Using Video Generation Models for Taxi OD Demand Matrix Prediction

Ivanov, Ivan Stanislav

July 13, 2024

Abstract

Predicting taxi demand is essential for managing urban transportation effectively. This study explores the application of next-frame prediction models—ConvLSTM and PredRNN—to forecast Origin-Destination (OD) taxi demand matrices using a concatenated dataset of NYC taxi data from early 2024. ConvLSTM achieved an RMSE of 1.27 with longer training times, while PredRNN achieved 1.59 with faster training. These models offer alternatives to traditional graph-based methods, showing strengths and trade-offs in real-world scenarios. Additionally, an open-source framework for model deployment is introduced, aiming to bridge the gap between research and practical implementation in taxi demand forecasting. Our code can be found on our Github.

Keywords: Taxi, Demand, forecasting, OD Matrix, Next-Frame Prediction Models

1 Introduction

Rapid urbanization and the influx of workers, visitors, and citizens into major cities necessitate effective transportation resource management to ensure smooth travel between different parts of the city. The taxi industry, a significant component of urban transportation systems, is projected to reach $153 billion globally by 2024 [Sta24]. Given the importance and convenience of taxis, ensuring that taxi demand in mega cities is met is a key task. Especially since there is a big gap between demanded and actually serviced taxi demand[KYZ+19]. [XSLW18] review the gap in New York City and several Chinese mega cities, and report that between 18% and 40% of requests are not answered simply due to scarce resources. Understanding and being able to predict taxi demand is crucial for service providers so that they can efficiently allocate resources and improve their service quality. One way to tackle this is through predicting Origin-Destination (OD) demand matrices, which represent the flow of taxis between different areas of a city over time.

Taxi demand prediction is inherently a spatial-temporal task. In the spatial dimension, demand can vary depending on the area of a city - residential, business, and entertainment districts. In the temporal dimension, it can fluctuate depending on the hour, or time of day (morning, lunch, afternoon, evening, night). Before the emergence of recent AI architectures, models concentrated on combining modules to capture both dimensions[WYC+21][JCZ+20][SDVB16]. Popular cases use a Graph Convolutional Network (GCN) for spatial dependencies, and Recurrent Neural Network and its variations (Gated Recurrent Unit and Long Short-Term Memory) for temporal dependencies (Table 1 provides a summary of popular OD taxi demand matrix prediction papers and their spatial and temporal modules). According to PapersWithCode (Fig. 1), the current State-of-The-Art (SoTA) for traffic forecasting (closest task to taxi demand forecasting) is based on a Graph Neural Network (GNN) architecture. Six out of the top ten best models are based on some kind of GNN, three based on the Transformer[VSP+17] architecture, and one on Other. As seen from Table 1 and Fig. 1, GNN based models present the majority, not only in traffic forecasting, but also in OD taxi demand prediction modelling. Nonetheless, systems like the public NYC taxi ride database stores information in tabular style. Transforming tabular to graph data can be very sensitive to the applied transformation methodology[LTCL24]. In addition, an OD taxi demand matrix is simpler compared to a graph because its entries directly represent taxi demand from each row to each column.

While there has been success using GNN based methods, advances in next-frame prediction[SBK+] and video generation[Ope] models pose an exciting opportunity for the OD taxi demand matrix prediction task. The reason graph-based methods became widely used is because they can better capture

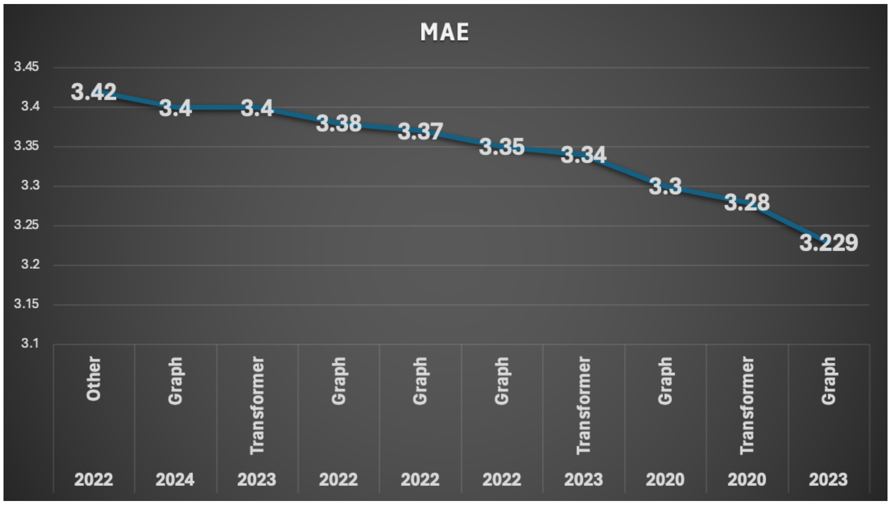

Figure 1: SoTA Traffic Forecasting models from PapersWithCode

both geographical and semantic (similar, but geographically distant regions) relationships between regions[HZWY22]. However in next-frame prediction models, the input can be of Euclidean structure. In these models, OD demand matrices at different time steps are thought of as frames, representing the evolving demand over time. Now, based on previous n frames, the model's aim is to predict the next frame(s). Majority of next-frame prediction works concentrate on human action, car camera and minor tasks like moving numbers (moving MNIST). Table 2 and 3 summarise the papers on next-frame prediction and video generation and the datasets/cases they have been applied to. Such models can capture the patterns over time and space[RVA$^+$] and provide a powerful tool for our OD taxi demand case.

In addition, we noticed there is a gap between research and application when it comes to applying any OD taxi demand model to real-life application[JL][DP23].

Therefore, We formulate the contributions of the paper as follows:

- Evaluate next-frame prediction models on the NYC taxi dataset
- Comparison between next-frame video generation models applied in the OD taxi demand matrix domain
- An open-source skeleton framework for in-practice deployment

The remainder of this paper is structured as follows: in Part 2, we provide a review on existing methods for OD matrix prediction, in Part 3 we provide Preliminaries, in Part 4 - Framework Design. Then in Part 5: Methodology, Path 6: Results, and ending with Part 7: Discussion and Part 8: Conclusion.

2 Previous work

Taxi demand has been explored from different perspectives - staring from methods like RNN and its variations in LSTM and GRU, CNN models, combination architectures with RNN and CNN modules, GNN, and recently video (next-frame prediction) models started to be applied as well. In this section, we explore the aforementioned methods and their application in OD taxi demand matrix problems.

2.1 Classical Models

The most famous and trivial methods used for taxi demand forecasting are Historical Average (HA) and Auto Regressive Integrated Moving Average (ARIMA). They are used in most of the papers from Tables 1 and 2 as baseline methods. HA calculates the average demand over a specified historical period and because of its simplistic implementation provides a good baseline forecast. However, it

Paper	Spatial module	Temporal module
CWGAN-div (2023)[WZSL23]	GAN	ResNet
STTCM (2024)[LLM+24]	Tree convolution	CNN+GLU
BGARN (2022)[STT]	GCN	RNN
DMGC-GAN (2022)[HZWY22]	GCN	GRU
MLRNN (2022)[ZZL+22]	RNN	RNN
ST-DCN (2022)[LSY+22]	Diffusion GCN	Hybrid diluted convolution+TCN
CACRNN (2020)[LWZT20]	GRU+LC	GRU+LC
DMVST-VGNN (2020)[JXS+20]	GAT	CNN+Transformer
MPGCN (2020)[SYG+20]	GCN	LSTM
MSTIF-Net (2020)[JCZ+20]	GCN	Seq2Seq
GEML (2019)[WYC+19]	Grid embedding	LSTM
H-CNN (2019)[KYZ+19]	CNN	CNN
LSTNet (2018)[LCYL17]	CNN	RNN
LSTM (2017)[XRBT17]	n/a	LSTM

Table 1: Popular graph-based OD taxi demand matrix models

cannot detect changing demand patterns, trends and seasonality. ARIMA models the next value in a time series as a linear function of its own past values and past forecast errors. ARIMA can process patterns like trend and seasonality, but fails to account for non-linear patterns.

To overcome the simplicity of the above methods, methods like Support Vector Regression (SVR) ([LZD+18]) and LSTM ([XRBT17]) were applied and are commonly used as baseline models (Table 1 & 2). SVR, though sensitive to hyperparameters, performs well as a baseline demand prediction model. LSTM has also established itself as a reliable baseline architecture for taxi demand forecasting, leveraging its ability to capture intricate temporal dependencies and nonlinear patterns in the data, making it a preferred choice in urban transportation forecasting tasks.

2.2 Spatial-Temporal non-GNN Models

Taxi demand data inherently possesses both spatial and temporal characteristics. Spatial due to different areas (i.e. residential, entertainment, business districts) having varying demand patterns, and temporal due to fluctuating demand throughout the day, week, and year, influenced by factors such as weather, events, and socioeconomic trends. Due to this complex relationship, research has focused on developing architectures that can capture both type of dependencies.

As seen from Table 1, an RNN (and its variations) is a common choice for an spatial-temporal (ST) model's temporal module. RNNs excel in modeling sequential data by maintaining and updating internal states that evolve with each input time step[DP23]. While RNNs perform well in extracting temporal features from data, CNNs have been introduced in order to capture spatial features. CNNs excel in processing spatial data such as maps or geographic information, enabling the ST model to integrate spatial contexts alongside temporal dynamics, thereby enhancing its predictive accuracy for complex urban phenomena like taxi demand forecasting[DP23]. Here methods like LSTNet[XRBT17], which uses a CNN module for spatial, and an LSTM module for temporal dependencies. LSTNet is commonly used as a baseline method.

2.2.1 Problems with CNNs for extracting spatial features

CNNs are great at capturing Euclidean-structured data. As such, research has criticised them for their inability to capture long-distance spatial dependencies in non-Euclidean structured data ([ZSZL][LYSL17][ZFWQ20]). Due to this, some papers ([ZZL+22][KW18][ZSZ+]) have developed architectures that use an RNN (or its variations - GRU/LSTM) for both spatial and temporal feature extraction. MLRNN focuses on temporal dependencies using recurrent layers with spatial embeddings and attention mechanisms, whereas HST-LSTM integrates hierarchical spatial representations and a specialized ST-LSTM to explicitly stack layers for capturing spatial dependencies at various scales. While such methods have found success in capturing ST features, to better capture spatial dependencies, Graph Neural Networks and specifically Graph Convolutional Networks began to be applied.

Model	Datasets
SimVP (2022a)[GTWL22]	Moving-MNIST TrafficBJ Human3.6 Caltech-pedestrian KTH
SimVP (2022b)[TGLL]	Moving-MNIST TaxiBJ WeatherBench Caltech-pedestrian KTH
ST Video Autoencoder (2016)[PHC16]	Moving-MNIST HMDB-51
ST-ResNet (2017)[ZZQ17]	TaxiBJ BikeNYC
STDiff (2023)[YB24]	KITTI Cityscapes KTH BAIR-Robot Stochastic-moving-MNIST
SUMformer (2023)[CLL+23]	TaxiBJ Chengdu-traffic NYC-Taxi
SwinLSTM (2023)[TLZT23]	Moving-MNIST Human3.6 KTH TaxiBJ
TAU (2023)[TGW+23]	Moving-MNIST TaxiBJ KTH Caltech-pedestrian
Learning Physical Laws (2024)[WHK+24]	2D-Bouncing 3D-Bouncing Roller-Pendulum Blocks Moon
VideoFlow (2020)[KBE+20]	Stochastic-Movement BAIR-Robot
Stochastic Frame Pred. (2024)[JKK+24]	Kinetics-400

Table 2: Models and their corresponding datasets

2.3 Graph Neural Networks

Taxi data possesses both spatial and temporal features and it can fit well within a Graph Neural Network model. In addition, graphs fit well into data structured in a non-Euclidean space ([HZWY22]). This would reflect the the real-world road network and urban layout that is exhibited by taxi data. In images and CNNs, the data is structured as a grid where each cell represents a pixel. In graphs and GNNs, data is split into nodes and edges, where nodes represent areas and edges can represent roads, and connections between areas.

2.3.1 Graph Convolutional Networks

As mentioned earlier, CNNs are inadequate in capturing non-Euclidean structure. Graph Convolutional Networks (GCNs) address this limitation. TGCN ([ZSZ+]) represents roads as nodes and by

Model	Datasets
Conv-TT-LSTM (2020)[SBK+]	KTH-action Moving-MNIST-2 Something-Something-V2
ConvLSTM (2015)[SCW+]	Moving-MNIST Radar-Echo
GDDNet (2023)[LC23]	KITTI Caltech-pedestrian UCF-101
Latent Video Transformer (2020)[RVA+] Latent Video Transformer	BAIR-Robot Kinetics-600
MIM (2018)[WZZ+]	Moving-MNIST TaxiBJ Radar-Echo Human3.6M
PredCNN (2018)[XWLW18]	TaxiBJ BikeNYC
Model	Datasets
PredRNNv2 (2022)[WWZ+]	Moving-MNIST KTH Radar-Echo Traffic4Cast BAIR-Robot
Scaling AR Video Models (2019)[WTU]	BAIR-Robot Kinetics

Table 3: Models and their corresponding datasets

stacking multiple GCNs it can better capture the topological structure of taxi road networks. [ZSZL] improves upon TGCN by introducing attention to the architecture. Such attention mechanisms are commonly applied, and they aim to recalculate the influence of different nodes and edges in the graph, which in turn helps to capture more complex and dynamic relationships. ASTGCN ([GLF+19]) splits its input traffic flow into recent hour(s), daily-periodic, and weekly-periodic, processes them through ST attention and ST convolution mechanisms and the outputs are re-weighted and fused. This method allows the model to learn dynamic location influences through the spatial attention, and correlations between time steps thanks to the temporal attention. For traffic forecasting, GMAN [ZFWQ20] uses node2vec for ST embeddings, incorporates spatial and temporal attention mechanisms, and applies a transformer attention layer within an encoder-decoder architecture to capture dynamic relationships and improve long-term traffic forecasting accuracy. MSTIF-Net ([JCZ+20]) integrates multi-graph representation, latent global situation representation, and a sequence-to-sequence learning framework (Seq2seq) with attention mechanisms for effective urban ride-hailing demand prediction. Now, in the taxi demand forecasting domain, ST-DCN ([LSY+22]) employs a two-phase graph diffusion convolutional approach enhanced with an attention mechanism. The paper introduces the usage of a multi-task learning module with a periodic-skip LSTM which works well for time series data with periodic patterns such as taxi demand. While these models work well for predicting taxi demand, they require the existence of graph-structured data, and we believe that constructing such that is unnecessary due to more modern architectures such as next-frame prediction models

2.4 Next-frame Prediction Models

Given a system's past and current state, next-frame prediction (NFP) models aim to predict the future state, typically using sequential data that resembles images. Specifically, in the context of OD taxi demand matrix, a single OD matrix is regarded as a frame and a series of such frames are input through the model, and the result is an OD matrix at a future time step. ConvLSTM ([SCW+]) is a popular baseline and referenced paper when referring to NFP models, and it integrates CNNs and LSTM to handle spatial and temporal data. However, due to its spatial module being based on CNN its quality in capturing spatial dependencies is limited and the resulting predicted future frames become blurry. To better learn long-range dependencies, ST-ResNet ([ZZQ17]) uses residual connections to effectively capture temporal closeness, period, and trend properties of crowd traffic. By employing residual convolutional units in three separate branches, ST-ResNet can model immediate past data (closeness), daily periodicity (period), and longer-term trends (trend) independently. This approach allows for a more nuanced understanding of spatio-temporal dynamics and enables the model to dynamically aggregate outputs from these branches, assigning different weights based on data characteristics.These advancements highlight a trend towards more sophisticated architectures. MIM ([WZZ+]) proposes an NFP model that specialises in learning higher-order non-stationarity from spatial-temporal data. It uses differential signals between adjacent recurrent states and stacked MIM blocks to model these non-stationarity properties. This is one of the NFP models which is tested on predicting taxi traffic flow state and outperforms the ConvLSTM architecture. PredCNN ([XWLW18]) also performs well in predicting taxi traffic flow state frames. However, here a novel CNN-based architecture is introduced that uses a cascade multiplicative unit (CMU) by stacking multiple convolutional layers are stacked in a sequence with each layer processing the output of the previous. This architecture improves training time and memory utilisation compared to RNN-based models. While previous methods employ a combination of convolutional networks with RNNs, SimVP [TGLL] uses a conventional CNN framework which significantly reduces model complexity, and thereafter training time. Occam's razor for models states that the simplest model is usually the best choice, and regardless of SimVP's simplistic architecture it managed to achieve SoTA results.

As seen above and in Tables 2 and 3, papers that develop NFP models tend to concentrate on testing their models in different domains - traffic, taxi, human action, etc. While some papers include datasets on taxi demand in their evaluation, OD taxi demand matrix is rarely the main prediction target.

2.5 Real-life Applications

OD taxi demand matrix prediction models lack real-world application ([DP23][JL]). Many paper share their code publicly on GitHub. While some papers share how to run a python script with input, instructions on what that input needs to look like, nor a clear pipeline of how to go from raw data to a prediction are provided (add Table showing which papers do and which do not). The majority either do not provide instructions on applying the data preprocessing steps to our own data, neither have clear instructions on running the full code. Such gaps can prevent practitioners from employing models in their own companies and using them for real-world application. Therefore, in our GitHub repository, we provide a full pipeline from raw data schema/format to getting prediction result in a reproducible pipeline using data and machine learning development tools like MLflow and Prefect for experiment tracking and pipeline orchestration. By providing a pipeline for practitioners to use, we hope to bridge the gap between research and practice in the OD taxi demand prediction domain.

3 Preliminaries

3.1 Time Slot

In the context of taxi demand forecasting, a time slot refers to a specific interval of time during which data is aggregated and analyzed. In this paper we divide our demand into 1 hour slots. However, time slots can vary in length, commonly ranging from minutes to hours. A time slot might be defined as every 15 minutes, half-hour, or hour. The choice of time slot duration depends on the granularity required for the prediction task and the nature of the demand patterns. Shorter time slots can capture more detailed variations in demand but may lead to increased computational complexity, while longer time slots simplify the data but may overlook short-term fluctuations.

3.2 Origin-Demand Matrix

The Origin-Demand (OD) matrix is a key concept in taxi demand forecasting, representing the demand between different regions within a city. Each element in the OD matrix corresponds to the number of taxi requests from an origin cell to a destination cell within a given time slot. Formally, if the grid divides the city into N cells, the OD matrix for a specific time slot is an N x N matrix where the entry

(i, j) indicates the demand for trips originating in cell i and ending in cell j. This matrix captures both spatial and temporal demand patterns, making it a crucial input for forecasting models that aim to predict future taxi demand based on past observations.

3.3 Request

A request refers to an individual instance of a taxi service case by a passenger. We do not have access to unanswered calls, so request refers to answered and fulfilled calls only. Each request typically includes information such as the time the request was made, the location of the passenger (origin), and the intended destination. These requests are aggregated into the OD demand matrix over the specified time slots to form the data used for forecasting.

4 Framework Design

4.1 Experiment tracking

We will run models and save their artifacts using MLflow. It provides capabilities for managing and tracking experiments in machine learning projects. It allows to log parameters, code versions, metrics, and artifacts when running machine learning experiments, making it easier to reproduce and compare results across different runs.

4.2 Infrastructure as a Service

Terraform is an Infrastructure as Code (IaC) tool that allows you to define and manage infrastructure resources in a declarative manner. It supports various cloud providers (such as AWS, Azure, Google Cloud) and can be used to provision and manage databases, storage buckets, virtual machines, and other infrastructure components needed for applications like MLflow. By using Terraform, we can automate the deployment and scaling of infrastructure resources, ensuring consistency and reliability across different environments.

4.3 Orchestration

Prefect is a workflow orchestration tool designed for automating data workflows. It provides a platform to build, schedule, and monitor workflows that involve tasks such as downloading data, preprocessing, model training, and prediction. Prefect's features include dependency management, error handling, and integration with various execution environments (local, cloud, Kubernetes). It is particularly useful for orchestrating complex data pipelines in machine learning projects, ensuring tasks are executed in the right order and managed efficiently. In this paper, Prefect is used to orchestrate pipelines for downloading NYC taxi data, preprocessing it, creating adjacency matrices, training models, and making predictions.

5 Methodology

5.1 Dataset

We use a concatenated NYC taxi dataset of Jan, Feb, Mar, and Apr 2024, made available online by the New York City Taxi and Limousine Commission. We filter the dataset only to include the Manhattan area as it is the only area with dense demand. The dataset contains various variables related to taxi trips, but for our video generation models - we are only interested in pickup time (tpep_pickup_datetime), dropoff time (tpep_dropoff_datetime) - in order to split the dataset into one hour time slots; pickup location id (PULocationID), and dropoff location id (DOLocationID) in order to populate the cells in the adjacency matrices which represent demand between two locations. The full process can be found in prefect_flows/create_adj_matrices.py in our repository. We use a 80/20 train/test split on the dataset for all models. The prediction target is demand between each region.

5.2 Model selection

We select Historical Average (HA) as our baseline model (code in prefect_flows/historical_avg_demand.py). In addition, we select ConvLSTM (code in prefect_flows/train_conv_lstm.py) and PredRNN (code in prefect_flows/train_pred_rnn.py) as video generation baseline models. ConvLSTM is based on the original paper, while PredRNN is an adjusted version from the paper's GitHub repository. Both ConvLSTM and PredRNN are trained to take the previous 10 OD matrices (10 x 1hr) and predict the next hour's OD matrix.

5.3 Evaluation

We employ Root Mean Square Error (RMSE) as our main evaluation metric to compare model performance. For example, RMSE of 1 would mean that, on average, the model's predictions deviate from the actual values by approximately 1 unit.

5.4 Hardware

Model training and evaluation was done on a Macbook Pro 14, M2 Pro chip, 10 core CPU, 16 core GPU, 16GB RAM.

6 Results

Model	**RMSE**	**Train Time**
Historical Average (HA)	21.54	42ms
ConvLSTM	1.27	3.2hr
PredRNN	1.59	2.2hr

Table 4: Comparison of RMSE and Training Time for Different Models (as recorded by MLflow)

6.1 Historical Average

HA is used as a baseline model and it manages to achieve an RMSE of 21.54.

6.2 ConvLSTM

ConvLSTM achieved the best results - RMSE of 1.27, and it took 3.2 hours to train the model. The model parameters are in Table 5. A model summary text file generated by PyTorch can be found in our repository. Since the ConvLSTM model resulted in the best RMSE, we created a pipeline for predicting unseen (not train/test) data. The used dataset is again NYC taxi from Jan 2023. The model expects input in the shape (16, 10, 1, 67, 67) (batch_size, input_length, channels, height, width), therefor the data was transformed and input. Figure 2 shows sample Predicted vs GT (ground truth) and difference (for more images, see the sample_conv_lstm_pred_img folder in our repository.

Parameter	**Value**
input_dim	1
hidden_dim	64
kernel_size	(3, 3)
num_layers	5
batch_first	True
lr	0.0001

Table 5: ConvLSMT Parameters

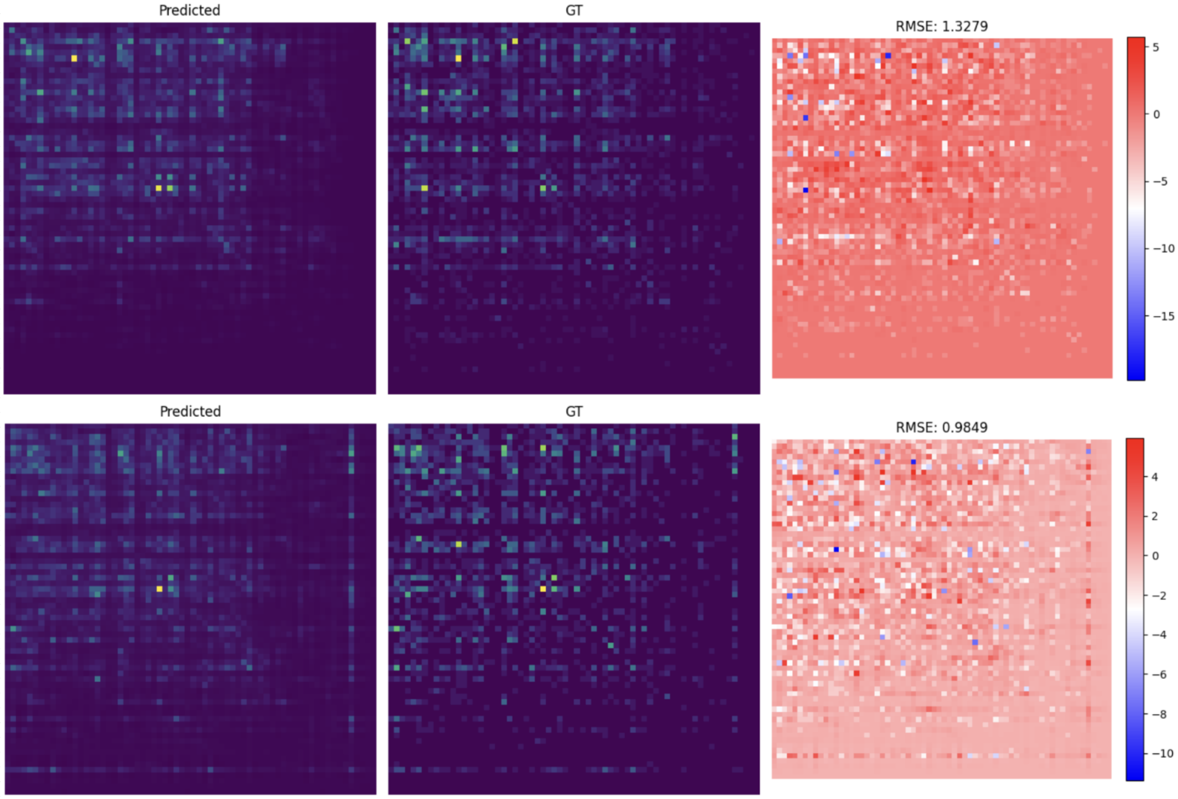

Figure 2: ConvLSTM: Prediction vs GT

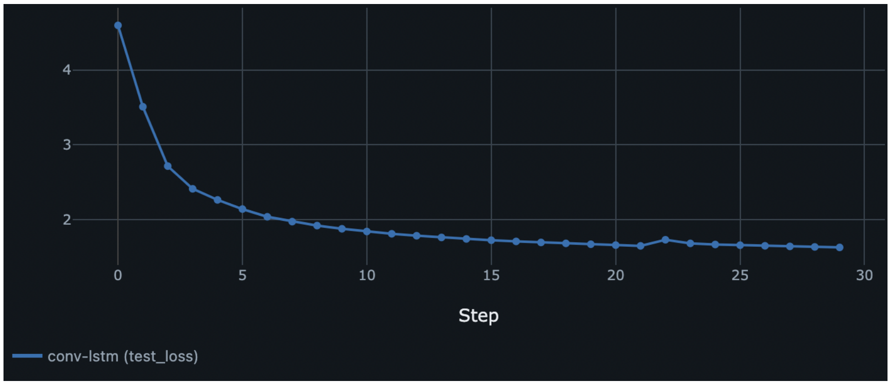

Figure 3: ConvLSTM test loss over epochs

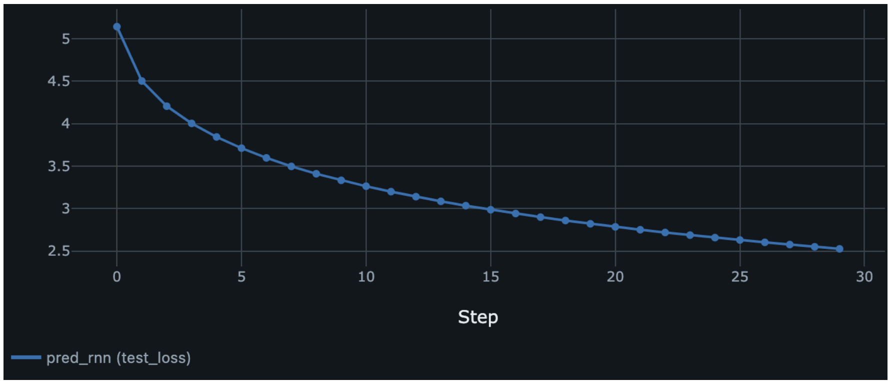

Figure 4: PredRNN test loss over epochs

6.3 PredRNN

PredRNN achieved an RMSE of 1.59 and took 2.2 hours to train. The model's parameters are in Table 6. A model summary text file generated by PyTorch can be found in our repository.

Parameter	Value
num_layers	1
num_hidden	64
patch_size	1
img_channel	1
img_width	67
filter_size	5
stride	1
layer_norm	False
device	'mps'
total_length	9
input_length	1
reverse_scheduled_sampling	1
lr	0.0001

Table 6: PredRNN Parameters

7 Discussion

ConvLSTM outperformed both HA and PredRNN in terms of predictive accuracy, albeit at the cost of longer training times. PredRNN, while slightly less accurate than ConvLSTM, offers a competitive alternative with faster training times. The choice between these models should consider the trade-off between computational resources and performance metrics, aligning with specific application requirements.

As seen from Figures 2 and 3, the test loss kept decreasing over time, but due to hardware limitations we only ran the training for both models for 30 epochs.

8 Conclusion

In this paper, we explored the application of next-frame video generation models for predicting Origin-Destination (OD) taxi demand matrices. We demonstrated that these models, which have shown success in domains such as human action prediction and autonomous driving, are also effective in capturing the spatio-temporal dynamics inherent in taxi demand data. By treating OD matrices as frames and utilizing the temporal sequence of these matrices, next-frame prediction models offer a robust alternative to traditional graph-based methods. The analysis provided insights into the strengths and limitations of two next-frame prediction models in the context of OD taxi demand forecasting. Additionally, we developed and tested a real-life pipeline, addressing the gap between research and practical implementation.

8.1 Limitations

- Hardware limitations causing slow training.
- Our analysis uses only data from Manhattan, New York, so the performance in regions with sparse adjacency matrices is uncertain.

8.2 Future Directions

- Explore additional models.
- Experiment with different models and hyperparameters.
- Utilize datasets from various regions.

References

[CLL+23] Jinguo Cheng, Ke Li, Yuxuan Liang, Lijun Sun, Junchi Yan, and Yuankai Wu. Rethinking urban mobility prediction: A super-multivariate time series forecasting approach. *arXiv preprint arXiv:2312.01699*, 2023.

[DP23] Zhibo Xing Dan Peng, Mingxia Huang. Taxi origin and destination demand prediction based on deep learning: a review, 2023.

[GLF+19] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 922–929, 2019.

[GTWL22] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3170–3180, 2022.

[HZWY22] Ziheng Huang, Weihan Zhang, Dujuan Wang, and Yunqiang Yin. A gan framework-based dynamic multi-graph convolutional network for origin–destination-based ride-hailing demand prediction. *Information Sciences*, 601:129–146, 2022.

[JCZ+20] Guangyin Jin, Yan Cui, Liang Zeng, Hanbo Tang, Yanghe Feng, and Jincai Huang. Urban ride-hailing demand prediction with multiple spatio-temporal information fusion network. *Transportation Research Part C: Emerging Technologies*, 117:102665, 2020.

[JKK+24] Huiwon Jang, Dongyoung Kim, Junsu Kim, Jinwoo Shin, Pieter Abbeel, and Younggyo Seo. Visual representation learning with stochastic frame prediction. *arXiv preprint arXiv:2406.07398*, 2024.

[JL] Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey.

[JXS+20] Guangyin Jin, Zhexu Xi, Hengyu Sha, Yanghe Feng, and Jincai Huang. Deep multi-view spatiotemporal virtual graph neural network for significant citywide ride-hailing demand prediction. *arXiv preprint arXiv:2007.15189*, 2020.

[KBE+20] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A conditional flow-based model for stochastic video generation, 2020.

[KW18] Dejiang Kong and Fei Wu. Hst-lstm: A hierarchical spatial-temporal long-short term memory network for location prediction. In *IJCAI*, volume 18, pages 2341–2347, 2018.

[KYZ+19] Jintao Ke, Hai Yang, Hongyu Zheng, Xiqun Chen, Yitian Jia, Pinghua Gong, and Jieping Ye. Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services. *IEEE Transactions on Intelligent Transportation Systems*, 20(11):4160–4173, 2019.

[LC23] Chenming Li and Xiuhong Chen. Future video frame prediction based on generative motion-assistant discriminative network. *Applied Soft Computing*, 135:110028, 2023.

[LCYL17] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks, 2017.

[LLM+24] Jianbo Li, Zhiqiang Lv, Zhaobin Ma, Xiaotong Wang, and Zhihao Xu. Optimization of spatial-temporal graph: A taxi demand forecasting model based on spatial-temporal tree. *Information Fusion*, 104:102178, 2024.

[LSY+22] Aling Luo, Boyi Shangguan, Can Yang, Fan Gao, Zhe Fang, and Dayu Yu. Spatial-temporal diffusion convolutional network: A novel framework for taxi demand forecasting. *ISPRS International Journal of Geo-Information*, 11(3):193, 2022.

[LTCL24] Cheng-Te Li, Yu-Che Tsai, Chih-Yao Chen, and Jay Chiehen Liao. Graph neural networks for tabular data learning: A survey with taxonomy and directions, 2024.

[LWZT20] Tong Liu, Wenbin Wu, Yanmin Zhu, and Weiqin Tong. Predicting taxi demands via an attention-based convolutional recurrent neural network. *Knowledge-Based Systems*, 206:106294, 2020.

[LYSL17] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[LZD+18] Guan Lian, Yaping Zhang, Jitamitra Desai, Zhiwei Xing, and Xiao Luo. Predicting taxi-out time at congested airports with optimization-based support vector regression methods. *Mathematical Problems in Engineering*, 2018(1):7509508, 2018.

[Ope] OpenAI. OpenAI SORA. https://openai.com/index/sora/. Accessed: 2024-06-18.

[PHC16] Viorica Patraucean, Ankur Handa, and Roberto Cipolla. Spatio-temporal video autoencoder with differentiable memory. *arXiv preprint arXiv:1511.06309*, 2016.

[RVA+] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer.

[SBK+] Jiahao Su, Wonmin Byeon, Jean Kossaifi, Furong Huang, Jan Kautz, and Animashree Anandkumar. Convolutional tensor-train lstm for spatio-temporal learning.

[SCW+] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting.

[SDVB16] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks, 2016.

[Sta24] Statista. Taxi - worldwide — statista market forecast, 2024. Accessed: 2024-06-18.

[STT] Jingran Shen, Nikos Tziritas, and Georgios Theodoropoulos. A baselined gated attention recurrent network for request prediction in ridesharing.

[SYG+20] Hongzhi Shi, Quanming Yao, Qi Guo, Yaguang Li, Lingyu Zhang, Jieping Ye, Yong Li, and Yan Liu. Predicting origin-destination flow via multi-perspective graph convolutional network. In *2020 IEEE 36th International conference on data engineering (ICDE)*, pages 1818–1821. IEEE, 2020.

[TGLL] Cheng Tan, Zhangyang Gao, Siyuan Li, and Stan Z. Li. Simvp: Towards simple yet powerful spatiotemporal predictive learning.

[TGW$^+$23] Cheng Tan, Zhangyang Gao, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, and Stan Z Li. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18770–18782, 2023.

[TLZT23] Song Tang, Chuang Li, Pu Zhang, and RongNian Tang. Swinlstm: Improving spatiotemporal prediction accuracy using swin transformer and lstm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13470–13479, 2023.

[VSP$^+$17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[WHK$^+$24] Thomas Winterbottom, G Thomas Hudson, Daniel Kluvanec, Dean Slack, Jamie Sterling, Junjie Shentu, Chenghao Xiao, Zheming Zhou, and Noura Al Moubayed. The power of next-frame prediction for learning physical laws. *arXiv preprint arXiv:2405.17450*, 2024.

[WTU] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models.

[WWZ$^+$] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip S. Yu, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning.

[WYC$^+$19] Yuandong Wang, Hongzhi Yin, Hongxu Chen, Tianyu Wo, Jie Xu, and Kai Zheng. Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1227–1235, 2019.

[WYC$^+$21] Yuandong Wang, Hongzhi Yin, Tong Chen, Chunyang Liu, Ben Wang, Tianyu Wo, and Jie Xu. Passenger mobility prediction via representation learning for dynamic directed and weighted graph, 2021.

[WZSL23] Ning Wang, Liang Zheng, Huitao Shen, and Shukai Li. Ride-hailing origin-destination demand prediction with spatiotemporal information fusion. *Transportation Safety and Environment*, 6(2):tdad026, 2023.

[WZZ$^+$] Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics.

[XRBT17] Jun Xu, Rouhollah Rahmatizadeh, Ladislau Bölöni, and Damla Turgut. Real-time prediction of taxi demand using recurrent neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 19(8):2572–2581, 2017.

[XSLW18] Ke Xu, Luping Sun, Jingchen Liu, and Hansheng Wang. An empirical investigation of taxi driver response behavior to ride-hailing requests: A spatio-temporal perspective. *PloS one*, 13(6):e0198605, 2018.

[XWLW18] Ziru Xu, Yunbo Wang, Mingsheng Long, and Jianmin Wang. Predcnn: Predictive learning with cascade convolutions. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2940–2947. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[YB24] Xi Ye and Guillaume-Alexandre Bilodeau. Stdiff: Spatio-temporal diffusion for continuous stochastic video prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6666–6674, 2024.

[ZFWQ20] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1234–1241, 2020.

[ZSZ$^+$] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutionalnetwork for traffic prediction.

[ZSZL] Jiawei Zhu, Yujiao Song, Ling Zhao, and Haifeng Li. A3t-gcn: Attention temporal graph convolutional network for traffic forecasting.

[ZZL$^+$22] Chizhan Zhang, Fenghua Zhu, Yisheng Lv, Peijun Ye, and Fei-Yue Wang. Mlrnn: Taxi demand prediction based on multi-level deep learning and regional heterogeneity analysis. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8412–8422, 2022.

[ZZQ17] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.