Traffic4cast at NeurIPS 2020 – yet more on the unreasonable effectiveness of gridded geo-spatial processes

Michael Kopp MICHAEL.KOPP@IARAI.AC.AT
HERE Technologies Zürich & IARAI, Vienna

David Kreil DAVID.KREIL@IARAI.AC.AT
Institute of Advanced Research in Artificial Intelligence (IARAI), Vienna, Austria

Moritz Neun MORITZ.NEUN@HERE.COM
HERE Technologies Zurich

David Jonietz DAVID.JONIETZ@HERE.COM
HERE Technologies Zurich

Henry Martin HENRY.MARTIN@IARAI.AC.AT
ETH Zürich & IARAI, Vienna

Pedro Herruzo PEDRO.HERRUZO@IARAI.AC.AT
Institute of Advanced Research in Artificial Intelligence (IARAI), Vienna, Austria

Aleksandra Gruca ALEKSANDRA.GRUCA@POLSL.PL
Department of Computer Networks and System, Silesian University of Technology, Gliwice, Poland

Ali Soleymani ALI.SOLEYMANI@HERE.COM
HERE Technologies Zurich

Fanyou Wu WU1297@PURDUE.EDU
Purdue University, Department of Forestry and Natural Resource, West Lafayette, USA

Yang Liu 230179629@SEU.EDU.CN
Southeast University, School of Transportation, Nanjing, China

Jingwei Xu XUJINGWEI1995@GMAIL.COM
Shanghai Jiao Tong University, Shanghai, China

Jianjin Zhang JIANJZH@MICROSOFT.COM
Microsoft, Beijing, China

Jay Santokhi JAY@ALCHERATECHNOLOGIES.COM
Alchera Data Technologies Ltd, Cambridge, UK

Alabi Bojesomo 100046384@KU.AC.AE
Electrical Engineering and Computer Science Department, Khalifa University, Abu Dhabi, UAE

Hasan Al Marzouqi HASAN.ALMARZOUQI@KU.AC.AE
Electrical Engineering and Computer Science Department, Khalifa University, Abu Dhabi, UAE

Panos Liatsis PANOS.LIATSIS@KU.AC.AE
Electrical Engineering and Computer Science Department, Khalifa University, Abu Dhabi, UAE

Pak Hay Kwok PAK_HAY_KWOK@HOTMAIL.COM

Qi Qi QIQ208@GMAIL.COM

Sepp Hochreiter SEPP.HOCHREITER@IARAI.AC.AT
Institute of Advanced Research in Artificial Intelligence (IARAI), Vienna, Austria

Editors: Hugo Jair Escalante and Katja Hofmann

Abstract

The IARAI *Traffic4cast* competition at NeurIPS 2019 showed that neural networks can successfully predict future traffic conditions 15 minutes into the future on simply aggregated GPS probe data in time and space bins, thus interpreting the challenge of forecasting traffic conditions as a movie completion task. U-nets proved to be the winning architecture then, demonstrating an ability to extract relevant features in the complex, real-world, geo-spatial process that is traffic derived from a large data set. The IARAI *Traffic4cast* challenge at NeurIPS 2020 build on the insights of the previous year and sought to both challenge some assumptions inherent in our 2019 competition design and explore how far this neural network technique can be pushed. We found that the prediction horizon can be extended successfully to 60 minutes into the future, that there is further evidence that traffic depends more on recent dynamics than on the additional static or dynamic location specific data provided and that a reasonable starting point when exploring a general aggregated geo-spatial process in time and space is a U-net architecture.

1. Introduction

In our first edition of our *Traffic4cast* competition at NeurIPS 2019, we encouraged contestants to predict traffic flow volumes, velocities, and dominant flow directions 15 minutes into the future on a unique, large, real world data set (Kreil et al., 2019). Although such forecasts are thought to be the basis for building and managing our cities to provide efficient and sustainable mobility (Bucher et al., 2019; Lee et al., 2018; Jonietz et al., 2018), this form of traffic prediction is still largely considered to be an unsolved problem (Guo et al., 2019). An innovation of our *Traffic4cast* 2019 competition was the chosen representation: we aggregated our traffic data from individual sensor measurements in space and time bins. The values of the spatial representation of each time bin could be interpreted and visualized as a ‘movie’ frame and thus our traffic prediction task was effectively a video frame prediction task. This is a highly active field with promising distinct approaches (Srivastava et al., 2015; Lee et al., 2018; Kwon and Park, 2019; Walker et al., 2016; Xue et al., 2016; Han et al., 2019; Oprea et al., 2020) which we hoped could be harvested by the traffic prediction community as well. Our 2019 competition yielded the following insights (Kreil et al., 2019).

- Re-phrasing traffic forecasting as a video prediction problem turned out to be of merit and – as independent work has shown on precipitation prediction (Agrawal et al., 2019) – should be considered a promising new technique in tackling complex geo-spatial processes. Capturing the complex spatio-temporal dependencies of such processes is known to be a hard problem, usually referred to as ‘spatial is special’ (Anselin, 1989), and our 2019 competition contributed evidence that neural network techniques in this simple video frame prediction setting yield promising results.

- Anecdotal evidence stemming from the attempt by some contestants indicated that trying to add additional prior static or dynamic location knowledge about the spatial bins did not seem to improve results significantly (Martin et al., 2019; Herruzo and Larriba-Pey, 2020), suggesting that such information was already encoded in the complex traffic data itself. Moreover, no successful strategies in our competition used recurring traffic patterns based on time of day or day of week, which would have been possible in the competition design, providing further anecdotal evidence that traffic is ‘quasi-markovian’, i.e. mostly dependent on the immediate past only.

- Most successful solutions indicated (Choi, 2019; Martin et al., 2019; Yu et al., 2019) that our discretized 'majority heading' channel was the least informative and was also hardest to predict.

For the *Traffic4cast* competition at NeurIPS 2020, we set out to examine further some of the findings discussed and challenge our assumptions in the design of the 2019 competition. For instance, the literature presents empirical evidence that the performance of numerous traffic prediction models would typically decrease significantly past a 15min horizon, and a majority of studies of 'short-term prediction' thus focus on such time horizons (Ermagun and Levinson, 2018a; Dunne and Ghosh, 2011; Ermagun and Levinson, 2018b; Lana et al., 2018). In line with common practice, we therefore restricted our 2019 competition to that horizon. The great success of neural techniques in extracting complex traffic patterns from our *Traffic4cast* 2019 large real world competition data in our chosen representation and the fact that most of the submitted solutions could easily be extended to predict past that time horizon, led us in our 2020 competition to test that hypothesis by asking participants to predict up to 60min into the future. Given the unexpected success of phrasing our traffic forecasting problem as a movie prediction task, we decided to maintain our simple spatial and temporal aggregation approach, but modify it due to the findings above. Furthermore, we wanted to double down on the implicit question thrown up by Martin et al. (2019) and Bucher et al. (2019) of whether additional static and dynamic data potentially relevant to the geo-spatial process of traffic could improve predictions. Here, we set out these ideas and learnings from our *Traffic4cast* competition at NeurIPS 2020. In the following section we describe our competition in detail including our new data format. In section 3 a short presentation of selected competition entries is given. We finally conclude with a summary of what we learned from *Traffic4cast* at NeurIPS 2020 and provide an outlook to possible exciting next steps in the coming year.

2. Competition Overview

2.1. The data

Building up-on the previous *Traffic4cast* 2019 competition (Kreil et al., 2019) we again provided a unique real-world industrial-scale data set. In addition to dynamic channels derived from trajectories of raw GPS position fixes (consisting of a latitude, a longitude, a time stamp, as well as the vehicle speed and driving direction recorded at the time) we also provided a dynamic channel containing information about incidents such as accidents, construction sites and other events that influence traffic. In addition, for each city, static channels that give additional contextual information about the densities of certain POI (point-of-interest) categories as well as about general road network topology and complexity properties was provided. The data is made available by HERE Technologies (`www.here.com`) with the GPS data originating from a large fleet of vehicles recorded throughout the course of the entire year 2019.

An unprecedented number of over 10^{11} GPS points were used to generate the data sets for Berlin, Moscow, and Istanbul. These three cities represent culturally and socially diverse metropolitan areas and, furthermore, they are also exhibiting a broad variety of road network topologies as it can be seen in Figure 1.

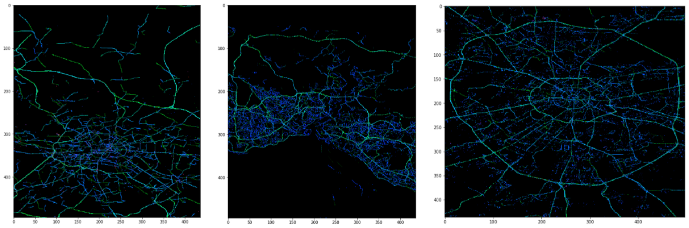


Figure 1: The cities Berlin, Istanbul and Moscow.

We aggregated the GPS measurements in spatial cell bins of approximately 100x100 meters size and 5-minute time bins. Figure 2 shows a simplified spatial tessellation as well as typical scenarios of what portions of a road network such a 100x100 meter cell contains in a dense city area. The output of this aggregation can be encoded as described below as an '8 channel movie' with each 5 minute bin representing a time frame and with the densities and other aggregate features of the cells being mapped to generalized pixel values in different channels. Hence, as output we receive such a generalized movie with 288 frames for a single day.

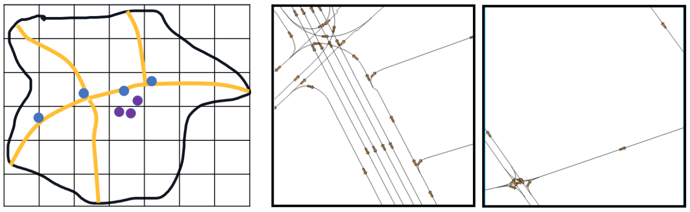


Figure 2: Spatial tessellation of the road network and GPS recordings.

2.1.1. Dynamic channels

Unlike the *Traffic4cast* at NeurIPS 2019 competition where we aggregated GPS features using 3 channels (Volume, mean speed and main heading) and hence even allowed an easy visualization treating the three channels as RGB values, this year we chose an 8-channel encoding (see Figure 3) where two features are calculated for each heading direction quadrant of North-East (0-90), South-East (90-180), South-West (180-270) and North-West (270-0):

- Volume: The number of probes points recorded from the collection of HERE sources capped both above and below and normalized and discretized to an integer number between 0 and 255.
- Mean speed: The average speed from the collected probe points. The values are capped at a maximum level and then discretized to $\{1, 2, \ldots, 255\}$, by linearly scaling the capping speed to 255 and rounding the resulting values to the nearest integer.

In addition, we provided a 9th dynamic channel with the incident level for each cell and time bin. This channel was generated using information from the HERE Traffic API

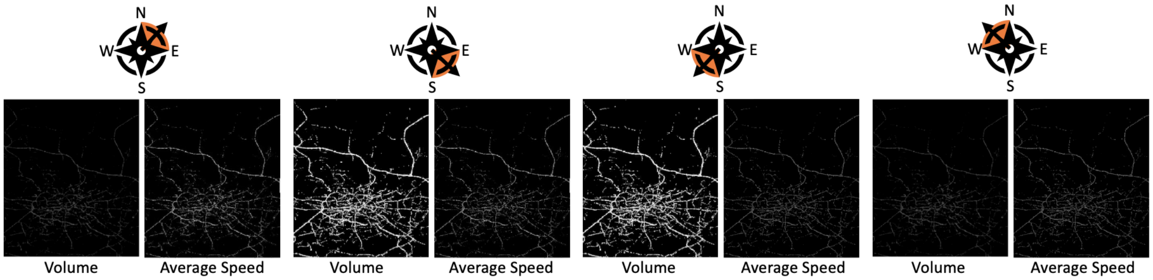
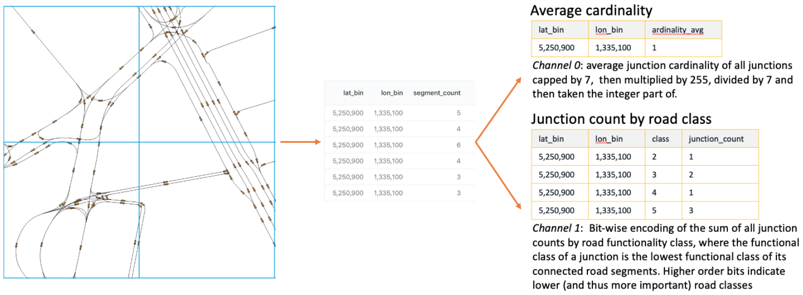


Figure 3: The 8 probe channels, 2 for each heading quadrant.

Incident Data which provides dynamic information both about short and long-term incidents such as accidents, congestion, road hazards, road closure, construction, planned events or weather. The incident level for the channel was encoded as follows. We used a static mapping of incident type and the incident criticality (minor, major, critical), then aggregated these over the same $100m \times 100m \times 5min$ spatio-temporal bins as used for the GPS data and, finally, mapping the sum of the resulting outcome into the discrete values $\{0, 50, 60, 70, \ldots, 240, 250, 255\}$ in such a way that a higher number represents more severe incidents.

2.1.2. Static channels

For the static channels only a spatial tessellation into the 100x100 meter cells was applied per city in order to encode contextual information about the cell. Thus, per city, 5 channels are provided to encode major POI (point-of-interest) categories. These heatmaps were generated using normalized counts of places in the HERE Places database (https://developer.here.com/documentation/places/dev_guide/topics/categories.html). The categories, selected to represent the predominant usage and activities in a cell or in the vicinity, were 'Eat, drink and entertainment', 'Hospital', 'Parking', 'Shopping' and 'Transport', respectively. Two further channels were generated to represent the general topology and complexity of the road network in each cell. The junction cardinality as well as the junction count per road class can be seen as a proxy to differentiate cells with e.g. a single complex junction from a cell with many small junctions. A simplified schema of the aggregation and encoding can be seen in Figure 4.



lat_bin	lon_bin	segment_count
5,250,900	1,335,100	5
5,250,900	1,335,100	4
5,250,900	1,335,100	6
5,250,900	1,335,100	4
5,250,900	1,335,100	3
5,250,900	1,335,100	3

Average cardinality

lat_bin	lon_bin	ardinality_avg
5,250,900	1,335,100	1

Channel 0: average junction cardinality of all junctions capped by 7, then multiplied by 255, divided by 7 and then taken the integer part of.

Junction count by road class

lat_bin	lon_bin	class	junction_count
5,250,900	1,335,100	2	1
5,250,900	1,335,100	3	2
5,250,900	1,335,100	4	1
5,250,900	1,335,100	5	3

Channel 1: Bit-wise encoding of the sum of all junction counts by road functionality class, where the functional class of a junction is the lowest functional class of its connected road segments. Higher order bits indicate lower (and thus more important) road classes

Figure 4: The road topology and complexity channels.

2.1.3. Output and data provisioning

Data was made available in HDF5 format. For each city, the data consisted of

- One file (h5) encoding a $(495, 436, 7)$ tensor of static information
- The dynamic part of the training set consisting of 181 dynamic layer files (h5) each containing a $(288, 495, 436, 9)$ tensor.
- The validation set contains 18 full days in the same format.
- The test set asking to make 500 predictions spread over 163 days for each city of the dynamic probe data portion, thus predicting 8-channels and 6 time slots per prediction (5min, 10min, 15min, 30min, 45min and 60min into the future). The test times, buffered by a 3h period, were randomly distributed across all possible time slots in these 163 test days as shown in Figure 5.

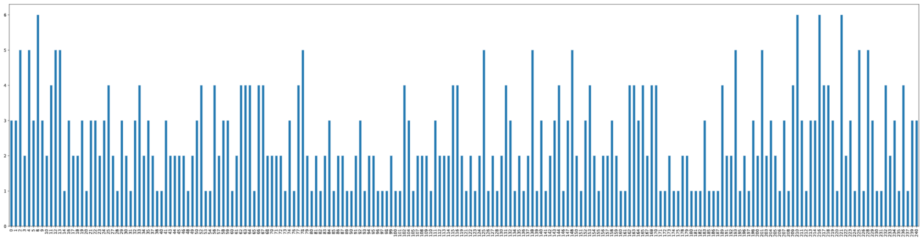


Figure 5: Distribution of number of test time slots chosen for each day in our test set.

2.2. Task and metric

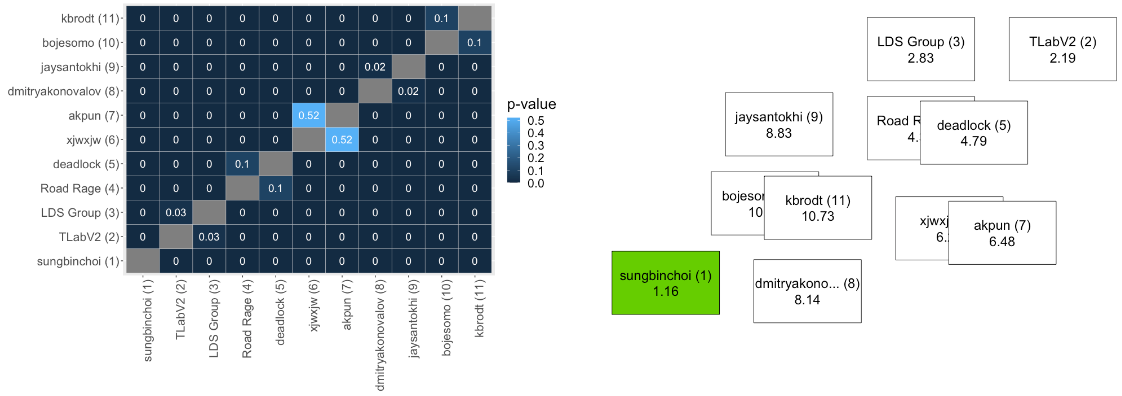
As outlined above, for each city contestants had to predict for 500 time slots (same for all cities), given the last 12 time bins (corresponding to one hour) of dynamic data (tensor for shape $(12, 495, 436, 9)$) a tensor of shape $(6, 495, 436, 8)$ representing predicting the time bins 5min, 10min, 15min, 30min, 45min and 60min into the future. The metric used was average mean squared error (MSE).

2.3. Competition results - statistical analysis

We determined the ultimate ranking for our *Traffic4cast* competition by assigning to each team its best leaderboard score. Since the differences among top scoring submissions were very small we were keen to understand whether the observed ranking was statistically significant. This problem can be framed as a problem of comparing the performance of multiple algorithms over multiple datasets; a problem for which the solution is already well established in the machine learning community. As suggested by the work of Demšar (2006), in order to assess if there are any significant differences among submissions we can apply the Friedman rank test (Friedman, 1937). Since the result of the Friedman test does not indicate which submissions are statistically different from each other, if it rejects the null hypothesis, we subsequently need to apply a mean rank post-hoc test for pairwise comparison between submissions. Multiple testing correction is required for the typically large number of pairwise comparisons to be made (Garcia and Herrera, 2008).

Typically, the competition leaderboard includes one score for each team (submission) and obviously no meaningful statistical analyses can be performed having only single score per team. However, having access to individual scores per city and test slots we can obtain a set of scores large enough to overcome the otherwise low power of the Friedman test. Two main assumptions of the Friedman test are block test independence and homogeneity of variance between blocks (Laurent and Turk, 2013; Pereira et al., 2015).

In order to assure that the Friedman test assumptions are fulfilled, we selected test days having at least two predictions, which gave us 127 test days. For each selected day we then randomly choose two prediction times. By averaging over 6 prediction time bins and over the 3 cities we obtained a total of 254 independent scores, which were then used to assess the statistical significance of the *Traffic4cast* leaderboard. The results of the Friedman test (p-value $<$ 2.2e-16) allowed us to reject the null hypotheses indicating that the score differences among submissions are indeed not random. Results of the mean rank post-hoc test show that scores for three top ranked submissions are statistically significantly different from one another as well as from the runner up on rank 4, fully justifying the competition awards to the three best ranked teams. The p-values of the pairwise post-hoc test, after applying the Benjamini-Hochberg multiple testing correction procedure (Benjamini and Hochberg, 1995), can be observed in the left figure of Figure 6. Most submissions ranked outside the top 3 fall into pairs of submissions with performance scores so similar that they cannot be called significantly different and thus cannot be distinguished reliably, as presented in the right figure of Figure 6.



	sungbinchoi (1)	TLabV2 (2)	LDS Group (3)	Road Rage (4)	deadlock (5)	xjwxjw (6)	akpun (7)	dmitryakonovalov (8)	jaysantokhi (9)	bojesomo (10)	kbrodt (11)
kbrodt (11)	0	0	0	0	0	0	0	0	0	0.1	
bojesomo (10)	0	0	0	0	0	0	0	0	0		0.1
jaysantokhi (9)	0	0	0	0	0	0	0	0.02		0	0
dmitryakonovalov (8)	0	0	0	0	0	0	0		0.02	0	0
akpun (7)	0	0	0	0	0	0.52		0	0	0	0
xjwxjw (6)	0	0	0	0	0		0.52	0	0	0	0
deadlock (5)	0	0	0	0.1		0	0	0	0	0	0
Road Rage (4)	0	0	0		0.1	0	0	0	0	0	0
LDS Group (3)	0	0.03		0	0	0	0	0	0	0	0
TLabV2 (2)	0		0.03	0	0	0	0	0	0	0	0
sungbinchoi (1)		0	0	0	0	0	0	0	0	0	0

Figure 6: Left figure represents p-values of the pairwise post-hoc test between each submission. The right figure is a graphical representation of statistical similarity among submissions. Separated nodes represent submissions statistically different from the others, while overlapping nodes represent submissions where the null hypothesis of being equal cannot be rejected (p-value cutoff = 0.05). The number in brackets represent rank of a submission (team) and the number below team name represents mean rank of the submission across scores.

3. Standout solutions

3.1. Utilizing U-net for the Future Traffic Map Prediction

Our winning submission utilized a U-net (Ronneberger et al., 2015) based model with skip connections on the encoder and decoder side. Such architectures have been widely used in various tasks including image classification and segmentation (Li et al., 2018; Weng et al., 2019; Chen et al., 2018; Zhou et al., 2019; Qamar et al., 2020). Training was done using MSE and the Adam optimizer scheme (Kingma and Ba, 2014). The final outcome was the combination of an ensemble of three U-net models (Model 1,2 and 3) whose architecture was found empirically.

3.1.1. Encoder structure

Model 1 has a similar structure to our last year's experimentation (Choi, 2019). In each encoding block, each convolution layer is densely connected to every other layer in a feed forward fashion (Huang et al., 2017). In Model 2, the max pooling layer is attached in parallel to the dense convolution layers and both their outputs are concatenated before being fed into the last convolution layer. At the end of each block, a convolutional pooling layer is used instead of an average pooling layer. In Model 3, the max pooling layer is added firsthand, followed by densely connected convolutional layers.

3.1.2. Decoder structure

In Model 1 and 2, each decoding block has one deconvolution layer, followed by one convolution layer.

In Model 3, linear interpolation layer is attached in parallel to the deconvolution layer and both their outputs are concatenated before being fed into the densely connected convolutional layers.

3.1.3. Combining three U-net models prediction

The predictions from three U-net models were combined by averaging. Although every single model showed practically similar performance, each model likely captured different aspects of the ground truth that leading to complementary features being considered. Combining predictions like this likely reduces the error, assuming each model has a separate, independent error distribution.

3.2. Traffic Map Movie Forecasting Based on HR-NET

By interpreting our challenge as a multi-task learning problem of predicting two channels in each direction, our second placed team found a non Unet based approach (see Wu et al. (2020a) and code at `https://github.com/wufanyou/Traffic4Cast-2020-TLab`).

3.2.1. Models

HR-NET as model architecture: HR-NET (Wang et al., 2020) was introduced to our competition, where HR-NET is an advanced network architecture for image segmentation that has demonstrated extraordinary performance on many tasks. Based on our experiments,

we can conclude that, in general, HR-NET performs better than Unet (Ronneberger et al., 2015).
ELU as hidden layer activation function: Given limited RAM, the focus was on in-place activation functions only. ELU performed surprisingly better than any other activation function. Therefore, ELU was used as the hidden layer activation function in most places of the final solution.

3.2.2. Features

In addition to the given 12×9 spatio-temporal channels and nine fixed spatial channels as the inputs, extra features were also incorporated and valued. In summary, we introduce three different types of features:
Periodic features: The similarity between traffic states of two different days can be attributed to the periodic characteristics of traffic states, which typically repeat every 24 hours. In total, 10×8 daily average statistics from $\{D-7\} \bigcup [D-3, D-1] \bigcup [D+1, D+3] \bigcup \{D+7\}$ were used, where D is the predicted day.
Time, Weekday and Holiday Features: Time, weekday, and holiday features are definitely useful in traffic flow prediction tasks. We used a two-dimensional vector to represent time in one day by projecting $[0, 287]$ to a unit circle. A one-hot vector to represent weekday and a Boolean value to represent the holiday was used.
Geo-Embedding features: Based on work in (Liu et al., 2020), embedding technique to generate regional 'personalized' temporal information and feed it into the convolutional neural network were employed. In this competition, a further method of geo-embedding to learn the inherent attributes of each location (*i.e.*, pixel) was proposed. A learnable tensor $C \times N \times M$ was concatinated to each input and optimized using the model.

3.2.3. Training

Due to device issues, most of the models were trained on a mini-batch size of 3×4 with $3\times$ 2080Ti GPU. Each model was trained for 15 epochs with an initial learning rate of 0.01 and a linear learning rate decay. The LAMB (You et al., 2020) optimizer and warm-up train was mainly found to yield better outcomes. Model training typically took 16-20 hours even with using SyncBatchNorm to stabilize and speed up the training. Also, in the final solution, half of the models are trained by both the training set and the validation set.

3.3. Towards Good Practices of U-net for Traffic Forecasting

One competitor, the LDS group, considered the traffic forecasting problem as a future frame prediction task with relatively weak temporal dependencies (due to stochastic urban traffic dynamics) and strong prior knowledge, *i.e.*, the roadmaps of the cities. For these reasons, a U-net as the backbone model was used, and a roadmap generation method proposed to make the predicted traffic flows more consistent. A fine-tuning strategy based on the validation set was used to prevent over-fitting, effectively improving the prediction results. The code is available at `https://github.com/ZJianjin/Traffic4cast2020_LDS`.

3.3.1. Approach

Network architecture. Based on the winning U-net (Ronneberger et al., 2015) architecture of *Traffic4cast* 2019. It consists of 8 dense blocks and 7 transpose convolutional layers. The output of the n-th dense block will be input to the $(7 - n)$-th transpose convolutional layer, which is a shortcut to retrieval of more information from the original image. A similar shortcut is used inside the dense blocks: each dense block has 4 convolutional layers, each layer will take the output of the previous layer and the original input of the dense block as inputs, following Huang et al. (2017). Dense blocks are followed by average pooling to half the size of the feature maps. The last dense block is followed by a single convolutional layer.

Two-stage training strategy. The pattern of traffic flow might be very different during different seasons and, in particular, between the period of the training data provided (January to June 2019) and the period of the test set (July to December 2019). Seasonal information to facilitate remedy that was provided, which, however, got no obvious gain. Noting that the validation set is from the second half of the year, a two-stage training strategy was employed. After pre-trained on the training set, the model will be fine-tuned on validation data.

3.3.2. Training

The mean square error was chosen as the loss function during training and Adam optimization with a $3e^{-4}$ learning rate was used. For each city, we have 181 train data files which contain traffic data of half a year, and 18 files for validation. The 12 consequent frames were used as inputs and predict 6 frames as the outputs. In addition to the original frames, the static data, which represents road intersections, were added as additional input features. All the input frames are normalized to 0–1.

For each city an independent model was trained with the two-stage training process outlined above. All of them were trained for 5 epochs and then fine-tuned for 1 epoch.

3.4. Graph Ensemble Net and the Importance of Feature & Loss Function Design for Traffic Prediction

The work submitted by Qi and Kwok (see Qi and Kwok (2020) and code `https://github.com/ivans-github/traffic4cast`) explores two different approaches to tackling the traffic prediction challenge. Firstly, they enhanced the winning U-net solution from last year (Choi, 2019) through improved feature and loss function design. Furthermore, they proposed a novel ensemble Graph Neural Network (GNN) architecture which, although unable to compete with the U-net solution, improved on the Graph-ResNets (Martin et al., 2020) from last year's competition.

3.4.1. Models

U-net: Having experimented with several U-net derivative architectures, the final U-net largely resembled last year's top-performing solution (Choi, 2019). Modifications were introduced mainly around the input and output layers to handle the increased numbers of input and output channels.

Graph Ensemble Net: We introduce a GNN architecture which takes advantage of the wide-ranging type of convolutional filters that were developed in recent years. To implement this, we adapted the Graph-ResNet (Martin et al., 2020), replacing the res-block with a new ensemble block. In the ensemble block, the inputs are passed through three different convolutional filters (Defferrard et al., 2016; Hamilton et al., 2017; Wu et al., 2019) with the outputs averaged before being fed into the next layer/block.

3.4.2. Features

Aside from collapsing the temporal dimension of the inputs into channels (resulting in a channel size of 108), some key feature designs which we implemented are summarised as:
Time of Day features: Since we know that there are large variations in traffic behaviour during the day (e.g. commuting hours), we decided to create an additional time of day feature which encoded the starting hour of the input.
Aggregation Features: For the U-net, we added extra features encoding the pixel-wise range, mean and standard deviations across the 12 timeframes. For the GNN, we condensed the data for the first 6 timeframes into mean, min and max features, which reduced the model size without losing performance.

3.4.3. Loss Function

The competition called for predicting 6 forward prediction time frames at 5, 10, 15, 30, 45 and 60 minutes. However, during training, a loss function calculated from all 12 frames across the 60-minute prediction window was used which enhanced the predictive accuracy.

3.4.4. Training

Most of the training was done on Paperspace notebook using a single P5000 GPU

3.5. Temporal Autoencoder for Frame Prediction

Santokhi et al. (2020)'s approach was to see our competition challenge as an image-to-image translation problem and they developed a Temporal Autoencoder using a U-Net style architecture (see also `https://gitlab.com/alchera/traffic4cast_2020`)

3.5.1. Model Architecture

In order to capture both spatial and temporal features, Convolutional LSTM layers (Shi et al., 2015) were used to build up the Autoencoder structure with 3 layers acting as the encoder and 3 acting as the decoder. Down-sampling via 3D Max-Pooling was used to reduce dimensionality and in turn ease training workload given compute resources available. Transposed Convolutions were used over standard interpolation up-sampling methods allowing the network to learn the up-sampling optimally via learnable parameters. To ensure topological features of the cities were not lost from down-sampling to a latent space and subsequent reconstructing via Transposed Convolutions, skip-connections were added. These were utilised in a similar way to how they are presented in standard U-Net architectures. The Temporal Autoencoder architecture can be seen in Figure 7 with the tensor shapes given at each stage `(frames, height, width, channels)`.

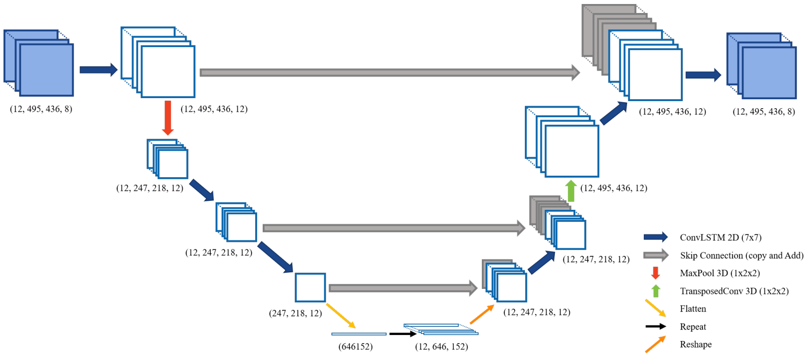


Figure 7: Model Architecture

3.5.2. Training

The model described was trained using 4 NVIDIA 1080 Titans with a batch size of 4 for 28 epochs taking a total of 9.5 hours. An ADAM optimiser paired with a cyclical learning rate (Smith, 2017) proved to be the optimal approach. A total of seven triangular oscillations (with minimum and maximum learning rates of 1×10^{-7} and 2×10^{-3} respectively) were made during the training cycle. This approach led to improved training efficiency, achieving lower loss scores in fewer epochs than standard approaches.

Given each city has significantly different topological features, driving behaviour and driving culture stemming from being in different countries, three separate models were trained; one for each city as it did not seem wise to encompass these factors into a single 'all-in-one' model. Model architecture, training procedure and hyper-parameters were kept exactly the same.

3.6. Traffic Flow Prediction Using Deep Sedenion Networks

The creators of this solution (see (Bojesomo et al., 2020)) used the spatio-temporal nature of the problem thus viewing it as a multi-modal and multi-task problem due to the variety of information presented in input channels and the need to forecast more than one time step in the future. Hypercomplex networks provide efficient means for representing multi-modal data. They have been previously used for solving multi-task problems (Wu et al., 2020b) with promising results. Hypercomplex networks are often used when the number of input channels (i.e. components) match or less than the dimensionality of hypercomplex numbers (Parcollet et al., 2019). This section describes the use of sedenion convolutions in a U-Net based model for tackling the multitask segmentation problem. The proposed model was described in more details in (Bojesomo et al., 2020) and implemented in `https://github.com/bojesomo/Traffic4Cast2020-DeepSedanionNetwork`. This model having just **628,592** trainable parameters achieved competitive result while being trained with limited resources.

3.6.1. Sedenion Convolution

A sedenion can be defined as a 16-dimensional algebraic structure, i.e., $X = x_0 + \sum_{k=1}^{15} x_k i_k$, where x_0 is the real part and $x_k, k \in (1, 15)$ are the imaginary parts with $i_k^2 = -1$. Sedenion based convolution can be represented as the multiplication of two sedenion numbers (Bojesomo et al., 2020; Saoud and Al-Marzouqi, 2020). Each component of the sedenion weight vector is re-used 16 times and this leads to a significant reduction in the number of network parameters. This reduction does not come at the expense of representational ability because all input feature maps of the sedenion convolution are used in representing each of the 16 sedenion outputs (Bojesomo et al., 2020). In the proposed sedenion neural network, input data is represented using 16 dimensions, where each dimension represents a single component of the sedenion complex vector.

3.6.2. Model Architecture

The proposed model uses a U-NET based architecture (Bojesomo et al., 2020). The model uses sedenion convolutions except for the *learnVectorBlock* which uses conventional convolution. The *learnVectorBlock* helps in getting the static input (7 channels) into an equal channel dimension as the dynamic input (9 channels).

3.6.3. Training

Our model was trained using two GeForce RTX 2080 Ti GPUs. We used mean squared error (MSE) as the loss function, with the Adam optimizer. The learning rate was initially set to $1e - 4$ and was manually reduced to $1e - 6$, when performance plateaued on the validation set. The proposed model was trained for 15 epochs.

4. Conclusion and Outlook

The *Traffic4cast* competitions aim to bring together scientists from the fields of ML, GISc, and transportation science. In this year's edition at NeurIPS 2020, we set out to explore the following research questions.

1. Sticking with our way of representing the underlying sensor data as aggregated in space and time bins, will providing additional ancillary static and dynamic information per spatial and spatio-temporal bin, respectively, add to the prediction accuracy?

2. Can modern ML-based traffic prediction techniques on large amounts of simply aggregated data in space and time avoid the decrease in prediction accuracy frequently reported for horizons larger than 15min (Ermagun and Levinson, 2018a; Dunne and Ghosh, 2011; Ermagun and Levinson, 2018b; Lana et al., 2018)?

3. Do the results from this year's competition provide further support for the role of the U-net and related encoder-decoder models as promising one-solves-all candidate architectures (Kreil et al., 2019) for geospatial processes?

The first question was heavily explored by our contestants. Choi (2020) and Qi and Kwok (2020) report that their models could only make limited use of the static data provided. Several participants tried to exploit apparent repeating patterns in traffic data. Wu et al. (2020a) report a small improvement in their model performance by adding features derived from traffic states over the previous week, adjusted by a random procedure due to the fact that our test set this year did not allow to derive this information. Moreover, a feature indicating to the model whether the prediction was during a holiday or weekend yielded a small improvement. Xu et al. (2020) report that their method of exploiting seasonal information by training on similar climate periods to those one wants to predict is effective in Moscow, but fails in Berlin and Istanbul, although their final model did not make use of this technique. Wu et al. (2020a) and Qi and Kwok (2020) allow location information to be added and learned, but report little improvement to their current models. There is anecdotal evidence only that the provided incidents information was of little help. Many participants discarded this channel as an input all together (e.g. Santokhi et al. (2020)) but still reached competitive scores. The results of our competition points to a strong affirmative answer to our second question. Similar and improved architectures from last year's competition (e.g. Choi (2020), Wu et al. (2020a)) show little deterioration over the longer time horizon. Nevertheless, as noted by Qi and Kwok (2020), a visual inspection of the actual predictions raises the question as to what part of the dynamics our MSE loss metric encourages solutions to capture. Although U-net architectures featured heavily in this year's competition amongst the highest scoring entries (Choi, 2020; Xu et al., 2020; Qi and Kwok, 2020; Santokhi et al., 2020) other approaches or combination with other approaches showed promise, too. Wu et al. (2020a) used HR-NET, an adaptation of Wang et al. (2020) originally designed for high resolution representation learning. Bojesomo et al. (2020) uses a sedenion based network architecture, achieving competitive results with few trainable parameters. Moreover, Qi and Kwok (2020) reports that merging the results of a U-net architecture with that of a Graph Ensemble Network (GEN) (extending the GNN in Martin et al. (2020)) leads to a performance improvement, indicating that their GEN can captures different dynamic aspects to their U-net approach. Taking into account the success of U-net architectures in precipitation prediction (Agrawal et al., 2019) on similarly spatially and temporally aggregated data, our competition results indicate that U-net architectures capture relevant parts of the dynamics and should thus be candidate architectures when faced with predicting geo-spatial processes aggregated in a similar way. Exploring and quantifying this further on different types of processes and tracking successful architectures could open the way to being able to predict such processes even if the underlying laws are unknown, known but unhelpful for real world scenarios or contain a large random element. Moreover, this aggregated approach also allows to learn pattern similarities between different process observations. Thus, one could feasibly try and predict 2-wheeler traffic from 4-wheeler traffic, or model the influence of weather on traffic as planned for *Traffic4cast* 2021. We will similarly extend this approach to other domains, such as multi-channel weather prediction (`www.traffic4cast.ai` launching shortly).

Acknowledgments

We would like to thank HERE technologies for making our competition data available.

References

Shreya Agrawal, Luke Barrington, Carla Bromberg, John Burge, Cenk Gazen, and Jason Hickey. Machine learning for precipitation nowcasting from radar images. *arXiv preprint arXiv:1912.12132*, 2019.

Luc Anselin. What is special about spatial data? alternative perspectives on spatial data analysis. Technical Report 89-4, National Center for Geographic Information and Analysis, University of California, Santa Barbara, 1989.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. Ser. B*, 57(1):289–300, 1995.

Alabi Bojesomo, Hasan Al Marzouqi, and Panos Liatsis. Traffic flow prediction using deep sedenion networks. 2020.

Dominik Bucher, Francesca Mangili, Francesca Cellina, Claudio Bonesana, David Jonietz, and Martin Raubal. From location tracking to personalized eco-feedback: A framework for geographic information collection, processing and visualization to promote sustainable mobility behaviors. *Travel behaviour and society*, 14:43–56, 2019.

Wei Chen, Boqiang Liu, Suting Peng, Jiawei Sun, and Xu Qiao. S3d-unet: separable 3d u-net for brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 358–368. Springer, 2018.

Sungbin Choi. Traffic map prediction using unet based deep convolutional neural network. *arXiv preprint arXiv:1912.05288*, 2019.

Sungbin Choi. Utilizing unet for the future traffic map prediction task traffic4cast challenge 2020, 2020.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv preprint arXiv:1606.09375*, 2016.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

Stephen Dunne and Bidisha Ghosh. Regime-based short-term multivariate traffic condition forecasting algorithm. *Journal of Transportation Engineering*, 138(4):455–466, 2011.

Alireza Ermagun and David Levinson. Spatio-temporal short-term traffic forecasting using the network weight matrix and systematic detrending. In *Compendium of papers of Transportation Research Board 97th Annual Meeting*, 2018a.

Alireza Ermagun and David Levinson. Spatiotemporal traffic forecasting: review and proposed directions. *Transport Reviews*, 38(6):786–814, 2018b.

Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701, 1937.

Salvador Garcia and Francisco Herrera. An extension on"statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons. *Journal of machine learning research*, 9(Dec):2677–2694, 2008.

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.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216*, 2017.

Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

Pedro Herruzo and Josep L. Larriba-Pey. Recurrent autoencoder with skip connections and exogenous variables for traffic forecasting. In Hugo Jair Escalante and Raia Hadsel, editors, *Proceedings of the NeurIPS 2019 Competition Track*, volume 123 of *Proceedings of Machine Learning Research*, pages 0–0. PMLR, 2020.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

David Jonietz, Dominik Bucher, Henry Martin, and Martin Raubal. Identifying and interpreting clusters of persons with similar mobility behaviour change processes. In *The Annual International Conference on Geographic Information Science*, pages 291–307. Springer, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

David P. Kreil, Michael K. Kopp, David Jonietz, Moritz Neun, Aleksandra Gruca, Pedro Herruzo, Henry Martin, Ali Soleymani, and Sepp Hochreiter. The surprising efficiency of framing geo-spatial time series forecasting as a video prediction task - insights from the IARAI traffic4cast competition at neurips 2019. In Hugo Jair Escalante and Raia Hadsell, editors, *NeurIPS 2019 Competition and Demonstration Track, 8-14 December 2019, Vancouver, Canada. Revised selected papers*, volume 123 of *Proceedings of Machine Learning Research*, pages 232–241. PMLR, 2019. URL http://proceedings.mlr.press/v123/kreil20a.html.

Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1811–1820, 2019.

Ibai Lana, Javier Del Ser, Manuel Velez, and Eleni I. Vlahogianni. Road traffic forecasting: Recent advances and new challenges. *IEEE Intelligent Transportation Systems Magazine*, 10(2):93–109, 2018. doi: 10.1109/mits.2018.2806634.

Roy St. Laurent and Philip Turk. The effects of misconceptions on the properties of friedman's test. *Commun. Stat. Simul. Comput.*, 42(7):1596–1615, 2013.

Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes. *IEEE transactions on medical imaging*, 37(12):2663–2674, 2018.

Z. Liu, Y. Liu, C. Lyu, and J. Ye. Building personalized transportation model for online taxi-hailing demand prediction. *IEEE Transactions on Cybernetics*, pages 1–9, 2020. doi: 10.1109/TCYB.2020.3000929.

Henry Martin, Ye Hong, Dominik Bucher, Christian Rupprecht, and René Buffat. *Traffic4cast*-traffic map movie forecasting–Team MIE-Lab. *arXiv preprint arXiv:1910.13824*, 2019.

Henry Martin, Ye Hong, Dominik Bucher, Christian Rupprecht, and René Buffat. Graph-ResNets for short-term traffic forecasts in (almost) unknown cities. In Hugo Jair Escalante and Raia Hadsel, editors, *Proceedings of the NeurIPS 2019 Competition Track*, volume 123 of *Proceedings of Machine Learning Research*, pages 0–0. PMLR, 2020.

Sergiu Oprea, Pablo Martinez-Gonzalez, Alberto Garcia-Garcia, John Alejandro Castro-Vargas, Sergio Orts-Escolano, Jose Garcia-Rodriguez, and Antonis Argyros. A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

T. Parcollet, M. Morchid, and G. Linarès. Quaternion convolutional neural networks for heterogeneous image processing. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8514–8518, 2019. doi: 10.1109/ICASSP.2019.8682495.

Dulce G. Pereira, Anabela Afonso, and Fátima Melo Medeiros. Overview of friedman's test and post-hoc analysis. *Commun. Stat. Simul. Comput.*, 44(10):2636–2653, 2015.

Saqib Qamar, Hai Jin, Ran Zheng, Parvez Ahmad, and Mohd Usama. A variant form of 3d-unet for infant brain segmentation. *Future Generation Computer Systems*, 108:613–623, 2020.

Qi Qi and Pak Hay Kwok. Traffic4cast 2020–graph ensemble net and the importance of feature and loss function design for traffic prediction. *arXiv preprint arXiv:2012.02115*, 2020.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Jay Santokhi, Pankaj Daga, Joned Sarwar, Anna Jordan, and Emil Hewage. Temporal autoencoder with u-net style skip-connections for frame prediction. 2020.

L. S. Saoud and H. Al-Marzouqi. Metacognitive sedenion-valued neural network and its learning algorithm. *IEEE Access*, 8:144823–144838, 2020. doi: 10.1109/ACCESS.2020.3014690.

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. *Advances in Neural Information Processing Systems*, 2015-January:802–810, 2015. ISSN 10495258.

Leslie N. Smith. Cyclical learning rates for training neural networks. *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, (April): 464–472, 2017. doi: 10.1109/WACV.2017.58.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015.

Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016.

J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. doi: 10.1109/TPAMI.2020.2983686.

Yu Weng, Tianbao Zhou, Yujie Li, and Xiaoyu Qiu. Nas-unet: Neural architecture search for medical image segmentation. *IEEE Access*, 7:44247–44257, 2019.

Fanyou Wu, Yang Liu, Zhiyuan Liu, Xiaobo Qu, Rado Gazo, and Eva Haviarova. Tlab: Traffic map movie forecasting based on hr-net. *arXiv e-prints*, pages arXiv–2011, 2020a.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

Jiasong Wu, Ling Xu, Fuzhi Wu, Youyong Kong, Lotfi Senhadji, and Huazhong Shu. Deep octonion networks. *Neurocomputing*, 397:179 – 191, 2020b. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.02.053.

Jingwei Xu, Jianjin Zhang, Zhiyu Yao, and Yunbo Wang. Towards good practices of u-net for traffic forecasting, 2020.

Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2016.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2020.

Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Crevnet: Conditionally reversible video prediction. *arXiv preprint arXiv:1910.11577*, 2019.

Yongjin Zhou, Weijian Huang, Pei Dong, Yong Xia, and Shanshan Wang. D-unet: a dimension-fusion u shape network for chronic stroke lesion segmentation. *IEEE/ACM transactions on computational biology and bioinformatics*, 2019.