

每日小结

	周一	周二	周三	周四	周五
早	小波分析代码	修改论文	小波分析代码	Graph 代码	Graph 代码
中	修改论文	小波分析代码	小波分析代码	论文修改	修改论文
晚	Discussion 补充	组会		Graph 代码	Discussion 补充

注：简单表述当前时间段工作，如看文献 1，整理数据等

科研详情

文献阅读

文献 1

题目: DiffTraj: Generating GPS Trajectory with Diffusion Probabilistic Model

作者: Yuanshao Zhu, Yongchao Ye, Shiyao Zhang, Xiangyu Zhao, James J.Q. Yu

出处: NeurIPS 2023

方法:

**DiffTraj: 利用扩散模型生成 GPS 轨迹**

DiffTraj 模型有效地将扩散模型的生成能力与源自真实轨迹的时空特征结合起来。其核心思想是通过反向轨迹去噪过程从白噪声中重建和合成地理轨迹。此外 DiffTraj 采用 Traj-UNet 深度神经网络来嵌入条件信息并在反向过程中准确估计噪声水平。

The figure consists of two parts. The left part shows a sequence of GPS trajectories represented as blue line plots on a grid. The trajectories start at  $x_0$  and end at  $x_T$ . A dashed box highlights the reverse process from  $x_t$  to  $x_{t-1}$ , with labels  $q(x_t|x_{t-1}, x_0^{co})$  and  $p_\theta(x_{t-1}^s|x_t^s, x_0^{co})$ . The right part shows a neural network architecture. It takes time  $t$  as input to a 'Sinusoidal' block, and the current position  $x_t$  and the start position  $x_0^{co}$  as inputs to a 'Traj-UNet' block. The 'Traj-UNet' block also receives input from a 'Wide & Deep' block. The output of the 'Traj-UNet' block is  $\epsilon_\theta$ , which is used to denoise the position from  $x_t$  to  $x_{t-1}$ .

Figure 1: An illustration for trajectory generation with diffusion model. (Left) Forward and the reverse process (multiple GPS trajectories presented). (Right) Coupled the neural network model structure for reverse denoising.

**Traj-UNet:** 在开发 DiffTraj 框架时，需要准确预测每个扩散时间的噪声水平。在实际生成时，模型必须捕获各种轨迹复杂的时空依赖性才能准确地预测每个扩散时间步长的噪声水平。为此，DiffTraj 构建了一个基于 UNet 架构的神经网络模型，Traj-UNet 由两部分组成：下采样和上采样，每个部分都有多个基于 CNN 的堆叠残差网络块。考虑到轨迹数据的长度各异，在将数据送入模型前，采用了线性插值（或填充）的方法将其规范化为统一的长度。

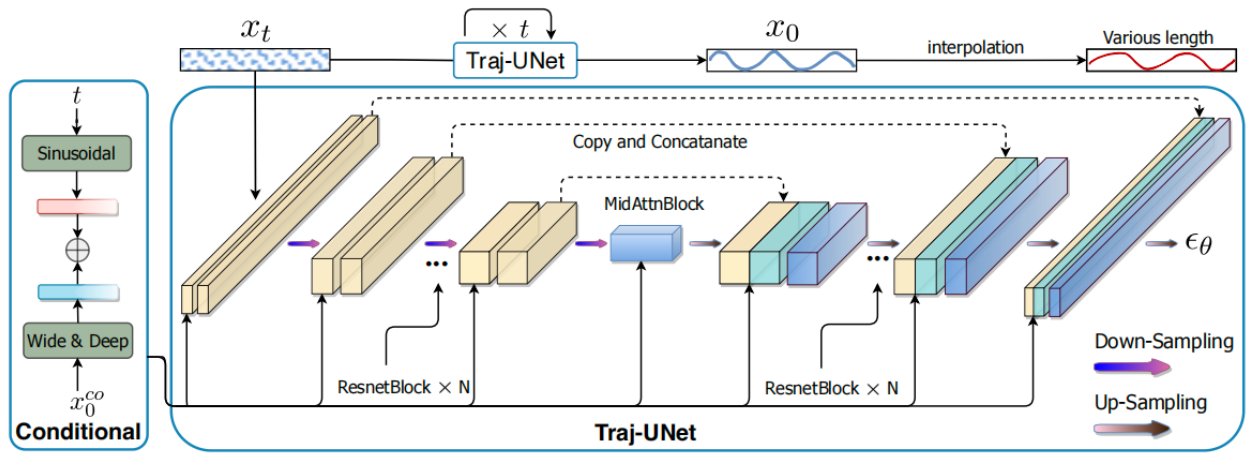


Figure 4: The network architecture used by DiffTraj in modeling  $\epsilon_\theta(x_t^s, t | x_0^{co})$  is divided into two modules, down-sampling and up-sampling, each containing multiple Resnet blocks.

**条件信息嵌入：** 此外，Traj-UNet 结合了时间步长嵌入和各种外部因素（例如旅行地区，出发时间等）。这些条件信息可以为生成过程提供有意义的指导，并确保合成轨迹表现出相似的模式和行为。因此，在提出的 DiffTraj 框架中，采用 Wide & Deep 网络结构来有效嵌入这些条件信息。其中 Wide 部分强调记忆，捕捉单个特征的重要性及其相互作用，而 Deep 部分则专注于泛化，学习高级特征组合。

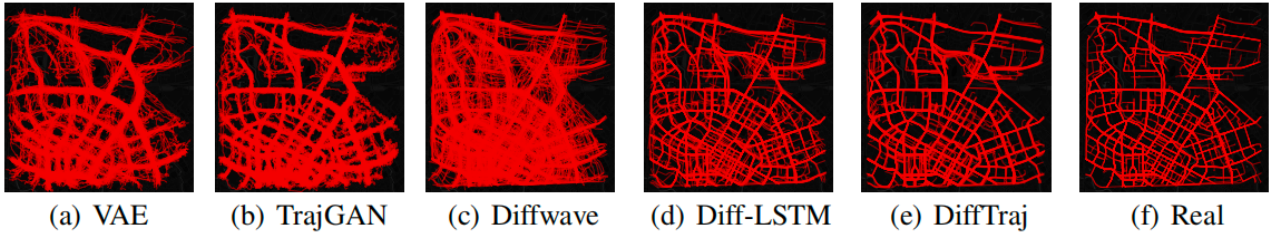
**生成多样性：** 在实际生成过程中，为了避免条件信息导致模型具有过于平滑或确定性的行为模式（这可能会破坏预期的隐私保护）。DiffTraj 采用了无分类器扩散指导方法，通过使用引入参数  $\omega$  来联合训练条件和无条件的扩散模型，并在样本质量和多样性之间进行权衡。

**加速采样：** 考虑到 DiffTraj 等扩散模型需要非常长的马尔可夫过程来生成高质量的轨迹，而事实上这是一个非常缓慢的反向扩散过程。因此，DiffTraj 采用 DDIM 进行反向扩散。通过这种方法，轨迹生成过程中的采样步骤可以从  $T$  步显著减少到  $S$  步（ $T \gg S$ ）。并且与典型的扩散模型如 DDPM 相比，该方法可以用更少的步骤生成同样高质量的样本。

Table 1: Performance comparison of different generative approaches.

Methods	Chengdu				Xi'an			
	Density ( $\downarrow$ )	Trip ( $\downarrow$ )	Length ( $\downarrow$ )	Pattern ( $\uparrow$ )	Density ( $\downarrow$ )	Trip ( $\downarrow$ )	Length ( $\downarrow$ )	Pattern ( $\uparrow$ )
RP	0.0698	0.0835	0.2337	0.493	0.0543	0.0744	0.2067	0.381
GP	0.1365	0.1590	0.1423	0.233	0.0928	0.1013	0.2164	0.233
VAE	0.0148	0.0452	0.0383	0.356	0.0237	0.0608	0.0497	0.531
TrajGAN	0.0125	0.0497	0.0388	0.502	0.0220	0.0512	0.0386	0.565
DP-TrajGAN	0.0117	0.0443	0.0221	0.706	0.0207	0.0498	0.0436	0.664
Diffwave	0.0145	0.0253	0.0315	0.741	0.0213	0.0343	0.0321	0.574
Diff-scatter	0.0209	0.0685	—	—	0.0693	0.0762	—	—
Diff-wo/UNet	0.0356	0.0868	0.0378	0.422	0.0364	0.0832	0.0396	0.367
DiffTraj-wo/Con	0.0072	0.0239	0.0376	0.643	0.0138	0.0209	0.0357	0.692
Diff-LSTM	0.0068	0.0199	0.0217	0.737	0.0142	0.0195	0.0259	0.706
DiffTraj	<b>0.0055</b>	<b>0.0154</b>	<b>0.0169</b>	<b>0.823</b>	<b>0.0126</b>	<b>0.0165</b>	<b>0.0203</b>	<b>0.764</b>

Bold indicates the statistically best performance (i.e., two-sided t-test with  $p < 0.05$ ) over the best baseline.

Figure 2: Geographic visualization of generated trajectory in Chengdu (Larger view in **Appendix D**).

启发:

1. 实验中，其他扩散模型变体 Diffwave、Diff-scatter 和 Diff-wo/UNet 由于网络结构的差异或缺乏 UNet 架构，其性能不如 DiffTraj。

文献2

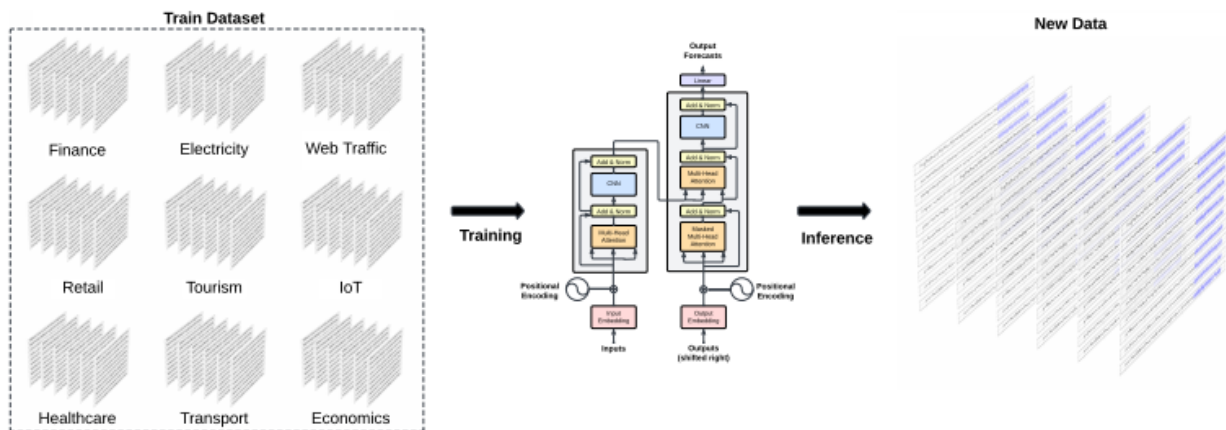
题目: TimeGPT-1

作者: Azul Garza, Max Mergenthaler-Canseco

出处: arxiv2023

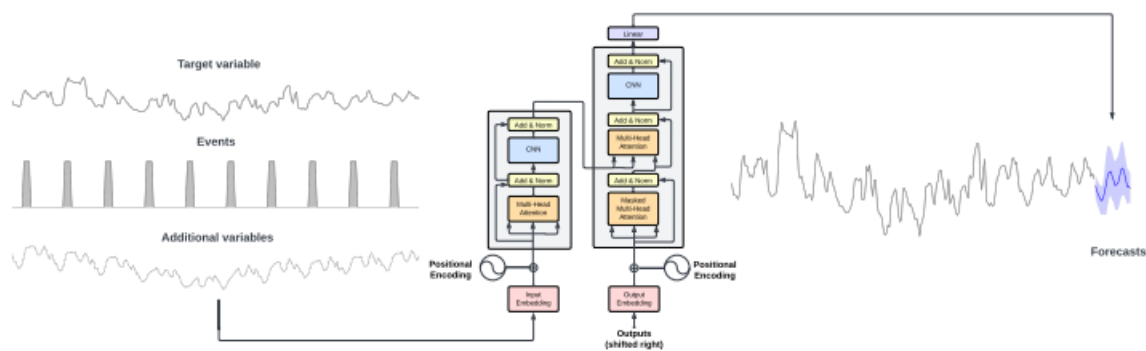
方法:

TimeGPT是一个基于Transformer的时间序列模型，采用历史值窗口来进行预测，并添加本地位置编码来丰富输入。该模型由多层编码器-解码器结构组成，每层都具有残差连接和层归一化。最后，线性层将解码器的输出映射到预测窗口维度。因此，该模型能捕捉过去事件的多样性，并正确地推断潜在的未来分布。



**Figure 2: TimeGPT was trained in the largest collection of publicly available time series, and can forecast unseen time series without re-training its parameters.**

TimeGPT 采用了公开可用的最大时间序列数据集进行训练，包含超过 1000 亿个数据点。这个训练集涵盖了来自金融、经济、人口统计、医疗保健、天气、物联网传感器数据、能源、网络流量、销售、交通和银行业等广泛领域的时间序列。由于这种多样化的领域集合，训练数据集包含具有各种特征的时间序列。在时间模式方面，训练数据集包含具有多种季节性、不同长度的周期和各种趋势类型的序列。测试集包括来自多个领域的 30 多万个时间序列，包括金融、网络流量、物联网、天气、需求和电力。



**Figure 3: Inference of new time series. TimeGPT takes the historical values of the target values and additional exogenous variables as inputs to produce the forecasts. We rely on conformal predictions based on historic errors to estimate prediction intervals.**

评估是在每个时间序列的最后一个预测窗口进行的，长度因采样频率而异。TimeGPT 使用先前的历史值作为输入，如 Figure 3 所示，而不重新训练其权重（零样本）。根据频率指定不同的预测视野，以表示常见的实际应用：月度为 12，周度为 1，日度为 7，小时度为 24。



	Monthly		Weekly		Daily		Hourly	
	rMAE	rRMSE	rMAE	rRMSE	rMAE	rRMSE	rMAE	rRMSE
ZeroModel	2.045	1.568	6.075	6.075	2.989	2.395	10.255	8.183
HistoricAverage	1.349	1.106	4.188	4.188	2.509	2.057	2.216	1.964
SeasonalNaive	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Theta	0.839	0.764	1.061	1.061	0.841	0.811	1.163	1.175
DOTheta	0.799	0.734	1.056	1.056	0.837	0.806	1.157	1.169
ETS	0.942	0.960	1.079	1.079	0.944	0.970	0.998	1.009
CES	1.024	0.946	1.002	1.002	0.919	0.899	0.878	0.896
ADIDA	0.852	0.769	1.364	1.364	0.908	0.868	2.307	2.207
IMAPA	0.852	0.769	1.364	1.364	0.908	0.868	2.307	2.207
CrostonClassic	0.989	0.857	1.805	1.805	0.995	0.933	2.157	2.043
LGBM	1.050	0.913	0.993	0.993	2.506	2.054	<b>0.733</b>	<b>0.709</b>
LSTM	0.836	0.778	1.002	1.002	0.852	0.832	0.974	0.955
DeepAR	0.988	0.878	0.987	0.987	0.853	0.826	1.028	1.028
TFT	0.752	0.700	0.954	0.954	0.817	0.791	1.120	1.112
NHITS	0.738	0.694	0.883	0.883	<b>0.788</b>	<b>0.771</b>	0.829	0.860
TimeGPT	<b>0.727</b>	<b>0.685</b>	<b>0.878</b>	<b>0.878</b>	0.804	0.780	0.852	0.878

启发:

1. 可不可以考虑用类似大模型做时序数据的预训练模型。TimeGPT作为预训练模型，它能够在不进行额外训练的情况下，在不同领域生成准确的预测，肯定也能做分类或者其他。
2. Timenet的零样本推理速度很快：平均为**0.6毫秒/序列**。与Numba编译结合的并行计算优化的统计方法平均训练和推理速度为**600毫秒/序列**，而**LGBM、LSTM和NHITS**等全局模型的平均速度为**57毫秒/序列**。由于其零样本能力，TimeGPT在总速度上优于传统的统计方法和全局模型。

#### 工作进展

- 1: 论文实验部分分析修改;
- 2: 增加了小波分析的**discussion**: 完成了代码，在选结果绘图

#### 下周计划

1. 修改完论文
2. 找长时序 graph，直接对节点做分类的相关文献
3. 数据清洗代码和建 graph 代码