DEEP TEMPORAL DEAGGREGATION: LARGE-SCALE SPATIO-TEMPORAL GENERATIVE MODELS

Anonymous authors

005 006

007

008 009 010

011

013

014

015

016

017

018

019

021

022

025

026 027 028

029 030 Paper under double-blind review

Abstract

Access to spatio-temporal trajectory data is essential for improving infrastructure, preventing the spread of disease and for building autonomous vehicles. However, it remains underutilized due to limited availability, as it cannot be shared publicly due privacy concerns or other sensitive attributes. Generative time-series models have shown promise in generating non-sensitive data, but show poor performance for large-scale and complex environments. In this paper we propose a spatio-temporal generative model for trajectories, TDDPM, which outperforms and scales substantially better than state-of-the-art. The focus is primarily on trajectories of peoples' movement in cities. We propose a conditional distribution approach which unlock out-of-distribution generalization, such as to city-areas not trained on, from a spatial aggregate prior. We also show that data can be generated in a privacy-preserving manner using k-anonymity. Further, we propose a new comprehensive benchmark across several standard datasets, and evaluation measures, considering key distribution properties.

1 INTRODUCTION

Time-series data of human mobility enables applications such as pandemic forecasting and 031 management (Ilin et al., 2021), smart city development (Wang et al., 2022), urban governance 032 (Xiong et al., 2024), human rights violation detection (Tai et al., 2022) and monitoring of 033 global migration induced by war and climate change (Niva et al., 2023; Alessandrini et al., 034 2020). Two major challenges stand in the way for using time-series data to these ends. The first is a shortage of publicly available data (Ansari et al., 2024). Data can only be collected and shared in limited capacity due to privacy concerns, business concerns and 037 national security, creating a silo effect. Secondly, predictions about unobserved parts in space, 038 about the future or even possible futures - given that different actions are taken - is often a 039 necessary complement to the readily collectable data. One such case is generating high-fidelity realistic spatio-temporal trajectory data, such as individual pedestrians navigating a city or 040 a building. Open problems within the road traffic domain (Lana et al., 2018), and using 041 human mobility data at large, is (1) high quality large-scale trajectories, (2) adaptation 042 to sudden environmental changes and (3) facilitation of what-if analysis for environmental 043 changes. 044

045 A solution to both challenges is to use time-series generative models to accurately capture the data distribution. These generative models can then be adapted to generate samples 046 that are private (Yoon et al., 2019b; Wang et al., 2023), resulting in synthetic non-private 047 datasets that can be made publicly available. Further adaptation allows for tasks such as 048 imputation or forecasting (Alcaraz & Strodthoff, 2023). However, current approaches have 049 crucial limitations limiting their real-world applicability: they can only generating very 050 short sequence length and they struggle to model complex distributions with sparse support, 051 including even the smaller of spatio-temporal datasets used in this work. 052

- **053** In this work, we introduce a new method for generating high-fidelity trajectory data for complex large-scale environments, Temporal Denoising Diffusion Probabilistic Model (TDDPM),
 - 1

072

073

077

079

094

096

097 098

099

100 101

102

103

104



Figure 1: TDDPM is trained on real 2D trajectories (left) to generate synthetic trajectories (right), conditioned on how likely it should be for the population of synthetic trajectories 074 to occupy space on the 2D plane. The latter is represented as a discrete distribution over 075 occupancy frequency, i.e., the marginal distribution over the trajectory probability distribu-076 tion if integrating out time. This yields both high-fidelity in-distribution generalization (top) and out-of-distribution generalization (bottom), the latter when conditioning on a marginal distribution not part of the training data (dashed rectangle). 078

capable of out-of-distribution generalization via deaggregation from spatial statistics to 081 temporal samples, i.e. trajectories. More specifically, we first adapt the denoising diffusion 082 model to generate time-series data. Secondly, we ask: (1) Can a deep generative model 083 (in our case a diffusion model) be made to stay true to a marginal probability distribution, while generating samples from the full distribution? and (2) How can out-of-distribution 085 generalization be achieved, to support synthetic data generation of new environments or 086 hypothetical scenarios? We demonstrate how this can be achieved via conditioning the 087 model on local information. More specifically, on a spatial aggregate prior which is a non-880 temporal marginal distribution over a set of trajectories, making deaggregation to individual trajectories possible. Figure 1 illustrates tasks for and capabilities of TDDPM in a setting of 089 mobility trajectory data in a big city. 090

- 091 Our main contributions are: 092
 - We propose TDDPM, a trajectory generation method which can generate high-quality synthetic trajectory data while accurately capturing intricate spatial and temporal distributions, scaling effectively across datasets of varying size and complexity. TDDPM implements a novel deaggregation technique, leveraging spatial aggregate priors, which also enables generation of synthetic data for new, unseen environments.
 - We harmonize different notions of synthetic data quality and design a comprehensive benchmark to thoroughly evaluate performance on the unconditional synthetic trajectory data task, for large-scale and complex environments.
 - We demonstrate that TDDPM achieve state-of-the-art on this task, with unprecedented levels of quality both on sample level, and distribution-wise. We also demonstrate that the novel deaggregation architecture of TDDPM achieve out-ofdistribution generalization performance similar to in-distribution generalization.
- 105 • We show that TDDPM's novel design makes it straight-forward to apply TDDPM to tasks such as synthetic data generation for new, unseen environments, and 107 hypothetical scenarios, as well as to privacy-preserving generation using k-anonymity.

¹⁰⁸ 2 Problem Definition

109 110 111

112

113

114

115

116

The task of learning an unconditional generative model is defined as learning a mapping f from samples drawn from a known distribution $\mathcal{D}_{\text{known}}$, e.g. standard normal distribution, to samples from an unknown target distribution $\mathcal{D}_{\text{unknown}}$. The mapping is learned without direct access to the unknown distribution, and is instead limited to a set of samples $X_{\text{train}} = \{x_1, \ldots, x_N\}$, where $x_i \sim \mathcal{D}_{\text{unknown}}$. Once the mapping has been learned, synthetic data can be generating by first sampling the known distribution, then passing the individual samples through the mapping function $X_{\text{synthetic}} = \{f(y_i)\}_{i=0}^M$, where $y_i \sim \mathcal{D}_{\text{known}}$. The goal of this mapping is for the synthetic samples to be *similar* to samples from the known distribution.

117 118 119

3 Related Work

120 121

122 Previous work on unconditional generation of time-series data has focused on variations of 123 the generative adversarial networks (GAN) architecture (Esteban et al., 2017; Yoon et al., 2019a; Jeon et al., 2022) and, more recently diffusion models (Zhu et al., 2023; Yuan & Qiao, 124 2024). See Appendix A.1 for a short description of the GAN-based architectures. There has 125 also been an interest in using time-series generation for imputation and forecasting (Alcaraz 126 & Strodthoff, 2023; Tashiro et al., 2021; Shen & Kwok, 2023; Dai et al., 2023; Feng et al., 127 2024). Transformer-based time-series foundation models has been proposed as a general 128 purpose forecasting tool (Ansari et al., 2024), but has not been evaluated on the unconditional 129 generation task. 130

DiffTraj (Zhu et al., 2023) adapts the denoising diffusion architecture (Ho et al., 2020) to generate GPS-trajectories. The U-Net architecture used in (Ho et al., 2020) for denoising, is also adapted to work with trajectories. As part of the generation process, the model is conditioned on individual trajectory-specific statistics such as velocity, distance, and departure time.

Diffusion-TS (Yuan & Qiao, 2024) also adapts the denoising diffusion architecture (Ho et al., 2020) to generate time-series data by implementing the denoising step with a multilayer neural network each consisting of a transformer block, a fully connected neural network as well as time-series specific layers with the aim of improved performance and interpretability. The architecture also has support for conditional generation to enable using the models for imputation and forecasting.

142 As Diffusion-TS move away from U-Net-approach in DiffTraj to a partial Transformer architecture, we extend this further to a full Transformer-based architecture. By doing so, 143 we can simultaneously avoid the additional time-series specific induction bias in Diffusion-TS 144 while gaining increased performance and capabilities for synthetic data generation. While 145 both DiffTraj and Diffusion-TS have support for using conditional information in their models, 146 our approach differs both in terms of flexibility and in the purpose. The conditional modeling 147 in Diffusion-TS is not for increased sample quality, but instead for enabling additional 148 capabilities for additional tasks, namely forecasting and imputation, which is different than 149 synthetic data generation, and cannot be directly adapted for use of local information. 150

In DiffTraj, conditional information is used to improve sample quality. Our approach also 151 use conditional information to improve performance, the difference is the type of information 152 used for the conditioning. We use statistics related to the local environment, which we refer 153 to as *local information*, which in general cannot be directly tied to any individual sample. In 154 DiffTraj the statistics are directly related to individual samples, which in practice create a 155 one-to-one mapping between sample and conditional. The use of local information allows 156 our method to generalize to entirely new environments and scenarios, while conditioning 157 on sample-level statistics only allows for generating samples within the same spatial region 158 trained on. Conditioning on information that is sample-specific also increase the risk of 159 memorization, as the model might simply learn to store and retrieve individual training data points (trajectories) when conditioning on the individual statistic of any single training 160 data point. In our work we use distinctly different conditional information for training and 161 sampling for evaluation, making our method less susceptible to memorization issues.



Figure 2: Comparision of the heatmap of the training data and the resulting synthetic data for the models compared in this paper. The training data is taken from Geolife and covers 40 and 161 km² respectively. TDDPM, our proposed method, scales successfully from the smaller region to the larger and without re-training. Meanwhile, the other models have to be re-trained for each dataset and are mostly unable to capture the training data faithfully.

180 181

189

190

191

192 193 194

195 196

175

176

177

178

179

182

183 Outside the field of unconditional time-series generation, the transformer architec-184 ture (Vaswani et al., 2017) demonstrated that attention as viable and flexible alternative 185 to recurrence in neural networks and the architecture itself became a common architecture. The vision transformer (ViT) (Dosovitskiy et al., 2020) is based on the transformer encoder 187 and splits images into several equally sized patches which are projected before given as input tokens to the encoder. 188

In practice, the non-temporal marginal distributions are realized as heatmaps. They are processed similar to how ViT processes images (Dosovitskiy et al., 2020), by splitting the heatmaps into 8x8 equally sized patches and linearly projecting the patches into tokens.

4 TEMPORAL DENOISING DIFFUSION PROBABILISTIC MODEL (TDDPM)

197 We propose the decomposition of unconditional generation into two separate steps: (1) 198 calculating an information rich, interpretable and concise aggregation of the available data 199 and (2) use a conditional generative model to de-aggregate into synthetic samples. More 200 precisely, instead of learning to generate samples from p(x) directly, we learn to generate 201 samples from the joint distribution p(x, l), where l is aggregates of local information. This is 202 achieved via the chain rule: p(x,l) = p(x|l) p(l), where p(x|l) is a conditional deep generative 203 model and p(l) can be modeled using a explicit and interpretable distribution. This allows for more accurate and representative modelling on challenging distributions, as well as enabling 204 out-of-distribution generalization when l include sufficient information. 205

206 Sampling p(x) is done by first sampling p(l) and then using these samples to sample 207 the conditional distribution p(x|l). In practice, we learn a conditional mapping function $g: \mathcal{D}_{\text{known}} \times \mathcal{L} \to \mathcal{D}_{\text{unknown}}$, which maps samples from a known distribution D_{known} and local information $l \in \mathcal{L}$ to synthetic samples, which should be *similar* to samples from the unknown distribution $\mathcal{D}_{\text{unknown}}$. For a collection of local information $L = \{l_i\}_{i=0}^M$, a 208 209 210 synthetic dataset is generated by $X_{\text{synthetic}} = \{g(y_i, l_i)\}_{i=0}^M$, where $y_i \sim \mathcal{D}_{\text{known}}$. For certain problems, if the choice of l is sufficiently informative, we can learn to generate data for a new 211 212 213 distribution $p^*(x)$ without having to retrain p(x|l). This is achieved by estimating a new distribution $p^*(x,l)$, s.t. $p^*(x,l) \approx p(x|l) p^*(l)$. This is not only a computationally efficient 214 way to estimate $p^*(x)$, but can also be useful in cases where we only have access to l, e.g. to 215 model scenarios or account for data distribution shifts.

4.1 OCCUPANCY FREQUENCY MARGINAL DISTRIBUTION AS LOCAL INFORMATION

218 We define local information ℓ as follows. Let \mathcal{G} be a grid where each cell has width and height 219 Δ . Each cell in \mathcal{G} is indexable by i, j for i = 0, ..., N and j = 0, ..., J such that $\mathcal{G}[i, j]$ is 220 the cell on the *i*th row and *j*th column on the grid. The grid \mathcal{G} has a position, rotation and 221 area in the world, such that grid cell $\mathcal{G}[i, j]$ has position $p_{i,j}$, rotation $\theta_{i,j}$ and area Δ^2 m².

Given sequences $x[n] \in \mathbb{X}$, n = 1, ..., N. We want to calculate a relative occupancy on the grid, defined as the ratio of $x[n] \in \mathbb{X}$ that is inside each grid cell as opposed to outside of the grid cell. The probability for an observation x to be present inside $\mathcal{G}[i, j]$ is therefore

$$\ell(\mathbf{p}, \theta, \Delta) \stackrel{\text{\tiny def}}{=} p(x \in \mathcal{G}[i, j]) \propto \sum_{x[n] \in \mathbb{X}} \sum_{n} \mathbb{1}\left(x[n] \in \mathcal{G}[i, j]\right),\tag{1}$$

where $\mathbb{1}(\cdot)$ is the indicator function which is 1 if the condition is true and 0 otherwise. This constitutes a discrete probability distribution on the occupancy frequency over the grid cells, as a marginal distribution over time, n, and an aggregation over space, \mathcal{G} with resolution Δ .

4.2 HIERARCHICAL OCCUPANCY FREQUENCY MIXTURE

To increase fidelity at large scale and to model macro-scale proportionality in spatial occupancy frequency, we learn a mixture model over the cells of rigid grid over the training data support. Let \mathcal{H} be a grid where each grid-cell H[a, b] consists of an individual sub-grid $\mathcal{G}_{a,b}$, each of which has $\theta = \mathbf{0}$ and 64×64 grid cells of size Δ .

$$p(x \in \mathcal{H}[a, b]) \propto \sum_{x[n] \in \mathbb{X}} \sum_{n} \mathbb{1}\left(x[n] \in \mathcal{H}[a, b]\right),$$
(2)

The mixture model is defined as

$$p(x) = \sum_{a} \sum_{b} p(x \in \mathcal{H}[a, b]) \ell(\mathbf{p}_{a, b}, \mathbf{0}, \Delta),$$
(3)

where $p(x \in \mathcal{H}[a, b])$ is the mixture weight for component (a, b) and $\ell(\mathbf{p}_{a,b}, \mathbf{0}, \Delta)$ is the occupancy frequency distribution for region (a, b).

4.3 Architecture

230

231 232

233 234

235

236

237 238

239 240 241

246

247 248

249

To approximate p(x|l) we propose an architecture based on the denoising diffusion architecture (Ho et al., 2020), using a transformer encoder (Vaswani et al., 2017) for denoising. At each denoising step, the transformer takes the entire noisy sequence, the current denoising step and the local information as input and predicts the noise added at the current step. An overview of the approach is shown in Figure 3.

We set l to be a local occupancy map of the training data trajectories for a given region r = (x, y, θ) where θ is rotation in $[0, 2\pi)$. The occupancy map contain information about relative frequency, i.e. where which areas in the regions are more frequently visited and which areas are never visited. They are calculated using all available training data in the given region and we set the dimensions to be 64×64 . The occupancy maps are tokenized similar to the vision transformer (Dosovitskiy et al., 2020), by splitting the occupancy map into 8×8 sub-maps and then linearly projecting them. See Appendix A.2 for full details.

262 4.4 TRAINING

We extract a training data set by uniformly sampling regions (including their rotation) and, for each region, saving a tuple consisting of an occupancy map of the region and a scaled sub-trajectory from the region. The sub-trajectories are transformed subtracting the regions origin point and by rotating it accordingly. If the sub-trajectory is longer than the target sequence length, a random sub-sequence is chosen. All regions have the same size, which corresponds to approximately 1% of the total area used for training. See Figure 1 for examples of occupancy maps and trajectories.



284

295 296

297

275

285 Figure 3: Overview of the architecture. In the unconditional part, each time point of the 286 noisy trajectory is converted into a separate token with positional embedding (Vaswani et al., 287 2017) used to embed its values and the time point, as well as a learned vector representing 288 its type. The denoising step token encodes the denoising step, the step is encoding using 289 positional encoding and then concatenated with a type vector. In addition to trajectories, 290 the transformer also take a marginal distribution to guide the denoising process to generate 291 samples with particular properties, improving in-distribution performance as well as enabling 292 generalization to previously unobserved areas. The marginal distribution is split into tokens, taking inspiration from ViT (Dosovitskiy et al., 2020), concatenated with a learned type vector and the corresponding position using positional embedding. 294

4.5 SAMPLING

The target region is split up into several subregions. For each subregion, an occupancy map is calculated and the number of total users in that subregion is counted. The occupancy map is calculated by dividing the region into a set of cells, counting the number of observations in each cell and lastly normalizing it. The model is then sampled once for each subregion, passing the occupancy map and drawing a number of trajectories linearly proportional to the total number of observations in the region.

When sampling the model, we follow the original sampling schema for denoising diffusion probabilistic models (Ho et al., 2020). Note the full occupancy map is available to the model throughout the sampling process to help guide it and that no noise is added to it. The occupancy map can also be manually modified to model hypothetical scenarios, such as a road being removed or even moved.

309 310

5 Evaluation

311 312

In this section, we start with a comparative study on the unconditional generation task. We compare with state-of-the-art methods from all three major paradigms: GAN, VAE and Diffusion. These include TimeGAN (Yoon et al., 2019a), TimeVAE (Desai et al., 2021), COSCI-GAN (Seyfi et al., 2022), DiffTraj (Zhu et al., 2023) and DiffusionTS (Yuan & Qiao, 2024). We compare across two large-scale GPS-based human trajectory datasets and across several evaluation measures, which together capture all wanted properties.

We then show how our proposed method can generalize to new unseen environments. First
by training on a limited part of the map and, using only local information, generating for the
remaining map. Secondly by synthetic local information, showcasing the capability of our
approach to answer what-if questions about hypothetical environments. Current approaches
in the literature are incapable of either. Lastly, we perform an ablation study on the use of
local information.

524	Evaluation Measures					
325	There are several perspectives on synthetic data quality in the literature (Alaa et al., 2022;					
326	Wu et al., 2021; Esteban et al., 2017) covering a range of partially-overlapping aspects. In					
327	this work, we propose to harmonize these and consequently span high fidelity, covering the					
328	support, moving beyond copying the training data, downstream task informativeness and					
329	distributional proportionality as captured by the following five qualities:					
330	asonsational proportionally, as exposed by the tono img nee quantion					
000	(I) <i>Fidelity</i> : (Alaa et al., 2022) The individual synthetic samples should have similar					
000	characteristics to, or be indistinguishable from, samples from the original distribution.					
332	(II) D_{i} (Also stal 2002) It should be not it for some that is data to be dream					
333 334	from any part of the unknown distribution's support.					
335 336 337	(III) <i>Proportionality:</i> (Wu et al., 2021) The probability of a sample occurring in the synthetic distribution should be proportional to the probability of a sample occurring in the unknown distribution.					
338 339	(IV) Usefulness: (Esteban et al., 2017) The synthetic data should capture aspects of the unknown distribution that is useful for downstream tasks.					
340	(V) Concretization: (Alap at al. 2022) Synthetic complex should not be copies of the					
341	(v) Generalization. (Alaa et al., 2022) Synthetic samples should not be copies of the training data.					
342						
343 344	As there is not a single measure that can encapsulate all qualities, we propose the use of the following evaluation measures to paint a fuller picture of the synthetic data quality:					
345						
346	• TSTR: Train on synthetic, test on real. We follow the evaluation done by					
347	TimeGAN (Yoon et al., 2019a) and use the synthetic data to train a GRU-based					
348	RNN on task of one step prediction using the synthetic data. The resulting model is					
349	then evaluated on the training data and the mean absolute value is reported. This evaluates the usefulness, fidelity and diversity of the synthetic data. Lower is better.					
350	• KL Divergence: Evaluates support coverage as well as proportionality					
351	KL (real synthetic) measures how well the synthetic distribution represents the real					
352	distribution with 0 being identical KL(synthetic []real) measures how well the					
353	synthetic distribution fits inside the real distribution i.e. it is sufficient that the					
354	synthetic distribution matches a single mode of the real distribution in order to get					
355	a low divergence value. In the latter case, the synthetic distribution consequently do					
356	not need to have the same support nor have any probability density outside of this					
357	mode. Symmetric KL weight these two together and Iansen-Shannon divergence is					
358	a more stable version of symmetric KL with regions without probability in either of					
250	the distributions					
303						
360	We also make use of the sample-level measures Density Error . Trip Error . Length Error					
361	and Pattern Score , introduced in Zhu et al. (2023) (see A.3.3 for a high-level description).					
362						
363	We use two publicly real-world GPS-trajectory datasets to run the evaluation: Geolife (Zheng					
364	et al., 2011) and Porto (Moreira-Matias et al., 2013). Geolife consists of data from 182 users					
365	collected over three years, resulting in 17,621 trajectories mostly centered around Beijing,					
366	China. Porto was originally released as part of the Taxi Service Trajectory challenge and					
367	contains data on 442 taxis in Porto, Portogal. The data was collected over more than a year.					
368	For both datasets, we use only time, longitude and latitude. The data is also filtered for					
369	the time to be approximately even between samples, one observation every 10 seconds for					
370	Geolite and one every 15 second for Porto.					
274						
3/1						

32/

5.1 Large-scale unconditional trajectory generation

In this experiment, we evaluate our proposed method on the unconditional generation problem for spatio-temporal data. For each dataset we use two different subsets, a smaller region and a larger region, to scale the difficulty of the problem. The smaller regions are denoted 25% and correspond to the the lower left quarter of the larger subsets. For each dataset and size, we train each model from scratch and then sample them to generate a synthetic dataset used for evaluation. The synthetic datasets are then evaluated with a range of evaluation measures and the resulting measures are presented in Table 1. The
spatial occupancy maps of the synthetic data and the training data are presented in Figure 2
(except for the second row of TDDPM). For a more detailed view of the synthetic data, see
Appendix A.3.5. DiffTraj is used in unconditional mode since we (1) disallow the use of
identical information between training and evaluation, (2) disallow individual sample-specific
conditional information and (3) the source code for conditional mode is missing.

The proposed method is less performant on TSTR on Geolife but the best performer on Porto. This is in contrast to ocular inspection (Appendix A.3.5) of the generated trajectories, where the proposed method and DiffusionTS produce synthetic data of clearly higher quality than the other approaches. On all the distributional distributional similarity measures, the proposed method is well ahead of the related work. TDDPM excels at support coverage and proportionality.

390 391

392

5.2 Generalizing to new environments and hypothetical scenarios

To evaluate the generalization capability of our proposed method, we generate heatmaps 393 outside of the area used for training the model. In regions outside the geographical area, 394 the heatmaps are generated with data previously not observed by the model. The model 395 is sampled once for each heatmap, the number of samples drawn proportional to the total 396 number of observations in the area of the heatmap. Figure 2 for TDDPM, second row, show 397 the synthetic data for out-of-distribution generalization from the bottom-left quarter of the 398 training data. To evaluate the quality of the resulting synthetic data, we run it through 399 the same evaluation suite as previously. The resulting measures are shown in Table 2 and 400 baselines are presented in Table 1. We observe that TDDPM beats state-of-the-art even 401 while generalizing from a fraction of the environment, where the previous state-the-art have access to data from the full environment. We also provide a detailed view of the samples and 402 the resulting heatmaps in Appendix A.3.5. Finally, a proof of concept of what-if-scenario 403 modeling is shown in Appendix A.3.1. 404

The results in Table 2 indicate that the out-of-distribution generalization with the local information acts as a regularizer, with better performance overall. $KL(S \parallel R)$ being higher for 100% than for 25% indicate a higher mode-fit for the 100% model while $KL(S \parallel S)$ being lower for 100% indicate a higher degree of generalization in terms of support and proportionality of the 25% model.

410 411

5.3 Private Synthetic Trajectories with K-anonymity

412 In this experiment, we apply k-anonymity (Samarati & Sweeney, 1998) to the local information 413 to generate privacy-preserving synthetic trajectory data. We use the model trained on 25%414 of the data and generate data for the full 100%. We protect the privacy of the users in 415 the previously unseen environment by applying k-anonymity when extracting the local 416 information. More precisely, for each cell in the spatial occupancy map we set cells to 417 zero if we count fewer than k users in corresponding area. We run the experiment for a 418 range of different values for k, see Table 3 and for baselines see Table 1. We see that the 419 performance remains relatively consistent for increasing k-values. The spatial occupancy maps of the synthetic data are shown in Figure 4. We observe that the quality of the 420 synthetic data is maintained even with higher degrees of privacy, causing only small increases 421 the measures. The quality decrease mostly affects areas with fewer users, as we might only 422 have a few observations for a given street. However, this does not seem to affect downstream 423 performance. In summary, the proposed approach maintain high synthetic data quality even 424 under strict privacy constraints. See A.3.2 for a more detailed elaboration on the privacy 425 protection of our model.

426 427

427 5.3.1 Ablation Study

In this experiment, we analyze the effect of conditioning on local information over having access to no local information. TDDPM operate in a two-step process, where the first step is to construct local information and the second step draw trajectories (samples) by conditioning on the local information. TDDPM without local information does no internal conditioning

Table 1: Evaluation of different models' performance across several datasets and measures.

10.1						
434	Measure	Model	Geolife 25%	Geolife 100%	Porto 25%	Porto 100%
435		TimeGAN	$0.151\pm.089$	$0.107\pm.096$	$0.064\pm.081$	$0.065 \pm .049$
436		TimeVAE COSCI CAN	$0.120 \pm .070$ 0.121 ± 060	$0.089 \pm .084$	$0.063 \pm .076$ $0.072 \pm .081$	$0.066 \pm .048$ 0.076 ± .057
437	TSTR (\downarrow)	Diffusion-TS	$0.121 \pm .009$ $0.120 \pm .070$	$0.030 \pm .030$ $0.102 \pm .097$	$0.072 \pm .031$ $0.066 \pm .081$	$0.066 \pm .052$
438		DiffTraj	$0.137\pm.100$	$0.092\pm.078$	$0.071 \pm .076$	$0.118\pm.067$
439		TDDPM	$0.148 \pm .099$	$0.127 \pm .089$	$0.061 \pm .073$	$0.065\pm.049$
440		TimeGAN	3.617	2.712	2.755	2.955
441		COSCI-GAN	2.079	2.419 2.546	$2.324 \\ 2.398$	2.316
442	$\mathrm{KL}(S \parallel R) \ (\downarrow)$	Diffusion-TS	1.049	2.048	2.505	2.633
443		DiffTraj TDDPM	2.310	2.909	2.483	3.521
110		TDDF M	0.071	0.419	2.040	0.851
444		TimeGAN	3.000 1.531	2.175 1.821	2.645 2.084	2.497 1.739
445	$\mathbf{W}\mathbf{I}(\mathbf{P} \parallel \mathbf{S})(\mathbf{I})$	COSCI-GAN	1.802	1.922	2.225	1.860
446	$\operatorname{KL}(n \parallel S) (\downarrow)$	Diffusion-TS	1.155	1.542	2.111	2.076
447		Diff Iraj TDDPM	2.069 0.524	2.287 0.434	2.461 1.797	2.906 1.338
448		TimeCAN	3 300	2 443	2 700	2 726
449		TimeVAE	1.711	2.120	2.204	2.028
450	KLaum (1)	COSCI-GAN	1.940	2.234	2.311	2.163
451	sym (*)	Diffusion-TS DiffTrai	1.102 2 189	1.795	2.308 2.472	2.354
152		TDDPM	0.597	0.426	1.919	1.095
452		TimeGAN	0.472	0.378	0.411	0.399
453		TimeVAE	0.318	0.337	0.359	0.326
454	$JS(\downarrow)$	COSCI-GAN	0.353	0.370	0.375	0.345
455		DiffTraj	0.384	0.393	0.391	0.463
456		TDDPM	0.119	0.088	0.299	0.193
457		TimeGAN	0.320	0.201	0.163	0.132
458		TimeVAE	0.141	0.060	0.073	0.051
459	Density (\downarrow)	Diffusion-TS	0.058	0.139	0.061	0.056
460		DiffTraj	0.229	0.175	0.172	0.406
461		TDDPM	0.037	0.029	0.197	0.036
462		TimeGAN	0.362	0.238	0.166	0.154
462	T (1)	COSCI-GAN	$0.171 \\ 0.249$	0.081 0.166	0.117 0.193	0.080
403	Trip (\downarrow)	Diffusion-TS	0.086	0.076	0.076	0.068
464		DiffTraj TDDDM	0.246	0.185	0.169	0.363
465		TDDFM	0.070	0.042	0.203	0.047
466		TimeGAN	0.027	0.013	0.349	0.017
467	Length (1)	COSCI-GAN	0.579	0.729	0.677	0.612
468	Deligen (‡)	Diffusion-TS	0.014	0.020	0.014	0.010
469		TDDTTaj	0.231 0.075	0.301	0.459	0.437
470		TimeGAN	0.500	0.700	0.710	0.760
471		TimeVAE	0.630	0.760	0.740	0.840
/70	Pattern (↑)	COSCI-GAN	0.540	0.680	0.660	0.780
470	(17	Diffusion-TS DiffTrai	0.890	0.820	0.760	0.830
4/3		TDDPM	0.900	0.860	0.780	0.840
474						

Table 2: Generalization experiment.

Target	Trained on	TSTR	$\mathrm{KL}(S \parallel R)$	$\mathrm{KL}(R \parallel S)$	KL_{sym}	$_{\rm JS}$
Geolife	$100\% \\ 25\%$	$\begin{array}{c} 0.127 \pm 0.089 \\ \textbf{0.094} \pm \textbf{0.076} \end{array}$	0.419 0.453	0.434 0.378	0.426 0.416	0.088 0.085

(the gray part at the right of Figure 3 is ignored). The results are shown in Table 4, which demonstrate a clear improvement using the two-step process (with conditioning).

Table 3: Evaluation of privacy-preserving trajectory data generation. k-anonymity has been achieved by filtering out areas with fewer than k users by setting the local information to 0. Lower score corresponds to better performance.



Figure 4: Spatial occupancy maps of the training data, as well as synthetic data generated with increasing levels of privacy. Privacy is achieved using k-anonymity by removing local information corresponding to fewer than k users. Increasing the privacy decreases the quality around the less travelled roads, but affects more travelled roads less.

6 CONCLUSION

Time-series generative models for trajectory data is a challenging problem. By decomposing the unconditional task into aggregating local information and then deaggregate into highfidelity trajectories, we demonstrate that TDDPM scales to problem sizes far beyond current state-of-the-art, without compromising fidelity in the generated trajectories. By applying kanonymity on the local information, we also show that high-quality trajectories can be shared without compromising on privacy. Finally, high-quality out-of-distribution generalization is demonstrated at scale, including for what-if scenario modeling when a road is moved.

Table 4: Ablation study of the effect of local information.

Measure	Model	Geolife 25%	Geolife 100%	Porto 25%	Porto 100%
TSTR (\downarrow)	TDDPM w/o local info	$\begin{array}{c} 0.148 \pm 0.099 \\ \textbf{0.119} \pm .070 \end{array}$	$\begin{array}{c} {\bf 0.127 \pm .089} \\ {0.149 \pm .106} \end{array}$	$\begin{array}{c} {\bf 0.061 \pm .073} \\ {0.073 \pm .072} \end{array}$	$\begin{array}{c} 0.065 \pm .049 \\ 0.075 \pm .057 \end{array}$
$\mathrm{KL}(S \parallel R) \ (\downarrow)$	$\begin{array}{c} {\rm TDDPM} \\ {\rm w/o\ local\ info} \end{array}$	0.671 0.900	0.419 2.198	2.040 2.432	0.851 2.234
$\mathrm{KL}(R \parallel S) \ (\downarrow)$	$\begin{array}{c} {\rm TDDPM} \\ {\rm w/o\ local\ info} \end{array}$	0.524 0.965	0.434 1.573	1.797 2.147	1.338 1.746
$\mathrm{KL}_{\mathrm{sym}}(\downarrow)$	$\begin{array}{c} {\rm TDDPM} \\ {\rm w/o\ local\ info} \end{array}$	0.597 0.932	0.426 1.886	1.919 2.290	1.095 1.990
JS (\downarrow)	TDDPM w/o local info	0.119 0.179	0.088 0.312	0.299 0.351	0.193 0.320
Density (\downarrow)	TDDPM w/o local info	0.037 0.061	0.029 0.089	0.197 0.125	0.036 0.056
Trip (\downarrow)	$\begin{array}{c} {\rm TDDPM} \\ {\rm w/o\ local\ info} \end{array}$	0.070 0.085	0.042 0.109	0.203 0.119	0.047 0.070
Length (\downarrow)	$\begin{array}{c} {\rm TDDPM} \\ {\rm w/o\ local\ info} \end{array}$	0.075 0.086	0.106 0.114	0.105 0.171	0.105 0.303
Pattern (\uparrow)	TDDPM w/o local info	0.900 0.800	0.860 0.740	0.780 0.720	0.840 0.820

References 541

553

567

- Ahmed Alaa, Boris Van Breugel, Evgeny S. Saveliev, and Mihaela van der Schaar. How 542 faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative 543 models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, 544 and Sivan Sabato (eds.), Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pp. 290–306. PMLR, 546 17-23 Jul 2022. 547
- Juan Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and fore-548 casting with structured state space models. Transactions on Machine Learning Research, 549 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=hHiIbk7ApW. 550
- 551 Alfredo Alessandrini, Daniela Ghio, Silvia Migali, et al. Estimating net migration at high 552 spatial resolution. Publications Office of the European Union, 2020.
- Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin 554 Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shub-555 ham Kapoor, et al. Chronos: Learning the language of time series. arXiv preprint 556 arXiv:2403.07815, 2024.
- Yun Dai, Chao Yang, Kaixin Liu, Angpeng Liu, and Yi Liu. Timeddpm: Time series 558 augmentation strategy for industrial soft sensing. *IEEE Sensors Journal*, 2023. 559
- Abhyuday Desai, Cynthia Freeman, Ian Beaver, and Zuhui Wang. TimeVAE: A variational 561 auto-encoder for multivariate time series generation. arXiv preprint arXiv:2111.08095, 562 2021.
- 563 Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, 565 et al. An image is worth 16×16 words: Transformers for image recognition at scale. arXiv566 preprint arXiv:2010.11929, 2020.
- Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-valued (Medical) Time 568 Series Generation with Recurrent Conditional GANs. arXiv preprint arXiv:1706.02633, 2017.570
- 571 Shibo Feng, Chunyan Miao, Zhong Zhang, and Peilin Zhao. Latent diffusion transformer for 572 probabilistic time series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pp. 11979–11987, 2024. 573
- 574 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In 575 Advances in Neural Information Processing Systems, volume 33, pp. 6840–6851. Curran 576 Associates, Inc., 2020. 577
- Cornelia Ilin, Sébastien Annan-Phan, Xiao Hui Tai, Shikhar Mehra, Solomon Hsiang, and 578 Joshua E Blumenstock. Public mobility data enables covid-19 forecasting and management 579 at local and global scales. Scientific reports, 11(1):13531, 2021. 580
- Jinsung Jeon, Jeonghak Kim, Haryong Song, Seunghyeon Cho, and Noseong Park. Gt-gan: General purpose time series synthesis with generative adversarial networks. Advances in 582 Neural Information Processing Systems, 35:36999–37010, 2022. 583
- 584 Ibai Lana, Javier Del Ser, Manuel Velez, and Eleni I Vlahogianni. Road traffic forecasting: 585 Recent advances and new challenges. IEEE Intelligent Transportation Systems Magazine, 586 10(2):93-109, 2018.
- Luis Moreira-Matias, Michel Ferreira, Joao Mendes-Moreira, L. L., and J. J. Taxi Service 588 Trajectory - Prediction Challenge, ECML PKDD 2015. UCI Machine Learning Repository, 589 2013. DOI: https://doi.org/10.24432/C55W25. 590
- Venla Niva, Alexander Horton, Vili Virkki, Matias Heino, Maria Kosonen, Marko Kallio, Pekka Kinnunen, Guy J Abel, Raya Muttarak, Maija Taka, et al. World's human migration 592 patterns in 2000–2019 unveiled by high-resolution data. Nature Human Behaviour, 7(11): 2023-2037, 2023.

598

599

631

632

633

634

638

639

640

641

- Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: 595 k-anonymity and its enforcement through generalization and suppression. 1998. 596
 - Ali Seyfi, Jean-Francois Rajotte, and Raymond Ng. Generating multivariate time series with COmmon Source CoordInated GAN (COSCI-GAN). Advances in Neural Information Processing Systems, 35:32777–32788, 2022.
- 600 Lifeng Shen and James Kwok. Non-autoregressive conditional diffusion models for time series 601 prediction. In International Conference on Machine Learning, pp. 31016–31029. PMLR, 602 2023.
- 603 Xiao Hui Tai, Shikhar Mehra, and Joshua E Blumenstock. Mobile phone data reveal the 604 effects of violence on internal displacement in afghanistan. Nature human behaviour, 6(5): 605 624-634, 2022. 606
- Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based 607 diffusion models for probabilistic time series imputation. In M. Ranzato, A. Beygelz-608 imer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), Advances in Neu-609 ral Information Processing Systems, volume 34, pp. 24804–24816. Curran Associates, 610 Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/ 611 cfe8504bda37b575c70ee1a8276f3486-Paper.pdf. 612
- 613 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. Von 614 Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), 615 Advances in Neural Information Processing Systems, volume 30, 2017. 616
- 617 Huandong Wang, Changzheng Gao, Yuchen Wu, Depeng Jin, Lina Yao, and Yong Li. Pategail: 618 a privacy-preserving mobility trajectory generator with imitation learning. In Proceedings 619 of the AAAI Conference on Artificial Intelligence, volume 37, pp. 14539–14547, 2023.
- 620 Yun Wang, Faiz Currim, and Sudha Ram. Deep learning of spatiotemporal patterns for 621 urban mobility prediction using big data. Information Systems Research, 33(2):579–598, 622 2022.623
- Qitian Wu, Rui Gao, and Hongyuan Zha. Bridging explicit and implicit deep generative 624 models via neural stein estimators. Advances in Neural Information Processing Systems, 625 34:11274-11286, 2021. 626
- 627 Gang Xiong, Zhishuai Li, Meihua Zhao, Yu Zhang, Qinghai Miao, Yisheng Lv, and Fei-Yue 628 Wang. Trajsgan: A semantic-guiding adversarial network for rban trajectory generation. 629 IEEE Transactions on Computational Social Systems, 11(2):1733–1743, 2024. doi: 10. 630 1109/TCSS.2023.3235923.
 - Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series Generative Adversarial Networks. Advances in neural information processing systems (NeurIPS), 32, 2019a.
- Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating syn-635 thetic data with differential privacy guarantees. In International Conference on Learning 636 Representations, 2019b. URL https://openreview.net/forum?id=S1zk9iRqF7. 637
 - Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable diffusion for general time series generation. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=4h1apFj099.
- ${\rm Hao} \quad {\rm Fu}, \quad {\rm Xing} \quad {\rm Xie}, \quad {\rm Wei-Ying} \quad {\rm Ma}, \quad {\rm and} \quad {\rm Quannan} \quad {\rm Li}.$ Yu Zheng, Geo-642 trajectory dataset - User Guide, geolife gps trajectories 1.1 edilife GPS tion, July 2011. URL https://www.microsoft.com/en-us/research/publication/ 644 geolife-gps-trajectory-dataset-user-guide/. 645
- Yuanshao Zhu, Yongchao Ye, Shiyao Zhang, Xiangyu Zhao, and James Yu. Difftraj: Gener-646 ating gps trajectory with diffusion probabilistic model. Advances in Neural Information 647 Processing Systems, 36:65168–65188, 2023.

648 A APPENDIX / SUPPLEMENTAL MATERIAL

650 A.1 EXTENDED RELATED WORK

TimeGAN (Yoon et al., 2019a) consists of a generative adversarial network (GAN) operating
inside the latent space of an autoencoder. To further improve performance, they add an
additional network with the task of predicting one time step ahead. The encoder, decoder,
supervisor, generator and discriminator are all implemented using autoregressive models and
in practice they use gated recurrent units (GRUs).

The TimeVAE (Desai et al., 2021) architecture is a variant of the popular variational autoencoder architecture. The autoencoder is trained with an additional loss component to have
the latent space conform to a known statistical distribution, in this instance a multivariate
normal distribution. The autoencoder is trained to both minimize the reconstruction loss,
as well as minimizing the divergence between the embedded data and the prior set for the
latent space.

663 COSCI-GAN (Seyfi et al., 2022) proposes to use a separate generative adversarial network
664 for each channel of the data. The individual GANs all share single source of noise as input to
665 the generator and, additionally they a central discriminator that is given the stacked output
666 from the all the generators as input.

A.2 Additional Architecture Details

An overview of the approach is shown in Figure 3. Positional encoding (Ho et al., 2020):

$$\operatorname{PE}_{(pos,2i)} = \sin\left(-e^{i}\frac{\log(10000)}{\frac{d}{2}-1}\right) \tag{4}$$

$$PE_{(pos,2i+1)} = \cos\left(-e^{i}\frac{\log(10000)}{\frac{d}{2}-1}\right)$$
(5)

The input to the transformer encoder is:

- L input tokens, each token corresponding to a time point in the noisy sequence. **Note:** The token size depends on the number of features of the dataset. It is a concatenation of:
 - $-x \in \mathbb{R}^{DN}$, each corresponding to a dimension observed at each time point encoded using positional encoding
 - $x\in\mathbb{R}^N,$ the time point encoded using the positional encoding introduced in (Vaswani et al., 2017)
 - $-~x\in\mathbb{R}^N,$ a learned vector encoding denoting that this is a token that corresponds to a noisy sequence
- **Conditional information:** 64 tokens, each corresponding to a patch of the heatmap and being a concatenation of:
 - $-x \in \mathbb{R}^N$, corresponding to the x position of the patch. Encoded using positional encoding (Vaswani et al., 2017).
 - $-x \in \mathbb{R}^N$, corresponding to the y position of the patch. Encoded using positional encoding (Vaswani et al., 2017).
- $-x \in \mathbb{R}^N$, corresponding to the intensity of the heatmap. Encoded using a linear projection (Dosovitskiy et al., 2020).
 - $-x \in \mathbb{R}^N$, a learned vector encoding denoting that this is a token that corresponds to the conditional information
- A token encoding the current denoising step:
 - $\ x \in \mathbb{R}^{N(D+1)},$ the denoising step encoded using positional encoding (Vaswani et al., 2017)

702	$-x \in \mathbb{R}^N$ a learned vector encoding denoting that this is a token that corresponds
703	to the denoising step
704	a and anneally and
705	We provide a list of hyperparameters in Table 5. During the pre-processing, we split
706	trajectories into sub-trajectories if any observation leaves the geographic bounds, exceeds the
707	velocity limit or if too long time has passed since the previous observation. The velocity limit
708	is introduced to eliminate large velocities caused by sudden position changes. In the raw data,
709	a trajectory can span several days with several hours between observations. Introducing a
710	time-limit allows us to break these trajectories into individual and more time-constrained
711	journeys. For the conditional dataset, we also calculate up to 10 regions for each trajectory. The period and we get it to $1/10$ of the width and height of the measurenties
712	The region size is fixed and we set it to 1/10 of the width and height of the geographic
713	bounds.

Table 5:	Experiment	hyperparameters
----------	------------	-----------------

716	Prepr	Preprocessing				
717	Max velocity	120 km/h				
718	Max Δt	10 s				
719	Sequence length	128				
720	Region size	(0.01, 0.01)				
721	Regions per trajectory	10				
722	Geograp	hic Bounds				
723						
724	Geolife 100%	(39.9, 40, 110.3, 110.4)				
725	Geoffie 25%	(39.9, 39.95, 110.3, 110.35)				
726	$\frac{1000}{25\%}$	(41.1, 41.2, -0.05, -0.55) (41.1, 41.15, 8.65, 8.6)				
727	10100 2570	(41.1, 41.13, -0.03, -0.0)				
728	Trans	Transformer				
729	Hidden dimension N	64				
730	Number of heads	4				
731	Number of layers	2				
732	Diff	fusion				
733	Data askadada	Consistent and the second seco				
734	Number of densising store	Loo				
735	Number of denoising steps	100				
736	Tra	ining				
737	Batch size	32				
738	Number of epochs	100				
739	Optimizer	Adam				
740	Learning rate	2.9×10^{-4}				
741	(β_1, β_2)	(0.9, 0.99)				
749	(1 1) 1 2)					

714 715

A.3 Additional Results

745 A.3.1 WHAT-IF ANALYSIS

747 Current approaches also struggle with environmental and context changes. The environment 748 (and how people behave within it) often undergo rapid changes. This necessitates data ageing mechanisms that allow models to adapt their prediction to changing circumstances (Lana 749 et al., 2018), such as for example road construction, traffic accidents, traffic-light malfunction 750 or other traffic flow and road topology changes. For analysis and what-if scenario modeling, it 751 is further important that the distributions of synthetic data is proportional to the probability 752 distribution of the real data: Certain locations and motion patterns are more frequently 753 occurring than others. This is important for policy making, planning and decisions to be 754 resting on correct risk assessments using for example Bayesian inference. 755

Proof-of-concept of this capability by TDDPM is shown in figure 5.

756 758 Move road 759 760 763 764 765 Condition 766 767 Sample 768 TDDPM 770 771 772 773

Figure 5: What-if-scenario where a road is removed and added.

PRIVACY PRESERVING TRAJECTORY GENERATION A.3.2

779 Here we elaborate on the privacy protection of our model and how it extends beyond the use 780 of k-privacy.

781 We have two groups of users: 1) A set of users who have given consent to having their data 782 used for training models (we chose bottom-left for ease of exposition) and 2) a set of users 783 whose privacy we want to protect (remaining 3/4 of the data, again for ease of exposition). 784 More specifically, we would like to protect against a membership inference attack where the 785 trajectory of a single user could be extracted.

786 We use data from the first group to train the model, by extracting local information and 787 training a conditional model, which maps from local information to individual trajectories. 788 The trajectories of these users are not protected, as the model might memorize their 789 trajectories. 790

For the second group we calculate local information, which is discretized spatially and 791 time is marginalized out. The result is a heatmap, where each cell contains the number of 792 observations in the corresponding geographical area. The private information left in the 793 local information is already minimal. In the worst case scenario, if a single person were to 794 travel in a less commonly traveled part of the map, a set of locations that the person visited 795 could be extracted. The exact timing and in which order the areas were visited would still 796 be unknown (protected by how the local information is calculated). For more visited areas, 797 this risk would be minimal, as it would be difficult to isolate a single person's set of visited 798 locations.

799 By applying k-privacy to the local information, we set cells with fewer than k observations 800 to 0. This protects users (or groups of users) with fewer than k observations in each cell. 801 Once applied, the risk of membership inference attacks is eliminated; an adversary cannot 802 extract a sequence of locations a user visited (now even in less frequently visited areas).

803 804 805

761 762

774

775 776 777

778

A.3.3 Additional Measures

806 Here we investigate the quality of the resulting synthetic data following the evaluation 807 methodology from DiffTraj (Zhu et al., 2023). The results are presented in Table 6. The measures are calculated for 3000 randomly drawn synthetic samples from each combination of 808 model and dataset and 3000 randomly drawn samples from the training data. The measures 809 are:

- **Density Error**: A pair of heatmaps of the training and synthetic data are calculated by dividing the city into 16×16 blocks. The number of observations in each block are counted and normalized. The Jensen-Shannon divergence of is between the training data heatmap and synthetic data heatmap.
 - **Trip Error**: Two heatmaps pairs, each 16×16 blocks, are calculated. The first pair is of the start positions for all trajectories, one heatmap for the training data and another for the synthetic data. The second pair is calculated from the last position for all trajectories. The Jensen-Shannon divergence is calculated once for each pair and then the average is reported.
 - Length Error: The distance between consecutive observations are calculated, once for the training data and once for the synthetic data. Histograms are calculated of each, with the number of bins set to 16. Finally, the Jensen-Shannon divergence is calculated from the histogram of training data to the histogram of the synthetic data.
 - **Pattern Score**: Using the heatmaps from Density Error, the top N areas (highest count) from the training and synthetic data are collected. The F-score is then calculated and reported.

830 Dataset Model Density (\downarrow) Trip (\downarrow) Length (\downarrow) Pattern (\uparrow) 831 TimeGAN 0.320 0.3620.027 0.500832 TimeVAE 0.4590.1410.1710.630833 COSCI-GAN 0.1970.2490.5790.540Geolife 25% 0.0580.0860.014Diffusion-TS 0.890835 DiffTraj 0.229 0.2460.2310.570836 TDDPM 0.037 0.070 0.0750.900 837 TimeGAN 0.201 0.238 0.013 0.700838 0.060TimeVAE 0.0810.0720.760COSCI-GAN 0.1390.1660.7290.680839 Geolife 100% Diffusion-TS 0.0610.0760.0200.820 840 DiffTraj 0.1750.1850.3010.730841 TDDPM 0.029 0.042 0.1060.860842 TimeGAN 0.1630.1660.0750.710843 TimeVAE 0.0730.1170.3490.740844 COSCI-GAN 0.1930.6600.1160.677Porto 25% 845 0.061 0.076 Diffusion-TS 0.014 0.760846 DiffTraj 0.1720.1690.4590.620TDDPM 0.1970.2030.1050.780847 848 TimeGAN 0.1320.1540.017 0.760TimeVAE 0.0510.080 0.201 0.840 COSCI-GAN 0.1020.1660.6120.780850 Porto 100% Diffusion-TS 0.0560.0680.010 0.830DiffTraj 0.4060.3630.4370.420852 TDDPM 0.036 0.047 0.840 0.105

Table 6: Evaluation measures based on the evaluation done in DiffTraj (Zhu et al., 2023).

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827 828 829

854 855

A.3.4 The effect of training data selection on generalization

856 In this experiment, we investigate the impact of the choice of geographical data on general-857 ization. We train TDDPM on two different sub-regions, the bottom left quadrant of Geolife 858 100% and the top right quadrant. We then compare to all other models, including TDDPM, 859 trained on data from all of Geolife 100%. The results are presented in Table 7. We see that 860 TDDPM trained on top-right quadrant performs best on the TSTR measure, giving our approach the best score out of all models we have compared. We also see that training on 861 the bottom-left yields an improvement over using training data from all of Geolife 100%, 862 at the cost of slightly higher KL(S||R). This suggests that data from some geographical 863 areas might contribute more towards certain scores or aspects of performance. Further, if

the intended use of the synthetic data is known, simply selecting all available training data might not give the best possible synthetic data. This opens up a promising direction for future work: For a given task, from what regions should the training data be gathered such that the resulting synthetic data is the most useful?

Table 7: Results of experiment evaluating the effect of geographical area on generalization performance. This experiment evaluates the performance of generating a synthetic dataset for Geolife 100%. There are three variants of TDDPM in this experiment, each having access to different data. The models that does not have access to all data are generalizing to the remaining geographical area. * For additional clarity, we mark the models that are trained on sub-regions and are generalizing to the entire region.

Model	Data	TSTR	$\mathrm{KL}(S \parallel R)$	$\mathrm{KL}(R \parallel S)$	$\mathrm{KL}_{\mathrm{sym}}$	$_{\rm JS}$
TimeGAI TimeVAE COSCI-G Diffusion- DiffTraj TDDPM	AN All	$\begin{array}{c} 0.107 \pm 0.096 \\ 0.089 \pm 0.084 \\ 0.090 \pm 0.090 \\ 0.102 \pm 0.097 \\ 0.092 \pm 0.078 \\ 0.127 \pm 0.089 \end{array}$	2.712 2.419 2.546 2.048 2.909 0.419	$2.175 \\ 1.821 \\ 1.922 \\ 1.542 \\ 2.287 \\ 0.434$	$\begin{array}{c} 2.443 \\ 2.120 \\ 2.234 \\ 1.795 \\ 2.598 \\ 0.426 \end{array}$	$\begin{array}{c} 0.378 \\ 0.337 \\ 0.370 \\ 0.298 \\ 0.393 \\ 0.088 \end{array}$
TDDPM'	* Bottom-left	0.094 ± 0.076	0.453	0.378	0.416	0.085
TDDPM'	* Top-right	0.086 ± 0.086	0.538	0.453	0.496	0.100

A.3.5 ADDITIONAL VISUALIZATION

Figure 6 show a detailed view of Geolife 25%, including training data, a heatmap of the training data, the synthetic data from TDDPM as well as a heatmap of the synthetic data. In Figure 7 we show TDDPM trained on Geolife 25% generalizes to Geolife (100%) without re-training. By comparing the samples, we see that the synthetic samples are similar to the original samples. The heatmaps are also similar, meaning that the model learns to preserve proportionality.

We also show random samples for random regions of the various datasets, comparing random samples from the original data to the synthetic data from the various models. This is shown in Figures 8, 9, 10 and 11. TDDPM, our proposed method, produces the synthetic samples that are most similar to the real data. The main exception is Porto 100% where it struggles to generate data that is similar to the real data. Diffusion-TS comes in as second, which can generate data for Geolife 25% but struggles to scale up to GeoLife 100% and the samples for Porto. The trajectories from the other methods show little to no resemblance to the real data.



Under review as a conference paper at ICLR 2025 $\,$

Figure 6: Interpolation experiment. The model has trained on this region is tasked to reconstruct it from the heatmaps. *Top left:* Data from Geolife used for training. *Top right:* heatmap of training data and areas used for creating query heatmaps for sampling the model, *Bottom left:* synthetic trajectories. *Bottom right:* heatmap of the synthetic data.





Figure 8: Random samples from training data and synthetic data across 11 different regions chosen at random, all from Geolife 25%



Figure 9: Random samples from training data and synthetic data across 11 different regions chosen at random, all from Geolife 100%



Figure 10: Random samples from training data and synthetic data across 11 different regions chosen at random, all from Porto 25%



Figure 11: Random samples from training data and synthetic data across 11 different regions chosen at random, all from Porto 100%

1240