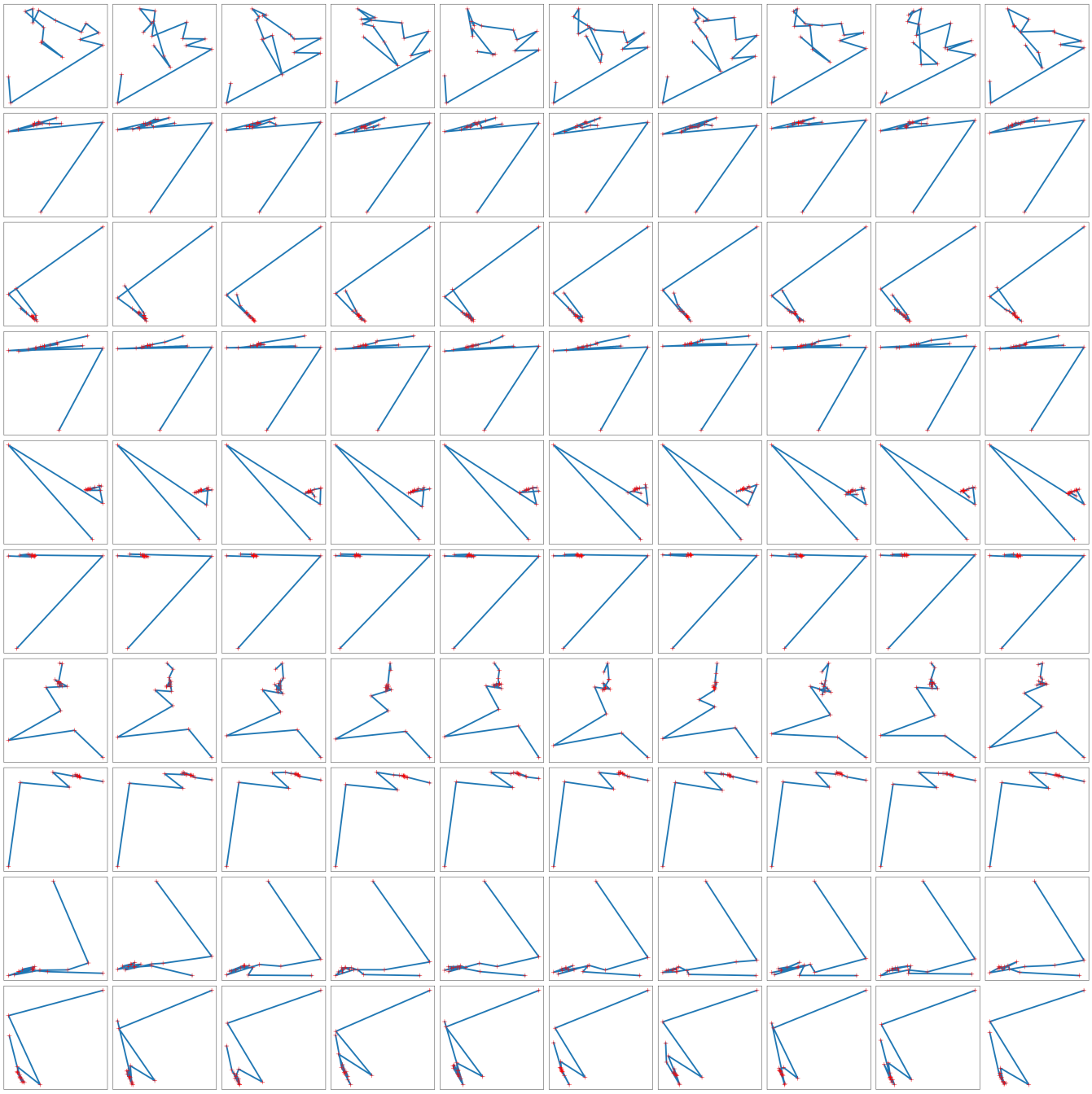


Figure 12: Case studies of f and z_t over Gowalla dataset