Vision Transformer and Inpainting based Approach for Short-term Forecasting of Passenger Loads on a Transit Metro Line. Focus on Explainability and Atypical Situation prediction

Thomas Bapaume, Etienne Côme, Mostafa Ameli, Jérémy Roos, Latifa Oukhellou

VISION TRANSFORMER AND INPAINTING BASED APPROACH FOR SHORT-TERM FORECASTING OF PASSENGER LOADS ON A TRANSIT METRO LINE. FOCUS ON EXPLAINABILITY AND ATYPICAL SITUATION PREDICTION

Thomas Bapaume (corresponding author)
University Gustave Eiffel, COSYS, GRETTIA, Paris, France
Régie Autonome des Transports Parisiens (RATP), Paris, France
https://orcid.org/0000-0001-7189-7490
thomas.bapaume@univ-eiffel.fr

Etienne Côme
University Gustave Eiffel, COSYS, GRETTIA, Paris, France
https://orcid.org/0000-0002-0459-6388
etienne.come@univ-eiffel.fr

Mostafa Ameli
University Gustave Eiffel, COSYS, GRETTIA, Paris, France
https://orcid.org/0000-0002-2470-6812
mostafa.ameli@univ-eiffel.fr

Jérémy Roos
Régie Autonome des Trasports Parisien (RATP), Paris, France
https://orcid.org/0000-0003-2848-8514
jeremy.roos@ratp.fr

Latifa Oukhellou
University Gustave Eiffel, , Paris, France
https://orcid.org/0000-0002-5193-1732
latifa.oukhellou@univ-eiffel.fr

Paper submitted for presentation at the 102nd Annual Meeting Transportation Research Board, Washington D.C., January 2023 to the Transportation Research Board AED50 Standing Committee on "Artificial Intelligence and Advanced Computing Applications".

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

AUTHOR CONTRIBUTION STATEMENT

All the authors have contributed to all aspects of this study, ranging from the conception and design of the methodology, analysis and interpretation of the results and discussion, and the manuscript preparation.

Word Count: 6279 words + 5 table(s) × 250 = 7529 words

Submission Date: November 23, 2023

ABSTRACT

This paper presents a deep learning approach for the real-time prediction of train passenger loads. We generate an image to represent the metro line data, wherein a pixel's coordinates determine a train departure, and its color denotes the train loads. A computer vision technique called inpainting is deployed to complete a real-time metro traffic image, which is equivalent to the prediction task. First, we introduce the transformation of the metro line data to an image, which takes into account all metro line constraints, e.g., irregular time sampling of trains. Second, we present the methodology to forecast the train loads based on two main deep learning approaches: U-Transformer and Channel Vision Transformer. These models offer a multi-step forecasting process over the metro line by extracting the visual features of the images. Third, we apply the proposed models to a real test case of the Paris metro line 9 to validate and benchmark our models against various classical and deep learning-based forecasting approaches. The results show that the proposed models outperform the existing models in forecasting. Fourth,we perform an In-depth analysis based on attention scores and latent spaces to interpret the performances of the proposed methods. Furthermore, we investigate the results of our models in seven atypical scenarios (e.g., strike, lockdown, and disruptions) to evaluate the robustness of the proposed approaches.

Keywords: Public Transport; Forecasting; passenger loads; Computer Vision; Inpainting; Deep Learning; Transformer.

INTRODUCTION

Forecasting the evolution of demand and supply are crucial for transport operators and travelers. In intelligent public transport systems, real-time forecasting information about passenger loads can be used by operators to anticipate and optimize their service level and by travelers to obtain robust information for planning their journey. In this context, short-term forecasting models can be used for the real-time prediction of passenger loads. In addition, a reliable short-term prediction model should be able to anticipate passenger loads during rare events such as strikes or disruptions. This study focuses on forecasting train loads for both normal and atypical situations in urban transit systems.

There are multiple approaches to represent and solve short-term prediction problems involving non-linear inference and recursive data with both spatial and temporal dependencies. Recently, deep learning and machine learning approaches have been widely used to deal with forecasting problems in transportation applications. Compared to classical time series methods, they have shown great accuracy in modeling sequential multivariate data (*1*) and offer better scalability. The main techniques deployed in the literature are based on the Recurrent Neural Network (RNN) or Long Short Term Memory (LSTM) framework (*2*). These approaches are capable of managing the short-term and long-term evolution of the target time series to be predicted. Several studies have proposed LSTM-based models to predict various target variables in urban transportation systems. (*3*) used LSTM to predict the bus demand of Melbourne city. (*4*) proposed an LSTM-based model for smart cards to forecast dynamic Origin-Destination matrices of a subway line in Rennes. (*5*) proposed a two-dimensional LSTM network for the same task to take into account the Spatio-temporal correlations between origins and destinations in a transit network. In addition to the above-mentioned works on the demand side, several studies focused on the prediction of variables corresponding to the supply side. For example, (*6*) proposed two LSTM-based models to achieve a multi-step prediction for the number of available bikes at the city scale. Moreover, many studies proposed a combination of two deep learning architectures, using LSTM to handle the temporal dependencies and another approach to deal with spatial dependencies. For example, (*7*) developed a deep Convolutional LSTM (ConvLSTM) to extract spatio-temporal information to forecast travel demand for Chengdu city in China. In recent years, graph methodologies have emerged to represent public transport networks and encode spatio-temporal features. In this regard, (*8*) used a three-level graph representation of the London metropolis to forecast bicycle usage. (*9*) relied on Probabilistic Graph Convolution to extract spatio-temporal features in order to forecast Sydney public transit demand. (*10*) merged recurrent neural network and graph convolution to learn spatio-temporal information to predict bus and taxi flow distributions in Beijing. Other classical machine learning strategies to forecast the time series of a public transport network can be also mentioned, such as Gradient boosting (*11, 12*), dynamic Bayesian network (*13*) or (*14*) which transformed the public transport load forecasting into a supervised classification task. However, LSTM-based models require time-sequential data and they focus mainly on temporal dependencies. There are various strategies to deal with spatial dependencies, such as graph-based or image-based approaches. In these approaches, the spatio-temporal series of a public transport network are represented as images, including the history of all train movements. This makes it possible to define prediction tasks more precisely over all trains moving inside a public transport system. In the present study, we used an image-based approach to consider spatio-temporal dependencies and define the prediction tasks.

All the above-mentioned studies consider the regular time sampling feature of urban transit

data. This means that the forecasting model is built according to a constant time interval in order to sample data without considering the flow dynamics of the mode of transport, e.g., trains and buses. However, in the real case, the dynamic evolution is basically irregular. There are few studies in the literature that have taken this point into account (*15, 16*).

In order to take into account the specificities of real train data, such as irregular time sampling of train stops and a univariate space defined by a metro line, the present study applied an image representation of the passenger loads. In other words, the idea is investigating forecasting over images through an inpainting task. Thus, with the image representation approach, the prediction task is transformed into an image completion task, wherein the pixels to be completed are related to the future data that we want to predict.

Inpainting is a well-known computer vision technique that aims to complete the missing area of an image. Adapting inpainting to a forecasting task is a recent approach used by the authors in (*17*) to extract features and forecast the level of traffic on a highway road. It was also used in our previous work (*16*) to build a framework that is able to forecast passenger loads over an entire metro line. In addition to using 2D images, (*18*) used inpainting to forecast an electric energy system temporal series by transforming them into a 3D image.

In (*16*), we designed an image inpainting architecture to implement deep learning methods for short-term prediction problems. We used the classical U-net method as the deep learning function. However, there are more advanced methods, such as attention-based models, to improve the prediction performance. The efficient performance of attention-based models, e.g., Transformer, is proven in computer vision where they have become the state of art (*19*). Transformer architecture, introduced in 2017 by (*20*), outperforms other solutions in various domains such as natural language processing with BERT architectures (*21*), or computer vision with Vision Transformer (ViT) (*22*). The present study aims to extend a forecasting inpainting architecture with self-attention mechanisms and Transformer to map the relationships between train departures and the dynamics of a metro line. We use self-attention mechanisms that show a high potential to be deployed for short-term prediction (*23–25*).

Moreover, most of the studies in the literature do not apply their methodology to atypical scenarios. Prediction is particularly difficult for theses cases since they are under-represented in the dataset. Therefore it is not straightforward to determine which methodology is robust for short-term prediction. Besides, among all the aforementioned methods, only two papers (*8, 13*) emphasize the explainability of the proposed forecasting methodology. Here, we have an assiduous focus on atypical situations to interpret the results of the proposed models.

To help position our contribution statement, we present in Table 1 a characterization of the existing prediction frameworks in the literature. A forecasting framework is often constructed according to the prediction objectives and the data available for its implementation. Thus, we have defined several categories to classify the various transit forecasting models as shown in Table 1: operation status, spatial information, time sampling, methodology, and explainability. Operation status denotes a study on normal or rare and complex forecasting situations. The spatial information defines the spatial scope of the forecasting ranging from local to global information (e.g., station, bus, train line, transit network). Time sampling informs about the aggregation of the data. Regular sampling means that data are aggregated at a specific time interval (e.g., every 10 minutes), while irregular sampling means that the data are aggregated according to a real event or change inside the network (e.g., train departure, taxi stops). The methodology column refers to the models that are deployed for the forecasting. Finally, the explainability refers to whether the

TABLE 1: The sample of different studies on short-term prediction for transport networks in the literature.

Research	Operation Status		Spatial information			Time sampling		Methodology					Explainability
	Normal	Atypical	Single station	Transit Line	Full network	Regular	Irregular	Classic methods	Deep learning methods				
									LSTM	Graph	Transf.	U-net	
Toqué et al. (*4*)	x		x			x			x				
Zhao et al. (*5*)	x				x	x			x				
Liu et al. (*6*)	x		x			x			x				
Liyanage et al. (*3*)	x			x		x			x				
Wang et al. (*7*)	x				x	x			x	x			
Colace et al. (*8*)	x		x			x		x					x
Li et al. (*9*)	x				x	x				x			
Peng et al. (*10*)	x				x	x			x	x			
Pasini et al. (*15*)	x		x				x		x				
Du et al. (*26*)	x	x			x	x			x	x			
Wu et al. (*11*)	x					x		x					
Egu and Bonnel (*12*)	x		x			x		x					
Roos et al. (*13*)	x		x			x		x					x
Heydenrijk-Ottens et al. (*14*)	x			x			x	x					x
Hao et al. (*25*)	x					x					x		
Wu et al. (*24*)	x			x		x			x	x	x		
Chen et al. (*23*)	x	x				x					x		
Bapaume et al. (*16*)	x			x			x					x	
This work	x	x		x			x				x	x	x

study interprets the models' results or not. Therefore, the contributions of this paper are detailed as follows:

- We developed two attention-based architectures, the U-transformer and the Channel Vision Transformer. To this end, we performed a benchmark on real load data collected during 3 years of operation of the Paris metro line 9 to compare these two architectures' ability to forecast loads of all the next 4 train departures of a metro line. A comparison of forecasting performances was carried out, including several deep learning and basic machine learning approaches.

- Secondly, we exploited the latent space of the prediction models to interpret the forecasting. We used attention scores to explain the results and to highlight the relationship between the prediction of one train departure (one pixel) and all train departures in an image.

- Finally, by using the image representation, we create various test sets representing atypical situations observed on a metro line, such as strikes or disruptions. The goal is to evaluate the robustness and generalization capacity of the proposed models by computing the forecasting performances in seven atypical contexts.

The remainder of this paper is organized as follows. In section 2, we introduce an image representation of a metro line. Next, we define the forecasting frameworks and transformer-based models in section 3. Section 4 presents various test sets to evaluate the prediction robustness of the proposed models in particular contexts. In addition, we present the results over five months of public transport images in section 4. Finally, in section 5, we outline the main conclusions of this paper and mention some future research directions.

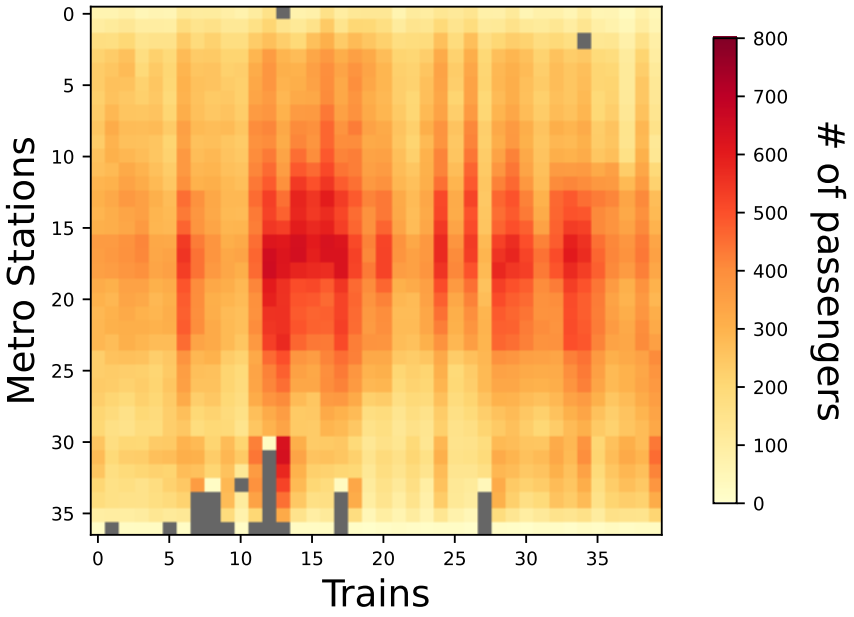

FIGURE 1: Image of the Paris metro line 9 for 40 trains and 36 stations. Grey pixels represent missing values

PROBLEM AND STATEMENT

This section presents the transformation of the short-term prediction problem into an image completion task by representing the train loads of a metro line as an image.

Metro traffic images representation

In (*16*), we defined an image representation of trains moving along a metro line to apply a forecasting framework. Figure 1 depicts an image denoted I that represents the train loads of all trains over a metro line in real-time. Each pixel encodes the train load corresponding to a single train identified by the column number at the station denoted by the row number. Therefore, each column represents the course of a train along the entire metro line, and each row represents the sequence of departures at each metro station. The color encodes the number of passengers for a train departure. The grey pixels in Figure 1 represent the lack of data due to missing data or no train stops.

Real Time Images and mask

Figure 1 represents an image at the end of the day, when all the trains have finished running. But in a real time perspective, the trains are moving along the metro line. Thus, at time t, many pixels are in fact missing in the image, depicted by the grey pixels in Figure 2(A). We denote the real time image as $I(t)$ at time t.

The image $I(t)$ is composed of two parts: a past area with the realized traffic at time t and a future area representing the forthcoming departure of trains to be forecasted. In addition, the size of the missing area can change depending on the number of trains in specific contexts (e.g., peak hours, off-peak hours). Forecasting the passenger loads of the next train departures consists in completing the missing area introduced by construction inside the image $I(t)$. This method is called inpainting in computer vision. The forecasting horizon is based on the number of pixels to

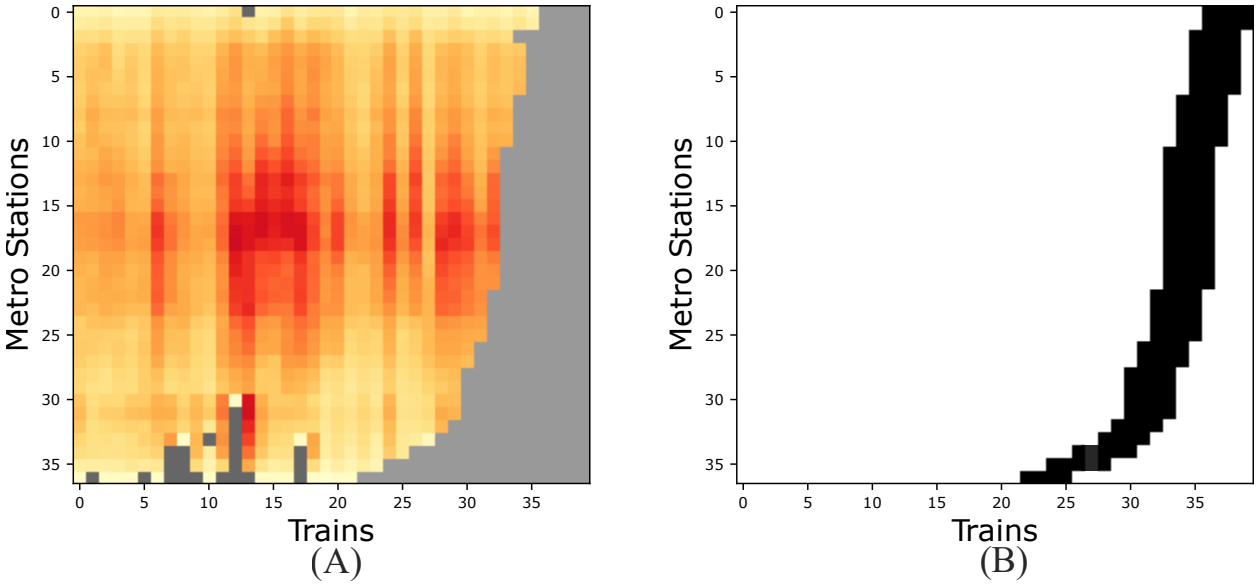

FIGURE 2: (A) Real time Image $I(t)$ of the Paris metro line 9, including 37 stations and 40 trains at time t; grey pixels represent future train stops. (B) Mask m_y of the pixels to be forecasted (cluster pixels) within the image.

predict for each station. A short-term forecast will focus on a few pixels per station, whereas a long-term forecast will focus on the whole picture. A mask m_y is defined to encode valid pixels, which must be predicted. Figure 2(B) depicts an example of mask to forecast the 4 next departures.

Thus, the image forecasting framework is divided into two serial steps, an inpainting task and a forecasting task, as shown in Figure 3. We denote the inpainted image and forecast image by $\hat{I}(t)$ and $\hat{I}_{for}(t)$, respectively. In addition, the input image $I(t)$ is composed of several sub-images called channels which encode several metro line variables.

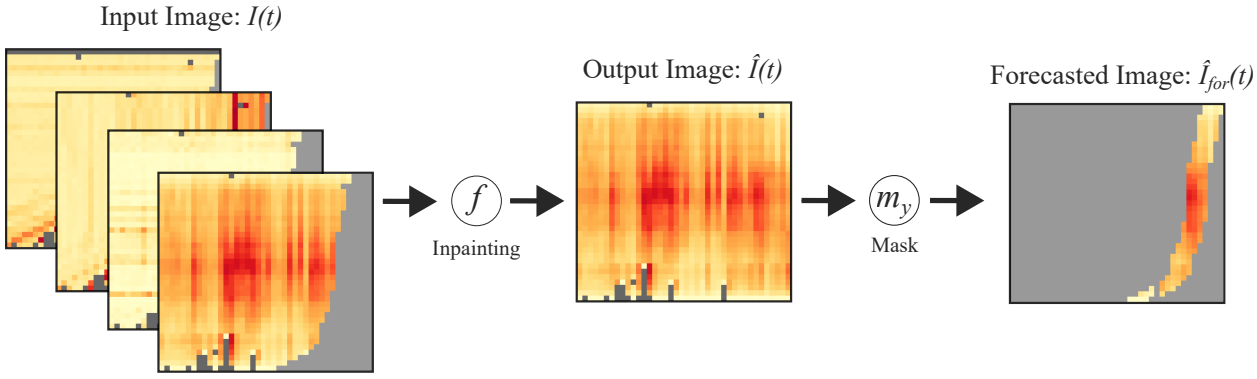

FIGURE 3: A two steps of forecasting framework: inpainting and forecasting. The inpainting image is the result of an inpainting function f. The forecasted image represents only the pixel considered by the prediction.

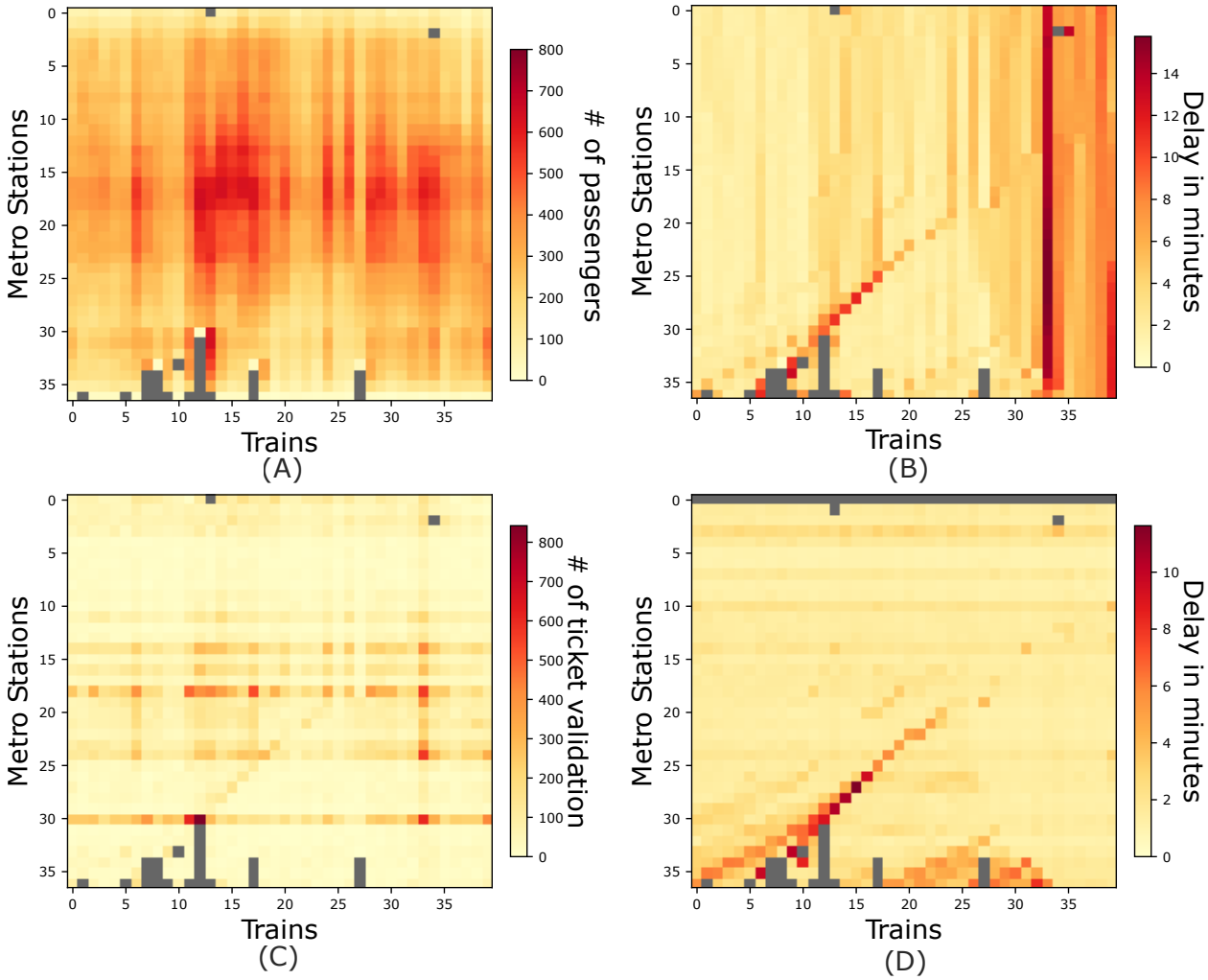

FIGURE 4: Channels used to generate a metro traffic image. (A) train passenger loads channel, (B) Waiting time channel, (C) Tape in ticketing channel, (D) Travel time channel

Dimension and Channel Encoding

In computer vision, we can define an image over 3 dimensions: the height h, the width w, and the channels d. A channel is a sub-image that encodes one variable. In standard RGB images (Red, Green, Blue), the image has 3 channels for each color. In this case, the height is the number of stations of the studied metro line, and the width represents the number of trains considered to generate the image, which is linked to the maximum capacity of the public transport line and the forecasting task. Here, the height and the width are set to respectively 37 (stations) and 40 (trains). The metro traffic image channels represent data of the train network. For this study, 4 variables were defined as shown in Figure 4: (A) train load, (B) waiting time, (C) remote ticketing validation (tap-in) at the station, and (D) travel time. These four channels are used as inputs of the inpainting framework where the goal is to complete only the load channel.

PROPOSED METHODOLOGICAL FRAMEWORK

This section introduces the inpainting framework to forecast the next trains passenger loads. We present two architectures that fit the inpainting function based on the attention mechanism.

Inpainting framework

In computer vision, image inpainting is a well-known approach used to complete images (*27, 28*). The goal is to extract visual features and contexts from the picture to paint all missing pixels, train departures in this study. The developed framework is divided in two steps: (i) the reconstruction of a missing part of an image $I(t)$ denoted by $\hat{I}(t)$ and (ii) the extraction of the forecasted loads from $\hat{I}(t)$ illustrated in Figure 3. In the first step, we applied an inpainting function f to an input image. Then, to extract future train loads denoted by $\hat{I}_{for}(t)$, we used a mask m_y in the second step to select the corresponding pixels from $\hat{I}(t)$. The following equation summarizes the forecasting task:

$$\hat{I}_{for}(t) = \hat{I}(t) \odot m_y, \text{ where } \hat{I}(t) = f(I(t)) \text{ and } \odot \text{ is the product of Hadamard} \quad (1)$$

Thus, the model must minimize the following lost function:

$$L(I(t), \hat{I}(t), I_{for}(t), \hat{I}_{Pred}(t)) = \alpha MSE(I(t), \hat{I}(t)) + \beta MSE(I_{for}(t), \hat{I}_{for}(t)) \quad (2)$$

The loss function in equation (2) is a pixel-wise loss function divided in two components, an inpainting term and a forecasting term. For more information about this framework, please refer to (*16*) This work aims to use the transformer model as our inpainting function f in the forecasting framework. In addition, this framework scales to all public transport lines by changing the size of the input image. Moreover, the proposed framework can be easily adapted to other forecasting targets such as delay or time travel. We can also potentially simultaneously predict delays and train loads, in which case the output would be a two channel image. The only requirement is related to vehicle granularity (e.g. train, bus) in order to build images.

Forecasting Transformer Models

Transformer and Vision Transformer

In recent years, deep learning has seen a major evolution with the new network architecture called the Transformer (*20*) which is fully based on attention mechanisms. This model tends to simulate the human behavior of attention. From a computer science view, the attention mechanisms are used to enhance the dependencies between elements of a sequence or an image. In addition, they can manage temporal dependencies without any recursive architecture. The Transformer was originally used in natural language processing (NLP) to map the dependencies between words in a sentence or text. In computer vision, the Vision Transformer (ViT)(*22*) model was proposed to adapt the attention mechanisms to the image classification task by constructing a visual word sentence called patches (e.g. a visual word is a small neighboring pixel group). Transformer model is now seen as the state of art in many tasks of computer vision such as classification (*29–31*), segmentation (*32, 33*), object detection (*34, 35*) or image generation (*36*). Recently, transformers are applied to short-term forecasting tasks. For example, (*23*) built a bi-directional encoder–decoder transformer model to estimate future and past traffic conditions to forecast transport flows. (*24*) forecasted time travel by integrating attention mechanisms to a ConvLSTM or (*25*) proposed to forecast metro passenger flows by merging a sequence to sequence architecture with an attention mechanism.

The Transformer is a deep learning architecture composed of a multi-head self attention module which uses attention to build various dependency representations of all pixels regardless

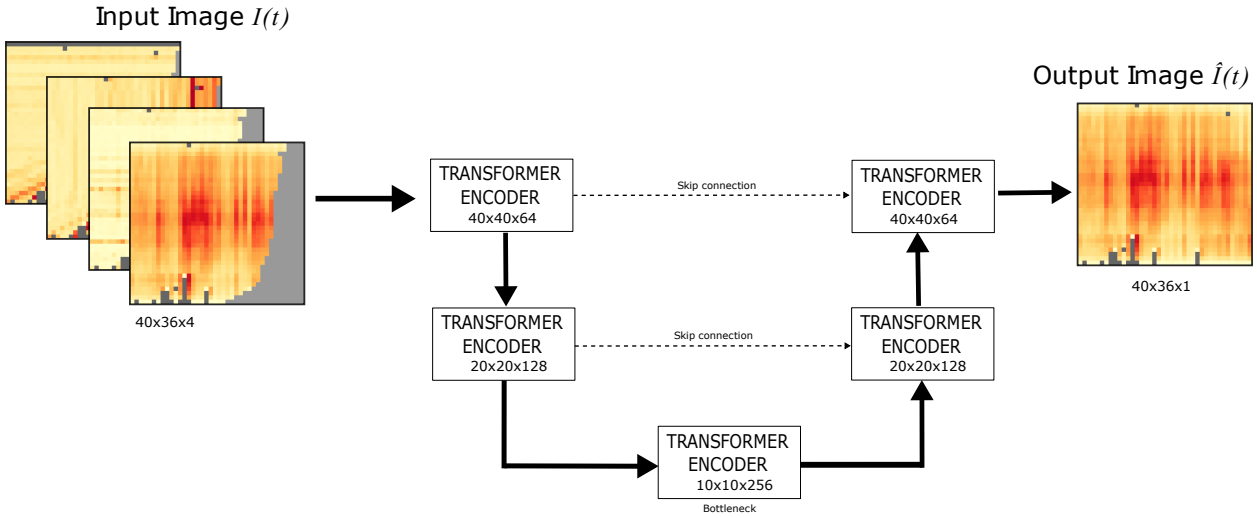

FIGURE 5: Architecture of the U-Transformer

their coordinates. It relies on a attention function given by the equation 3:

$$Attention(Q,K,V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3}$$

where $Q \in \mathbb{R}^{n \times d_k}$ is the query vector, $K \in \mathbb{R}^{n \times d_k}$ is the key vector, and $V \in \mathbb{R}^{n \times d_k}$ is the value vector with n the number of pixels or patches, and d_k the dimension of the key vector K.

In our computer vision task, Q, K and V vectors are computed from the pixels of the image $I(t)$ through projection matrices denoted W^Q, W^K, W^V. The goal of attention is to compute the similarity between the query vector Q and the key vector K. Then we pass the pixel information through the Value vector V. In order to maintain the coordinate information of a pixel, Transformer uses a positional encoding to propagate this information through the architecture. Noted that a transformer uses several projection matrices in order to map multiple representations of the same input image $I(t)$ defined as $Head$. In computer vision, in order to reduce the size of the input vectors, a non-overlapping patch decomposition (*22*) is used to create an ordered sequence of small picture from an image called patch (e.g. patch of 16×16 or 32×32 pixels). For more details about transformer and attention mechanisms, the reader can refer to (*20, 22*).

U-Transformer

This work proposes two deep learning architectures using a transformer as a primary component. The goal is to integrate the relationship representation of attention to the inpainting function f.

The first model noted as U-Transformer is an evolution of the model U-net. The idea is to replace the convolution and pooling layers with a transformer, as shown in Figure 5 for both the encoder and decoder part of the U-shape model. The architecture is based on the Swin-Unet proposed by (*33*) for image segmentation. But other strategies merge transformer blocks and U-net models. For example, (*37*) aims to apply attention to some parts of the models like feature space or (*38*) only on the decoder. The U-Transformer is divided into two steps. Besides, the pooling task is achieved by concatenating patches together.

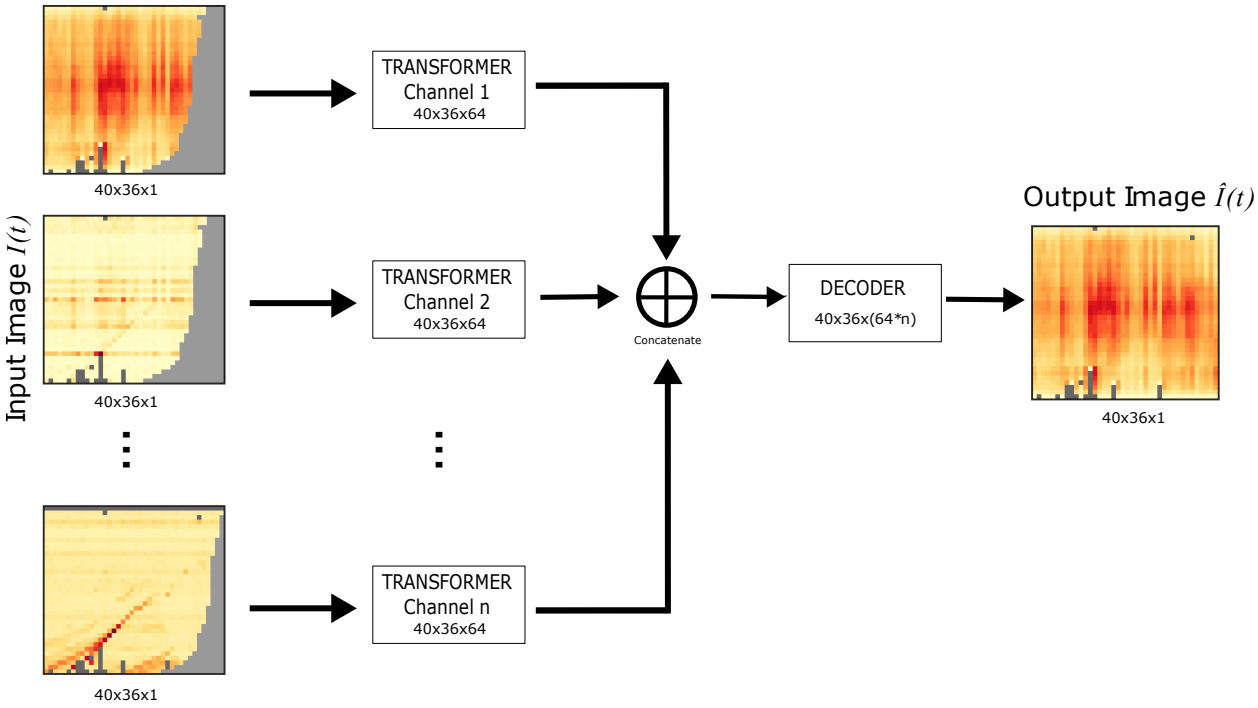

FIGURE 6: Architecture of the Channel Vision Transformer

Channel Vision Transformer

A metro traffic image is composed of channels related to each problem variable. But some of these variables are linked to a train (passenger load or travel time), and others are linked to a station (the ticketing data or waiting time). The goal is to build an architecture that considers this characteristic and computes an attention score independently for each variable related to a train or a station. The Channel Vision Transformer is built with this objective, and it is similar to Channel-wise methodologies (*39*, *40*). This new architecture applies transformer encoders for each channel composing $I(t)$ (Figure 1(b)). The proposed model is composed of 4 inputs, one per channel, to which we apply a transformer encoder to extract features from each variable as shown in Figure 6. The patch size used in this model is 1×1 meaning that the patch is equivalent to a train departure—the image's reconstruction results from one convolution performed on the concatenation of each transform encoder.

Attention based model limitation and learning

Transformer models may have relevant performances in many domains. However, training them can be costly in terms of computing resources, mainly in terms of computer memory (graphic processing unit - GPU) required for calculating attention scores. Thus, a patch encoding of 1 considerably increases memory usage and impacts the architecture and training. The training is performed on an NVIDIA RTX 3090 graphic card with 24 GB of RAM and the TensorFlow python framework and python 3.8. To train the transformer models, we limited the number of heads and the projection size of the multi-head attention layer to 16 and 64. In addition, the batch size is also impacted by the transformers' resource cost. We fixed it to 8 or 4 depending on the model. Compared to the convolution model, the learning of the transformers is constrained by the GPU memory at our disposal.

1 deepl

FORECASTING RESULTS, AND MODEL EXPLAINABILITY

Model configuration and numerical experiments

In order to evaluate the performance of our methodology, we applied it to a real data set collected from Line 9 on the Paris metro, which includes train loads, ticket validations (tap-in), train waiting times, and travel times. The size of the generated images was $36 \times 40 \times 4$: 36 stations (i.e., we omitted the last station), 40 trains, and 4 channels (i.e., input variables). These images were generated from the data collected every minute during the daily operating times of the metro line over a long period from January 2019 to April 2021.

To benchmark our two proposed models, we compared them to other forecasting models. The first model was a Naïve approach in which the predicted values are equal to the last passenger load at each station. We also considered several deep learning approaches, including two classical methods - a Neural Network (NN) and a fully Convolutional Neural Network (CNN), and four recent methods - a Convolutional Neural Network combined with a Neural Network (CNN+NN) (*17*), a U-net Model (*16*), a one Transformer Model (*20*), and a Vision Transformer (VIT) model (VIT)(*22*).

Table 2 summarizes the parameters of all nine considered models. Readers should note that this study used the Soft Plus output activation function to match a positive passenger load. For the learning step, all the models were trained with the Adam optimizer and a learning rate of 0.0001. To perform benchmarking, we split the data set into learning and test sets. The first data set was composed of approximately 700,000 images from January 2019 to April 2021 with 1,200 images per day, i.e., one image per minute between 5:30 am to 1:30 am the next day. The test set had to cover the majority of loading situations, such as strikes, nominal traffic, holidays, etc., and could not be chosen randomly. This constraint was due to the image format, where overlapping could occur between neighboring images during training and bias the outputs. Thus, over three years, the test set consisted of 5 full months of images (2 for 2019, 2 for 2020 and 1 for 2021) and a total of around 220,000 images.

It denotes the number of parameters to be learned, and the activation function for the output layer for deep learning approaches. The number of the head by encoder and the patch size for transformer-based models are also presented. The output activation function used in this study is the $Softplus$ to match with a positive passenger load.

TABLE 2: The settings of all the benchmarked models.

Models (reference)	#parameters	Batch size	Head	patch size
Naïve	-	-	-	-
NN	27,354,528	128	-	-
CNN	20,254,465	128	-	-
CNN + NN (*17*)	25,418,272	128	-	-
U-net (*16*)	16,552,065	128	-	-
Transformer (*20*)	78,081	8	16	1
VIT (*22*)	6,465,345	8	64	3
U-Transformer (ours)	29,839,681	4	64	3
Channel Vision Transformer **(ours)**	21,902,849	4	16	1

Forecasting and Inpainting results

We trained the forecasting models and applied them to the test set. Table 2 reports the results of both the inpainting ($\hat{I}(t)$) and forecasting ($\hat{I}_{for}(t)$) tasks for the 4 next departures from each station (see Figure 3). We used two standard evaluation metrics, Root Mean Square Error (RMSE) and Weighted Mean Average Error (WMAPE) (*13*). For inpainting, the U-net model outperformed the other models with a WMAPE of 6.6% and an RMSE of 29%. Regarding the passenger load prediction task, the U-transformer provided the best results with a WMAPE of 11.2% and an RSME of 31%. It is worth noting that the transformer model tended to overfit in inpainting (a WMAPE of 8.3%) and performed poorly in forecasting (a WMPE of 18.2%). However, these results were estimated from the whole test set in which atypical situations had not been identified. Before analyzing the performance of methods in atypical cases, we will discuss the explainability of the methods in order to understand how they accomplished the inpainting and forecasting tasks.

TABLE 3: Inpainting and forecasting results based on weighted mean absolute percentage error (WMAPE) and Root Mean Square Error (RMSE).

	Inpainting		Forecasting	
Models (reference)	RMSE	WMAPE	RMSE	WMAPE
Naïve	63	14.2	50	19.9
NN	60	22.4	48	18.9
CNN	41	12.3	33	13.1
CNN + NN (*17*)	44	15.9	36	14.8
U-net (*16*)	**29**	**6.6**	31	12.3
Transformer (*20*)	30	8.3	43	18.2
VIT (*22*)	42	12.9	33	12.4
U-Transformer (ours)	34	9.9	**31**	**11.2**
Channel Vision Transformer **(ours)**	32	9.2	32	12.1

Forecasting explainability

We applied two approaches in order to interpret the results: Attention Score and Latent Space Exploration. In this subsection, we shall present and discuss the results of both analysis methods.

Attention Score Analysis

Transformer-based models are able to provide attention scores that can improve the explainability of our prediction models. The attention scores allow us to interpret the prediction models and visualize the relationships between the pixels that generate the forecast. This subsection will focus on the explainability of the Channel Vision Transformer model based on the attention scores generated by the equation 3 for the different input variables. Note that this model was chosen instead of the U-transformer model because (i) it gives similar results to the U-transformer and has a patch encoding of size 1; (ii) it is a channel-wise approach that allows us to observe attention for each channel of the image. We can therefore compute attention scores for each individual pixel in $\hat{I}_{for}(t)$ and interpret them. For instance, Figure 7 presents the computed attention scores for one pixel of $\hat{I}_{for}(t)$ in relation to the other pixels. Figure 7 (a) shows the location of the targeted pixel (green pixel) in $\hat{I}_{for}(t)$.

We limited our analysis to 4 heads per channel, but we have developed a visualization tool to explore the entire image and all the attention scores. Figure 7 (b) depicts the attention scores

measured between the green pixel and all the pixels for 4 heads from the Transformer encoder of the load channel. Figure 7 (c) and (d) show the same representation for the travel time and waiting time channels. The attention scores from the load channel in Figure 7 (b) show that the transformer model focused on the most recent train departures (e.g., Head0) or on the previous train loads (e.g., Head4) to perform prediction. For the travel times shown in Figure 7 (c), we can see that the focus was more on future train departures rather than past journeys. For the waiting times (see Figure 7 (d)), heads 1 and 5 show that attention was focused on the majority of past train departures. In addition, some heads (e.g., Head1 or Head5 of Figure 7 (b)) did not show any significant attention scores meaning that there was no relationship between the pixel and the image that activated the head in question. With attention scores, we can observe that both proposed methods took account of the important information from the other pixels to perform forecasting. However, this function is not the only way to interpret the results, and we also investigated other representation learning methods such as Latent Space.

Latent Space exploration

To explain the forecasting model's choices and results, it is possible to explore the feature space the model builds to represent all the training images. The U-Transformer architecture is similar to an auto-encoder model wherein the feature space, called the latent space, is represented by a bottleneck in the U-shaped architecture. We used t-distributed stochastic neighbor embedding (t-SNE) projection (*41*) to reduce the high dimensional latent space into 2D in order to obtain an understandable visible representation. The t-SNE was only used for visualization and it has the advantage of keeping the same distance between the 2D and the original high-dimensional space. Figure 8) presents the results of the t-SNE projection method for each year.

We can observe in Figure 8 that the model identified the main component in the center of the figure and several small groups of around it. The main component represents the nominal case for transit images. The information for each year is shown in a different color. The surrounding groups represent atypical situations that do not match the nominal case, for example disruptions and strikes. Note that the forecasting period included the COVID 19 crisis, which impacted mobility. We can observe 3 significant groups of images from 2020 and 2021 (highlighted by circles in Figure 8)). They show the three French Covid-19 lockdowns. Although it is not shown in this visualization, time is the main characteristic of the latent space. Two images taken at the same time of day will be close in the latent space. On the contrary, an image from the morning and an image from the afternoon will be distant from each other. Thus, as we can expect, the model forecasts loads according to calendar information with an important distinction between images collected under atypical situations and the nominal case.

In-Depth analysis of forecasting results for atypical situations

In practice, a public transport operator evaluates transport supply on the basis of demand estimated by historical observation studies. Passenger load prediction can be trivial when there is no variation from the plan (timetables). However, the real supply can be uncertain and disturbed by atypical events (i.e., events that are not foreseen by the operator). Thus, we selected several real atypical scenarios based on three criteria: (i) subsequent knowledge of the metro traffic, (ii) descriptive statistics computed on the images, and (iii) latent space clustering. The goal was to evaluate the robustness of the framework and the proposed forecasting models for the whole test set and for specific test sets.

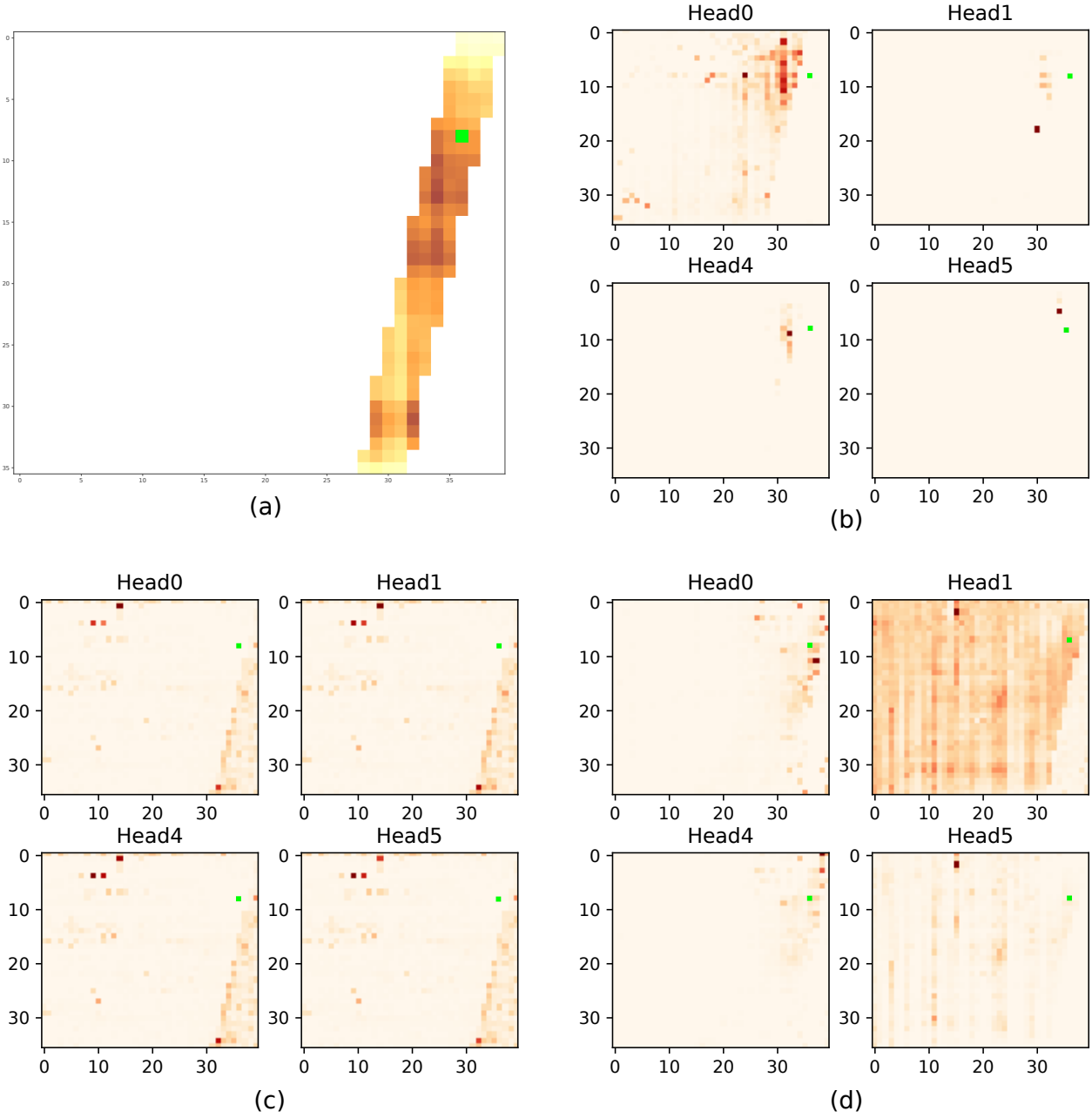

FIGURE 7: Visualisation of attention scores of 4 heads for a single pixel from the Channel Vision Transformer: (a) Studied $I(t)$ with the targeted pixel; (b) Attention scores from the passenger load channel; (c) Attention scores from the travel time channel; (d) Attention scores from the waiting time channel; [A deeper red color indicates a higher attention score]

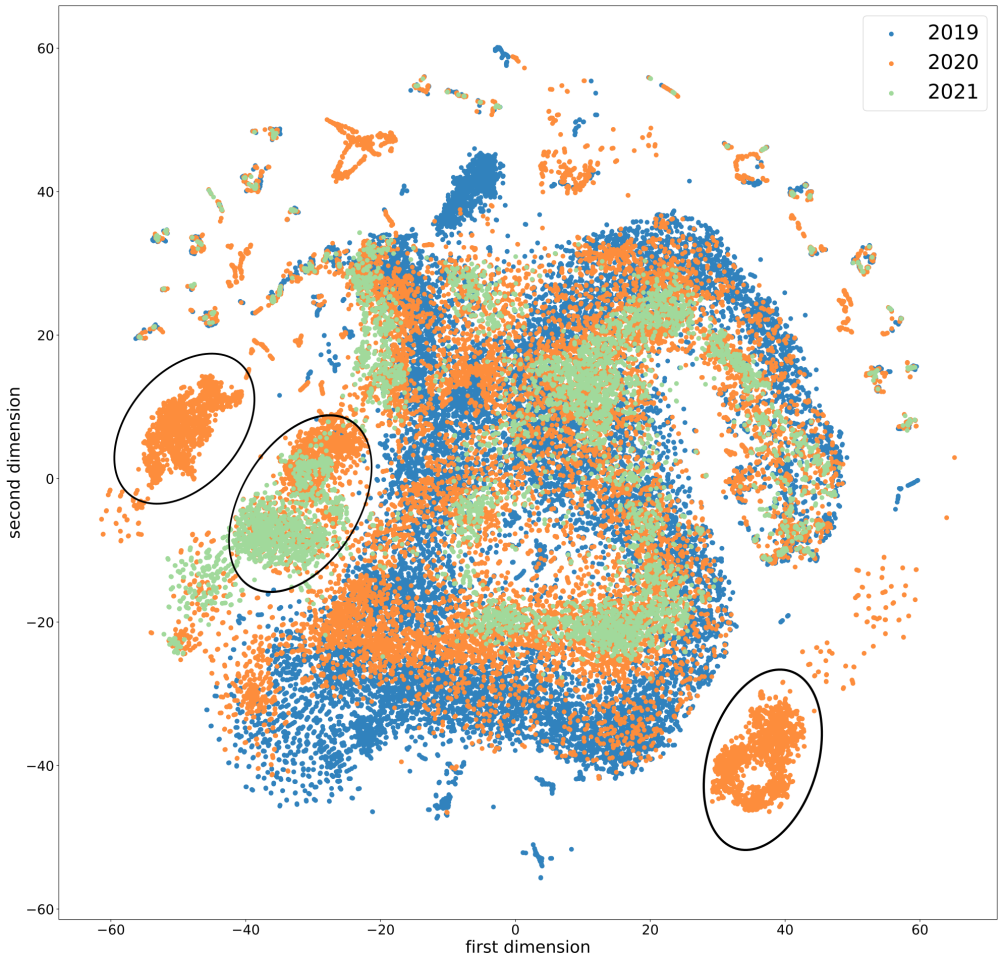

FIGURE 8: : Latent space of the U-Transformer obtained by the t-SNE representation on the annual data

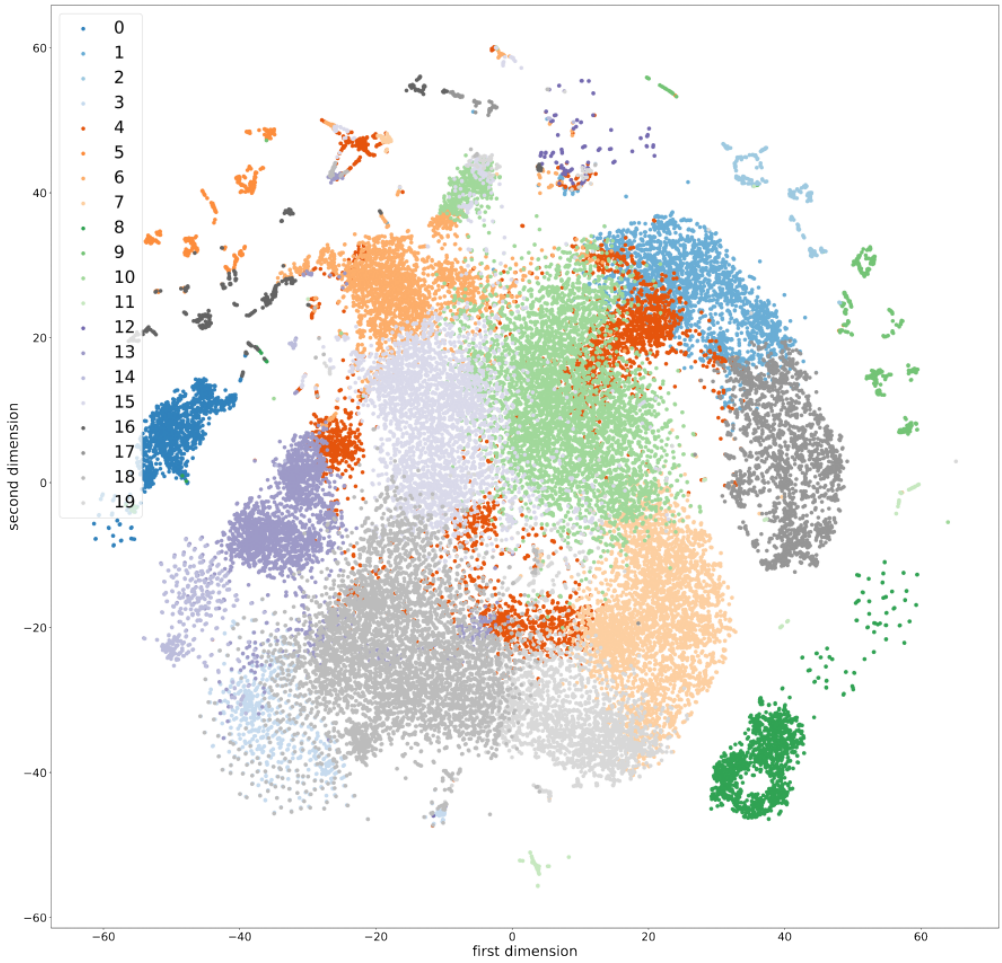

FIGURE 9: Latent space of the U-Transformer t-SNE representation. Different colors distinguish the 20 classes obtained from K-means clustering.

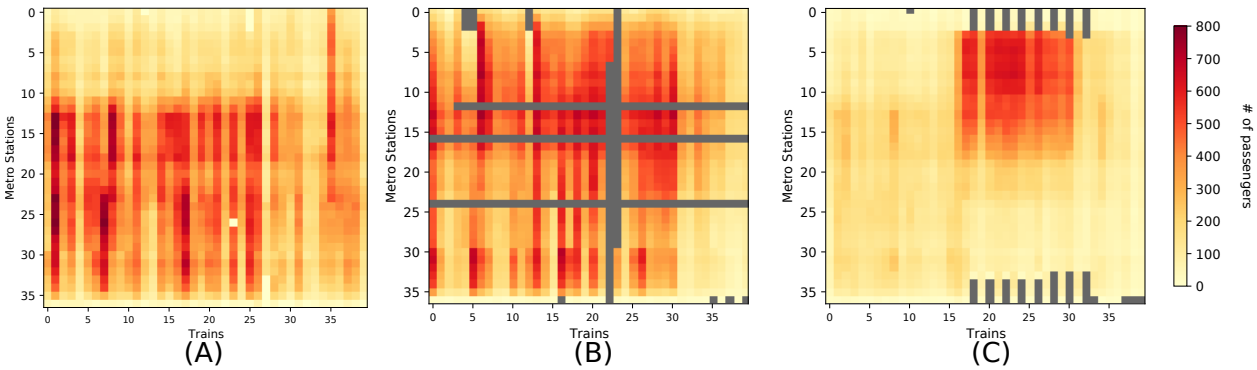

FIGURE 10: Images of the selected atypical groups: (A) Cluster 3 represents a sudden heavy load in the middle of the line (i.e., 11th station); (B) Cluster 12 represents a disrupted situation and a scenario with closed stations; (C) Cluster 18 represents a scenario with a sporting event when a high load moves directly from the first station of the metro line.

TABLE 4: Forecasting results based on weighted mean absolute percentage error (WMAPE) for the entire test set and the atypical situations.

Models (reference) / WMAPE [%] for	Testset	Delay	High load	Strike	Lockdown
Naïve	19.9	27.3	20.1	38.2	21.1
NN	18.9	28.4	20.1	40.2	26.8
CNN	13.1	24.2	13.0	29.8	18.6
CNN + NN (*17*)	14.8	24.2	14.3	32.4	25.9
U-net (*16*)	12.3	19.3	11.6	**28.1**	16.8
Transformer (*20*)	18.2	26.6	26.6	35.5	20.1
VIT-3 (*22*)	12.3	20.3	12.1	31.1	18.2
U-Transformer (ours)	**11.4**	**19.1**	**11.2**	29.6	**13.7**
Channel Vision Transformer **(ours)**	12.1	19.9	12.0	30.0	14.2

TABLE 5: Forecasting results based on weighted mean absolute percentage error (WMAPE) for the entire test set and specific atypical situations: Cluster 3 represents a sudden heavy load in the middle of the line (i.e., 11th station). Cluster 12 represents a disrupted situation and a scenario with closed stations; Cluster 18 represents a scenario with a sporting event.

Models (reference) / WMAPE [%] for	Testset	cluster 3	cluster 12	cluster 18
Naïve	19.9	28.9	**29.8**	22.3
NN	18.9	28.5	60.5	18.7
CNN	13.1	21.4	34.2	14.6
CNN + NN (*17*)	14.8	25.9	40.2	15.6
U-net (*16*)	12.3	19.5	30.4	13.9
Transformer (*20*)	18.2	31.2	75.1	18.2
VIT-3 (*22*)	12.3	20.9	52.8	13.1
U-Transformer (ours)	**11.4**	**18.4**	58.9	**12.5**
Channel Vision Transformer **(ours)**	12.1	19.5	51.6	13.1

The first criterion was based on the use of event databases and calendar information. We were able to identify two specific cases - *Strike* and *Lockdown* - which differed from the nominal cases. In the Paris Metropolis, there was a long transport strike in December 2019. The Lockdown data set is represented by images impacted by the COVID-19 crisis during March 2020. The second criterion was based on descriptive statistics computed on the images. For example, the distribution of the number of missing pixels or the maximum duration of a metro journey allowed us to identify disrupted events. We can then extracted various test sets. The first extracted set is called the *Delay* set, and involves images where the length of a metro journey exceeded the nominal case by more than 10 minutes. The *high load* set was created from images in which the average loads were higher than those in the nominal case. Finally, the last criterion was derived from the latent space of the U-Transformer. The approach was to apply unsupervised clustering to partition the latent space into a reduced set of clusters including some atypical groups of images as shown in the t-SNE visualization in the Figure 9. We chose to cluster the training images using a k-means algorithm with 20 clusters (the number of clusters was chosen according to operational considerations). The aim was to detect atypical clusters that were not extracted in an unsupervised way using the first two criteria. We excluded the Lockdown and Strike clusters that were visible in this latent space. Thus, we selected 3 clusters: Cluster 3, Cluster 12 and Cluster 18 which represent atypical situations from an operational point of view, accounting for a total of around

10% of the data. Figure 10 shows one image per cluster. Cluster 3 corresponds to the appearance of high loads from the 11th station on the line. Cluster 12 represents disrupted images or images with closed stations, while cluster 18 represents images with a high load at the beginning of the line (i.e. sporting events). Finally, the trained k-means algorithm was used on the images of the test set to extract atypical images from the selected clusters.

Table 4 and Table 5 summarize the results obtained from all the atypical groups and clusters presented previously. Prediction performance was evaluated with the WMAPE metric. With regard to these results, the U-Transformer model outperformed all the other models for all test cases, except for *Strike* and *Cluster* 12. The U-net model provided the best results for *Strike* with a WMAPE value of 28.1% and the Naïve model outperformed all the other models on the *Cluster* 12 test set obtaining a WMAPE value of 29.8%.

It is also noteworthy that delayed trains (e.g. images with atypical travel times and waiting times) had a greater impact on the forecasting results than high loads. The *Delay*, *Strike* and *Cluster* 12test sets had a WMAPE value that was at least 8% higher than the full test set. In contrast, the forecasting performance of the *high load*, *Lockdown* and *Cluster* 18 sets where the load was generally higher was close to the *nominal* performance given by the U-Transformer model. We observed a maximum difference of 3% between the WMAPE value for the test sets we have mentioned and the nominal case. Consequently, the results provide numerical proof that the proposed U-Transformer architecture can provide more accurate predictions of future train loads, particularly for atypical situations.

CONCLUSIONS AND PERSPECTIVES

In this paper, we have proposed a deep learning framework for the short-term prediction of train passenger loads in an urban public transit network. It combines vision transformer and inpainting approaches to forecast all the desired future train loads simultaneously (i.e., not recursively). Two new architectures (U-Transformer and Channel Vision Transformer) have been proposed based on self-attention in order to reformulate and solve the forecasting task as an inpainting problem. We have compared the performances of the proposed architectures with state-of-the-art methodologies on a real data set covering 3 years on a Paris metro line. The benchmark results show the effectiveness of the proposed methods over multiple test sets. In addition, the proposed approach outperforms other prediction models, particularly for atypical situations such as lockdown, strikes, high loads, and delays. The proposed methodology can classify atypical situations according to their type, e.g., strike or lockdown. We have identified 7 groups of atypical situations in this study (Tables 4 and 5). Moreover, we have performed an in-depth analysis based on attention scores and latent spaces to interpret the performances of the forecasting methods.

The proposed methodology can be extended in several directions. We are currently exploring two avenues. The first sets out to improve the performance of the forecasting models on atypical cases, by over-sampling atypical images from the latent space. We are also investigating a train-based model that considers our images as a sentence of a train courses where attention mechanisms can be applied to extract train features and complete future train sequences. Doing so, it allows reducing the allocation and usage of GPU memory during the forecasting process. Another possible perspective for this work would be to test our models on an public database in the framework of an open source project that our 3 years of data currently used does not allow. Moreover, the proposed methodology can predict supply and demand by reconstructing an image with a passenger load channel and a waiting time channel. Exploring the latent space provides us with a possible approach to achieve anomaly detection (*42*) in a transport network. This research has shown that images can provide an effective way of detecting abnormal situations in public transport.

REFERENCES

1. Kim, Y. J., S. Choi, S. Briceno, and D. Mavris, A deep learning approach to flight delay prediction. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, IEEE, 2016, pp. 1–6.
2. Hochreiter, S., The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 6, 1998, pp. 107–116.
3. Liyanage, S., R. Abduljabbar, H. Dia, and P.-W. Tsai, AI-based neural network models for bus passenger demand forecasting using smart card data. *Journal of Urban Management*, 2022.
4. Toqué, F., E. Côme, M. K. El Mahrsi, and L. Oukhellou, Forecasting dynamic public transport Origin-Destination matrices with long-Short term Memory recurrent neural networks. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1071–1076.
5. Zhao, Z., W. Chen, X. Wu, P. Chen, and J. Liu, LSTM network: a deep learning approach for short-term traffic forecast. *Iet Intelligent Transport Systems*, Vol. 11, 2017, pp. 68–75.
6. Liu, X., A. Gherbi, W. Li, and M. Cheriet, Multi Features and Multi-time steps LSTM

Based Methodology for Bike Sharing Availability Prediction. *Procedia Computer Science*, Vol. 155, 2019, pp. 394–401, the 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019),The 14th International Conference on Future Networks and Communications (FNC-2019),The 9th International Conference on Sustainable Energy Information Technology.

7. Wang, D., Y. Yang, and S. Ning, DeepSTCL: A Deep Spatio-temporal ConvLSTM for Travel Demand Prediction. In *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.
8. Colace, F., M. D. Santo, M. Lombardi, F. Pascale, D. Santaniello, and A. Tucker, A multi-level graph approach for predicting bicycle usage in London area. In *Fourth International Congress on Information and Communication Technology*, Springer, 2020, pp. 353–362.
9. Li, C., L. Bai, W. Liu, L. Yao, and S. T. Waller, Graph neural network for robust public transit demand prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2020.
10. Peng, H., H. Wang, B. Du, M. Z. A. Bhuiyan, H. Ma, J. Liu, L. Wang, Z. Yang, L. Du, S. Wang, et al., Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Information Sciences*, Vol. 521, 2020, pp. 277–290.
11. Wu, W., Y. Xia, and W. Jin, Predicting bus passenger flow and prioritizing influential factors using multi-source data: Scaled stacking gradient boosting decision trees. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 4, 2020, pp. 2510–2523.
12. Egu, O. and P. Bonnel, Medium-term public transit route ridership forecasting: What, how and why? A case study in Lyon. *Transport Policy*, Vol. 105, 2021.
13. Roos, J., S. Bonnevay, and G. Gavin, Short-Term Urban Rail Passenger Flow Forecasting: A Dynamic Bayesian Network Approach. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016, pp. 1034–1039.
14. Heydenrijk-Ottens, L., V. Degeler, D. Luo, N. Van Oort, and H. Van Lint, Supervised learning: Predicting passenger load in public transport. In *CASPT Conference on Advanced Systems in Public Transport and TransitData*, 2018, pp. 30–32.
15. Pasini, K., M. Khouadjia, A. Samé, F. Ganansia, and L. Oukhellou, LSTM Encoder-Predictor for Short-Term Train Load Forecasting. In *ECML/PKDD*, 2019.
16. Bapaume, T., E. Côme, J. Roos, M. Ameli, and L. Oukhellou, Image Inpainting and Deep Learning to Forecast Short-Term Train Loads. *IEEE Access*, Vol. 9, 2021, pp. 98506–98522.
17. Ma, X., Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors*, Vol. 17, 2017, p. 818.
18. Liu, Y., S. Dutta, A. W.-K. Kong, and C. K. Yeo, An Image Inpainting Approach to Short-term Load Forecasting. *IEEE Transactions on Power Systems*, 2022.
19. Zhai, X., A. Kolesnikov, N. Houlsby, and L. Beyer, Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12104–12113.
20. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention Is All You Need*, 2017.
21. Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2018.
22. Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. De-

hghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, 2021.

23. Chen, C., Y. Liu, L. Chen, and C. Zhang, Bidirectional Spatial-Temporal Adaptive Transformer for Urban Traffic Flow Forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
24. Wu, J., Q. Wu, J. Shen, and C. Cai, Towards attention-based convolutional long short-term memory for travel time prediction of bus journeys. *Sensors*, Vol. 20, No. 12, 2020, p. 3354.
25. Hao, S., D.-H. Lee, and D. Zhao, Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system. *Transportation Research Part C: Emerging Technologies*, Vol. 107, 2019, pp. 287–300.
26. Du, B., H. Peng, S. Wang, M. Z. A. Bhuiyan, L. Wang, Q. Gong, L. Liu, and J. Li, Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 21, No. 3, 2019, pp. 972–985.
27. Bertalmio, M., G. Sapiro, V. Caselles, and C. Ballester, Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 417–424.
28. Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, Vol. 3, 2014.
29. Yu, Y., T. Jiang, J. Gao, H. Guan, D. Li, S. Gao, E. Tang, W. Wang, P. Tang, and J. Li, CapViT: Cross-context capsule vision transformers for land cover classification with airborne multispectral LiDAR data. *International Journal of Applied Earth Observation and Geoinformation*, Vol. 111, 2022, p. 102837.
30. Khan, S., M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021.
31. Liu, L., W. Hamilton, G. Long, J. Jiang, and H. Larochelle, A universal representation transformer layer for few-shot image classification. *arXiv preprint arXiv:2006.11702*, 2020.
32. Gao, Y., M. Zhou, and D. N. Metaxas, UTNet: a hybrid transformer architecture for medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2021, pp. 61–71.
33. Cao, H., Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation. *arXiv preprint arXiv:2105.05537*, 2021.
34. Beal, J., E. Kim, E. Tzeng, D. H. Park, A. Zhai, and D. Kislyuk, Toward transformer-based object detection. *arXiv preprint arXiv:2012.09958*, 2020.
35. Carion, N., F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, End-to-end object detection with transformers. In *European conference on computer vision*, Springer, 2020, pp. 213–229.
36. Lin, C.-H., E. Yumer, O. Wang, E. Shechtman, and S. Lucey, St-gan: Spatial transformer generative adversarial networks for image compositing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9455–9464.
37. Chen, J., Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

38. Petit, O., N. Thome, C. Rambour, L. Themyr, T. Collins, and L. Soler, U-net transformer: Self and cross attention for medical image segmentation. In *International Workshop on Machine Learning in Medical Imaging*, Springer, 2021, pp. 267–276.
39. Sheng, H., S. Cai, Y. Liu, B. Deng, J. Huang, X.-S. Hua, and M.-J. Zhao, Improving 3d object detection with channel-wise transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2743–2752.
40. Chen, L., H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5659–5667.
41. Hinton, G. E. and S. Roweis, Stochastic neighbor embedding. *Advances in neural information processing systems*, Vol. 15, 2002.
42. Wang, X., A. Fagette, P. Sartelet, and L. Sun, A Probabilistic Tensor Factorization Approach to Detect Anomalies in Spatiotemporal Traffic Activities. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1658–1663.