Large-Scale Measurements and Prediction of DC-WAN Traffic

Zhaohua Wang, Zhenyu Li, Heng Pan, Guangming Liu, Yunfei Chen, Qinghua Wu, Gareth Tyson, and Gang Cheng

Abstract—Large cloud service providers have built an increasing number of geo-distributed data centers (DCs) connected by Wide Area Networks (WANs). These DC-WANs carry both high-priority traffic from interactive services and low-priority traffic from bulk transfers. Given that a DC-WAN is an expensive resource, providers often manage it via traffic engineering algorithms that rely on accurate predictions of inter-DC high-priority (delay-sensitive) traffic. In this paper, we perform a large-scale measurement study of high-priority inter-DC traffic from Baidu. We measure how inter-DC traffic varies across their global DC-WAN and show that most existing traffic prediction methods either cannot capture the complex traffic dynamics or overlook traffic interrelations among DCs. Building on our measurements, we propose the ***In**terrelated-**Te**mporal **G**raph Convolutional **Net**work* (IntegNet) model for inter-DC traffic prediction. In contrast to prior efforts, our model exploits both temporal traffic patterns and inferred co-dependencies between DC pairs. IntegNet forecasts the capacity needed for high-priority traffic demands by accounting for the balance between resource provisioning (*i.e.,* allocating resources exceeding actual demand) and QoS losses (*i.e.,* allocating fewer resources than actual demand). Our experiments show that IntegNet can keep a very limited QoS loss, while also reducing overprovisioning by up to 42.1% compared to the state-of-the-art and up to 66.2% compared to the traditional method used in DC-WAN traffic engineering.

Index Terms—Data Center Networks, Traffic measurement, Traffic Prediction, Graph Convolutional Network

1 Introduction

Large cloud service providers often use tens of geographically distributed data centers (DCs), that are interconnected by a wide-area network (WAN), to host diverse services. Services are replicated across these DCs to process users' requests locally for better Quality-of-Experience (QoE). As such, large volumes of data are transferred among DCs for synchronization and backup purposes. These data center wide-areas networks (DC-WANs) also carry delay-sensitive services that have stringent Quality of Service (QoS) requirements, *e.g.,* responding to user web requests [1]. Delay-sensitive traffic is usually classified as high-priority, while bulk transfers receive a best-effort service using the remaining bandwidth resources. A DC-WAN, however, is an expensive resource, and providers have to optimize the bandwidth allocation with traffic engineering (TE) solutions to make full use of the network.

In recent years, there have been several DC-WAN TE solutions based on software-defined networking (SDN) [2], [3] or fine-grained policy enforcement [4], [5]. These methods tend to rely on predictions of inter-DC traffic, especially delay-intolerant *high-priority* traffic for interactive services. These predictions are used to inform TE and bandwidth allocation in DC-WANs. Most notably, they involve reserving a certain portion of capacity to guarantee the QoS for high-priority traffic. The remaining capacity is then allocated to low-priority traffic for network data transfers across geo-distributed data centers [6], [7], [8], [9], [10], [11], [12].

Accurate inter-DC traffic prediction has many challenges. As described in previous work [13], inter-DC traffic is harder to predict than many other types of network traffic. This is because it is dominated by large applications that contribute to complicated aggregated traffic patterns. Existing DC-WAN traffic prediction methods often use statistical models, *e.g.,* moving average (MA), exponentially weighted moving average (EWMA), and Auto-Regressive Integrated Moving Average (ARIMA). However, these traditional linear models are not able to characterize high-dimensional and non-linear temporal patterns [14], and thus yield low accuracy. Despite this, TE solutions, such as SWAN [2] and BwE [4], assume the ability to accurately predict traffic. Although these studies utilize real inter-DC traffic traces for evaluation, the WAN traffic patterns of large-scale production DCs remain poorly understood. To accurately predict traffic, neural networks (*e.g.,* LSTM, ANN) have also been used to capture the non-linear temporal dynamics for link utilization [13]. Nevertheless, applying these without an understanding of inter-DC traffic patterns will not yield high accuracy. For instance, we later show that information about inter-dependencies between DC pairs can improve prediction accuracy (which was largely ignored by prior studies). In addition, existing traffic prediction methods used in DC-WAN TE are designed to guarantee enough capacity for high-priority traffic by simply inflating the predicted traffic demands according to historical estimation errors. A large

- *Corresponding author: Zhenyu Li.*
- *Zhaohua Wang, Zhenyu Li, Heng Pan and Qinghua Wu are with the Institute of Computing Technology, Chinese Academy of Sciences, and University of Chinese Academy of Sciences. Zhenyu Li, Heng Pan and Qinghua Wu are also with the Purple Mountain Laboratories.*
  (E-mail: wangzhaohua@ict.ac.cn, zyli@ict.ac.cn, panheng@ict.ac.cn, wuqinghua@ict.ac.cn).
- *Guangming Liu, Yunfei Chen, and Gang Cheng are with Baidu Inc.*
  (E-mail: liuguangming@baidu.com, chenyunfei@baidu.com, chenggang06@baidu.com).
- *Gareth Tyson is with the Hong Kong University of Science & Technology (GZ) and Queen Mary University of London.*
  E-mail: gtyson@ust.hk.

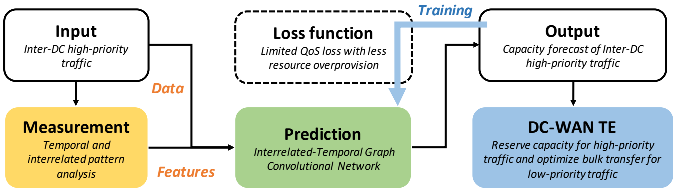

Input
Inter-DC high-priority traffic
Loss function
Limited QoS loss with less resource overprovision
Training
Output
Capacity forecast of Inter-DC high-priority traffic
Data
Measurement
Temporal and interrelated pattern analysis
Features
Prediction
Interrelated-Temporal Graph Convolutional Network
DC-WAN TE
Reserve capacity for high-priority traffic and optimize bulk transfer for low-priority traffic

Fig. 1: Overview of the role of our work in DC-WAN traffic engineering.

prediction error can lead to traffic overprovisioning, which in turn lowers the utilization of WAN bandwidth resources. Thus, the reserved capacity should be forecast by accounting for a balance between the QoS loss of high-priority services and resource overprovisioning.

To address these challenges, in this paper, we focus on understanding the patterns of high-priority inter-DC traffic and develop techniques to forecast the capacity required by high-priority traffic of individual DC pairs, for traffic engineering purposes. Figure 1 outlines the role of our work in DC-WAN TE. We argue that accurately estimating *input* high-priority inter-DC traffic requires a deeper understanding of traffic dynamic patterns in large-scale DC networks. Motivated by this, we offer the first *measurement study* of high-priority DC-WAN traffic patterns in a large-scale DC network with respect to temporal and interrelated traffic pattern analysis. Based on our findings, we propose a learning-based traffic *prediction* method (*i.e.,* IntegNet) that makes use of input features to provide the *output* capacity forecast. During the model training phase, the output is used to evaluate a *loss function* that quantifies the prediction error, which is designed to account for the cost balance between resource overprovisioning and QoS losses of high-priority services. Finally, the forecast capacity is fed to *DC-WAN traffic engineering*, in which the forecast capacity is reserved for high-priority services and the remaining capacity is allocated to optimize bulk transfers.

To this end, we gather Netflow data from the data center network of Baidu, a large-scale web and cloud provider with tens of geo-distributed DCs serving millions of users each day. Its geo-distributed DCs, with a complex service mix, make it an interesting exemplar for the examination of traffic patterns across modern DCs. Using this Netflow data, we first perform a large-scale analysis of high-priority traffic patterns in Baidu's DC-WAN. As an extension to our previous work [15], here we focus on exploring the temporal traffic patterns of individual DC pairs and how these patterns correlate with each other across DC pairs. We find that the aggregated traffic (which is relatively stable) provides limited insight into the patterns of individual pairs of DCs (which tend to exhibit far greater variability). We then reveal that the various traffic dynamics of different services lead to these complicated temporal patterns. We also explore how this variability correlates across DCs, and find that topological and service dependencies between DC pairs predictably impact these patterns.

Motivated by the above observations, we choose to utilize neural network models to capture the high-dimensional temporal traffic patterns. We also propose to organize the network traffic between DCs as a graph structure based on the topology- and service-level relationships and exploit a graph convolution to directly extract the traffic correlation features. These two components lead to our design of the Interrelated-Temporal Graph Convolutional Networks (IntegNet) model for predicting the capacity needed for high-priority inter-DC traffic demands. In contrast to prior works, IntegNet exploits both temporal features and the information about the topological/service relationships between DC pairs. Besides, we propose a novel customized loss function to evaluate the prediction accuracy, which is meaningful for balancing the QoS losses and resource overprovisioning.

We evaluate IntegNet using the Netflow traffic data collected in Baidu's DC-WAN. The experimental results show that our model outperforms other state-of-the-art methods in balancing underestimation (which leads to QoS losses for high-priority services) and overestimation (which leads to resource overprovisioning). IntegNet reduces the overprovisioning by 66.2%, 57.6%, 42.1%, 41.6% compared with MA, EWMA, LSTM, and GAT, respectively. This is achieved with limited QoS loss for under 0.3% of the total inter-DC traffic. We also show that the delay for computing predictions is about 10^{-3} seconds on a mid-range server, which is practical for DC-WAN traffic engineering. Our findings have key implications for improving DC-WAN traffic engineering. To summarize, the main contributions of this paper are:

- **Inter-DC Traffic Analysis** (§5): By analyzing Netflow data of inter-DC traffic collected in Baidu's DC-WAN, we find that the stability of high-priority traffic varies heavily across DC pairs. The various traffic dynamics of different services lead to this complicated temporal pattern. We also observe correlations among the traffic dynamics between DC pairs. In particular, the most strongly-correlated DC pairs have topological relationships, or host similar services.
- **Inter-DC Traffic Prediction** (§6): Exploiting the above findings, we design the IntegNet model for predicting high-priority inter-DC traffic demands. IntegNet combines a Temporal Convolution layer (TCN) to capture temporal traffic dependencies, and an Interrelated Graph Convolution (GCN) layer to capture inter-DC traffic correlations. Based on these features, IntegNet generates predicted traffic rates between DC pairs for the next time segment. We also propose a customized loss function to quantify the prediction error, which is designed to account for the cost balance between overestimation and underestimation.
- **Evaluation** (§7): We conduct comprehensive experiments to evaluate IntegNet using data from Baidu's DC-WAN. Our results demonstrate the effectiveness of the IntegNet model in improving the inter-DC traffic prediction performance with respect to both resource provisioning and QoS losses. IntegNet can be integrated with popular DC-WAN traffic engineering solutions for online traffic demand prediction.

The rest of this paper is organized as follows. §2 introduces the related work. §3 and §4 describes the background and data collection methodology. §5 measures the high-priority WAN traffic characteristics with respect to temporal dynamics among data centers. §6 describes the design of our Interrelated-Temporal Graph Convolutional Network model. §7 evaluates IntegNet. Finally, §8 concludes the paper.

2 Related Work

NetFlow-based Measurements of DCNs. NetFlow is a widely used monitoring tool with a variety of applications in data center networks. Specifically, network operators and managers rely on sampled NetFlow data to study traffic characteristics, *e.g.,* traffic demand, communication patterns, and traffic stability [16], [17], [18]. These studies provide network operators with important insights for network fabric design and service deployment. There are also several traffic engineering solutions in DC-WANs that utilize NetFlow data to evaluate the effectiveness of proposed methods [2], [5], [12]. In this paper, we use sampled NetFlow data for the analysis of variation of DC-WAN traffic.

DC-WAN Traffic Analysis. There have been a range of measurement studies reporting the nature of traffic within DCs, including from Microsoft [16], [19], [20], [21], and Facebook data centers [18]. Based on various data sources (*i.e.,* NetFlow, SNMP, socket-level logs and packet traces) these measurements primarily focus on traffic characteristics inside DCs in terms of traffic exchange, flow characteristics and packet arrival patterns. In contrast, our work focuses on *inter*-DC traffic. In this regard, our work is closely related to [17], which studied the inter-DC traffic in Yahoo! However, their analysis focuses on analyzing the IP communication patterns and correlations with client traffic characteristics. Further, their scale (5 DCs) is much smaller than ours, and the service mix is much simpler.

In our previous work [15], we examined WAN traffic characteristics with NetFlow and SNMP data in Baidu's DCN from the perspectives of traffic demand, traffic communication among DCs, and the traffic patterns of diverse services. Specifically, we revealed high predictability of overall traffic demands, as well as disparity of WAN traffic among services. These observations motivate the need for more accurate estimation methods for DC-WAN traffic engineering at the service level. This paper gives further insights into the various traffic dynamic patterns of individual DC pairs and how this variability correlates across DC pairs.

DC-WAN Traffic Prediction. Many traffic engineering methods in DC-WANs have been designed to estimate high-priority traffic for efficient bandwidth allocation. These works usually rely on traditional statistical methods such as MA [2], [4], [5] and EWMA [7] to perform traffic prediction. However, we later show that these linear models are not able to capture the complex dynamics of inter-DC traffic, resulting in poor estimation accuracy. Li *et al.* [13] combine wavelet transformation with artificial neural network (ANN) to improve prediction accuracy of the *aggregated* WAN traffic for a specific data center. However, as discussed in §5, the aggregated traffic pattern differs radically from individual pairs of DCs. In contrast, our solution focuses on accurate traffic estimation for *individual* DC pairs.

ISP Network Traffic Prediction. There is also work in traffic prediction for ISP networks. These methods vary from statistical time-series methods to ANN-based models. Yoo *et al.* [22] developed a prediction model with Seasonal Decomposition of Time Series by Loss (STL) and ARIMA on SNMP data. Some works [23], [24], [25], [26], [27], [28] leverage RNN models (*e.g.,* LSTM, GRU) to capture temporal dependencies of traffic matrices for traffic prediction. Bega *et al.* designed DeepCog [29] by utilizing 3D-CNN models to capture both temporal and spatial correlations of traffic demands of network slices for capacity forecasting in 5G networks. Other works propose hybrid deep learning models for spatiotemporal predictions [30], [31], [32], [33]. They utilize CNN-based or autoencoder-based models for spatial modeling and use LSTM-based models for temporal modeling. These works capture the spatial correlations among traffic flows based on geographical information or from the network topology perspective.

In contrast, our IntegNet model captures not only the temporal dependencies, but also the topological and service-level correlation between DC pairs. Such a correlation may not hold in an ISP network (*e.g.,* ISP links may exhibit limited service-level similarity). We also note that the use of GCN-based models (as opposed to CNN and RNN) has also proved effectiveness in other domains [34], [35], [36], [37]. We contribute to this wider field of literature, showing the efficacy of GCN solutions for traffic prediction.

Bandwidth Provisioning. traffic engineering solutions in DCNs [2], [4], [10] reserve bandwidth for the high-priority traffic by inflating the predicted demand with an empirical prediction error. In doing so, they aim at providing sufficient bandwidth for high-priority traffic. However, the poor prediction accuracy with blind inflation will lead to bandwidth overprovisioning, and lower the utilization of WAN links. Some studies have noticed the importance of balancing the resource overprovisioning and possible QoS losses due to resource underprovisioning. Krithikaivasan *et al.* introduce a forecast cost function [38] that is defined by allowing a different penalty associated with the under- and over-forecast for ISP network traffic prediction. Bega *et al.* design a loss function [29] that is tailored to account for 5G operators' desired balance between resource overprovisioning and service request violations, so as to minimize the economic cost. Inspired by these studies, the loss function in our model also accounts for the cost balance between overestimation and underestimation.

3 Background and Motivation

In this section, we first introduce key concepts on DC-WAN traffic engineering. We then briefly describe the large-scale DC network that we examine, and the diverse services hosting in DCs.

3.1 DC-WAN Traffic Engineering

DC-WANs carry both *high-priority* and *low-priority* traffic. Typically, high-priority traffic is driven by Internet-facing requests (*e.g.,* web search queries) and thus is delay-sensitive. Low-priority traffic is usually from batch computing services (*e.g.,* Hadoop, Spark) and thus can tolerate delays within pre-assigned deadlines. The WAN that interconnects DCs is an expensive resource. Most DC-WAN traffic engineering solutions aim at making full use of bandwidth resources, guaranteeing preferential provision for high-priority traffic [2], [3], [4], [5]. To this end, DC-WAN traffic

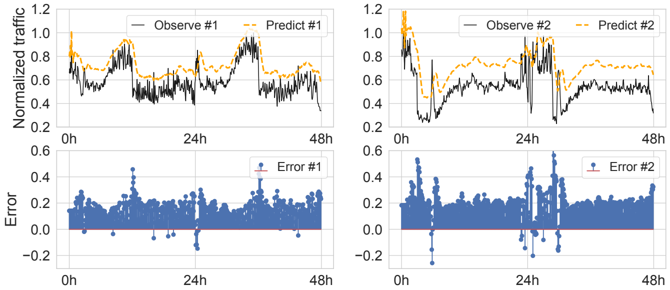


Fig. 2: Top: observed and predicted high-priority traffic demands of two example DC pairs at 5-minute intervals during two days; using traditional MA-based prediction method [2], [4], [10]. Bottom: overprovisioning (positive) and unserviced (negative) traffic demands over time. The traffic volume is normalized by the maximum volume.

engineering methods usually estimate traffic demands of high-priority services and then reserve this capacity to guarantee the service. The remaining capacity is then allocated to low-priority traffic.

In practice, we observe that about 38% of inter-DC traffic is generated from high-priority services, which exhibit peaks and valleys traffic patterns (as shown in Figure 2). This proportion is higher than the range of 5%-20% observed in prior works [2], [3], [10]. As such, the capacity forecast of high-priority inter-DC traffic is of great importance for DC-WAN traffic engineering. Accurate predictions can save capacity resources for low-priority traffic while guaranteeing the quality of high-priority services (QoS). The reserved capacity is required to account for the balance between the loss of QoS (*i.e.,* allocating fewer resources than the actual demand) and resource overprovisioning (*i.e.,* allocating resources exceeding actual demand).

Existing methods used by Software Defined WANs (SD-WANs) often rely on the average or median traffic volume in the last few time slots to estimate the high-priority services' demand [2], [5]. However, as we will show in this paper, high-priority traffic in some WAN links may experience significant variations over short time periods, resulting in a large estimation error. In order to guarantee enough capacity for high-priority traffic demands, a popular solution to compensate is setting aside "headroom" [2], [4], [10], where the size of the headroom is dependent on the empirical prediction error. A larger prediction error will require more headroom (*i.e.,* more resource overprovisioning), which in turn leaves less room for low-priority traffic, lowers the utilization of WAN links and degrades the performance for bulk transfers. As exemplified in Figure 2, this solution leads to large resource overprovisioning over most of the time. The lower graphs highlight significant wasteful overprovisioning (using mean average predictions). Despite this, the quality of high-priority services still meets losses when traffic surges occur.

Therefore, the focus of our work is to precisely predict the capacity required by high-priority inter-DC traffic. Our goals are two-fold: (*1*)To guarantee the QoS of high-priority services with little unserviced traffic demands; and (*2*)To reduce traffic overprovisioning to save bandwidth resources.

3.2 Baidu's Data Center Network

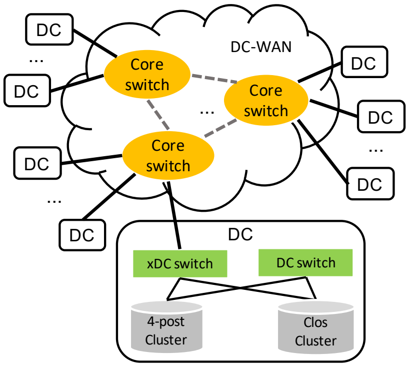


Fig. 3: Overview of Baidu's data center network.

Our paper is underpinned by a large-scale dataset collected across Baidu's DC network (DCN). This DCN hosts services, ranging from traditional web search to emerging vehicle auto-driving. Parts of the services (*e.g.,* web search, location-based services) directly serve tens of millions of users every day, while others (*e.g.,* Hadoop) act as the infrastructure to support Internet-facing services. The DC network is built on an infrastructure of DCs connected through high bandwidth (multiple Tbps) wide area networks (WANs). Figure 3 depicts the *simplified* topology of Baidu's DCN. DCs connect to the WAN via the core switches, and form a full-meshed core network at the overlay layer. The traffic that goes out of a DC flows through xDC (cross-DC) switches to the core switches, while the traffic destined to servers within clusters is transferred via DC switches. Each cluster either employs a typical 4-post structure or a Spine-Leaf Clos design. Overall, Baidu's DCN topology is similar to others (*e.g.,* Facebook [18], Microsoft [16]). However, Baidu's DCN hosts many services that have not been reported in other DC networks, such as emerging distributed AI and vehicle auto-driving services [39]. This makes it an interesting exemplar for measuring modern DC-WAN traffic patterns.

There are over 1K services hosted in Baidu's DCN. For context, these services can be divided into 10 categories based on their functionalities: *Web* (search engine), *Computing* (batch computing), *Analytics* (news feeds, ads and user behavior analysis), *DB* (database), *Cloud* (cloud storage and computing), *AI* (distributed machine learning), *File System* (distributed file systems), *Map* (location-based services), *Security* (security management) and Others. We argue that this variety of commercial services provides the research community with a unique opportunity to understand the traffic patterns of modern DC-WAN.

4 Data Collection Methodology

We have gathered 18 days of Netflow data from Baidu's DCN. This section provides a brief overview of the data collection methodology.

4.1 Traffic Data Collection

We rely on Cisco's Netflow service to gather traffic data from Baidu's DCN. It provides access to summarized IP

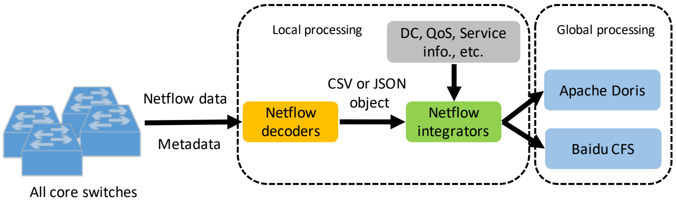


Fig. 4: Netflow data collection architecture.

flow records within the networks [40]. Figure 4 summarizes the Netflow data collection. We collect Netflow data from *all* the core switches in Figure 3 across Baidu's entire DCN for our analysis of inter-DC traffic. This results in an average of over 10 TB of raw Netflow data each day.

The active timeout for NetFlow on all the core switches is set to 1 minute.[1] Each flow records the aggregated flow information obtained from the sampled packet headers with 1:1024 sampling rate; a log contains the source and destination IP addresses, transport-layer port numbers and IP protocol. The collected flow data, along with other metadata such as the machines' IP addresses and capture timestamps, are first processed by *Netflow decoders*, which convert each log into a CSV or JSON object.[2] The parsed data is then streamed to *Netflow integrators* using a distributed subscribing and streaming system.

The *Netflow integrators* aggregate the traffic flow data at one minute intervals and further annotate records with additional information such as the DC, service and QoS information (indicating the priority of the flow) corresponding to each flow log. The *DC information* is identified via querying a directory that maintains the mapping between IP addresses and DCs. Since flows that transferred across different core switches are not repeatedly counted, we merge flows that belong to the same DC pair to identify inter-DC traffic. The *service information* is identified via querying a directory that keeps the mapping between IP addresses and port numbers to services. The priority of the flow is marked by the end server in each packet using the DSCP field, which is identified as *high-priority* and *low-priority* according to the sensitivity to loss and delay.

Netflow integrators then feed data into *Apache Doris*, a fast analytics database [41] and *Baidu CFS*, a bespoke cloud file system built for data storage. For each core switch, *Netflow Decoders* and *Netflow Integrators* are distributed locally in multiple DCs that connect to it for processing the collected flow data (*i.e.*, local processing). The data analytics and storage systems (*i.e.*, *Apache Doris* and *Baidu CFS*) are centrally deployed in a set of DCs for processing the global flow data from all the core switches across the DCN (*i.e.*, global processing). Note that, during the collection of the data used in this paper, we did not notice any abnormalities in our Netflow data collection system.

4.2 Dataset Summary

We collected the Netflow data for a period of 18 days. In this paper, we focus on forecasting capacity demands of *high-priority* inter-DC traffic (due to its importance in traffic engineering solutions). Thus, we then extract all traffic labeled high-priority by the DSCP field.

We aggregate the high-priority inter-DC traffic into 5-minute intervals (5,184 inter-DC traffic matrices in total).[3] We find a skewed traffic distribution, where 25% of DC pairs contribute 99% of high-priority traffic. Moreover, the set of heavy hitters that contribute 99% of traffic remains the same over time. Besides, among the remaining DC pairs accounting for the last 1% of high-priority traffic, the average traffic volume is below 100Mbps. The small traffic volume of these DC pairs has little impact on resource capacity and does not require a sophisticated forecast. A reserved capacity can be directly given based on the historical traffic values. We therefore focus on the traffic of 330 heavily loaded DC pairs, covering 30 DCs. Note, we also analyze the service composition in high-priority traffic of these heavily loaded DC pairs. We find that the traffic distribution across different types of services remains relatively stable over time.

5 Inter-DC Traffic Dynamics Analysis

In this section, we provide an analysis of high-priority WAN traffic dynamic characteristics in Baidu's DCN. We first analyze the overall temporal dynamics of inter-DC traffic (§5.1), and then look into the dynamic patterns at the service level (§5.2). We further investigate the correlation of traffic dynamics among DC pairs (§5.3). Our analysis sheds light on the potential of inter-DC traffic prediction, which we later use to underpin §6.

5.1 Overview of Traffic Dynamics

Inter-DC Traffic Dynamics. We first examine the variations in aggregated high-priority traffic $T(t)$ by computing its change rate, $r_{Agg.}$, as:

$$r_{Agg.}(t) = \frac{|T(t+\tau) - T(t)|}{T(t)} \tag{1}$$

Figure 5 (upper plot) presents the variation of the aggregated traffic (normalized to the maximum traffic volume) at 5-minute intervals during three days. We see a clear diurnal trend across days, and also relatively stable patterns over short time periods (*e.g.*, 5-minute intervals). This is evidenced by the low change rate ($r_{Agg.}$) of the aggregated traffic (about 1.1% in average and 0.8% in median).

To further reveal the traffic dynamics of individual DC pairs, for each time period, we represent the high-priority traffic between DCs as a traffic matrix (TM). We then examine the variation of the inter-DC traffic matrix over time. Specifically, for a time point t, the change rate r_{TM} of the traffic matrix TM is computed as [20]:

$$r_{TM}(t) = \frac{|TM(t+\tau) - TM(t)|}{|TM(t)|} \tag{2}$$

where the numerator is the absolute sum of the entry wide differences of the two matrices at adjacent time intervals, and the denominator is the absolute sum of entries in

[1] Flows longer than 1 minute are stored across multiple records.

[2] Incorrectly formatted records are discarded. The percentage of failed records is around 0.00001%.

[3] Traffic engineering in a DC-WAN is usually scheduled at the granularity of 5 minutes.

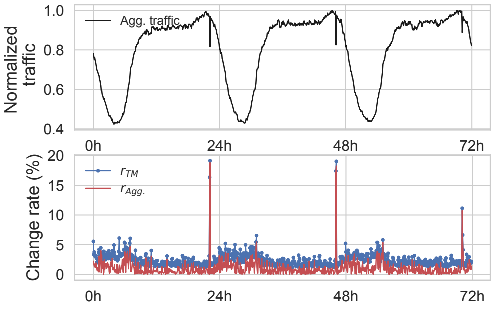


Fig. 5: Change rate of the aggregated high-priority traffic and the traffic matrix between DCs at 5-minute intervals during three days.

$TM(t)$, which equals to the aggregated traffic $T(t)$. Note that even if the aggregated traffic remains unchanged (*i.e.*, $r_{Agg.} = 0$), the exchanged traffic patterns among DCs (measured by r_{TM}) may change greatly. To understand the variations in traffic exchanged among DCs, Figure 5 (lower) plots the r_{TM} over time. We observe that the traffic exchanged across DCs is more dynamic than the aggregate traffic for most of the time intervals (about 2.5% in average and 2.2% in median). In particular, the inter-DC traffic shows variations during certain time intervals, even when the aggregated traffic remains almost unchanged (*i.e.*, $r_{Agg.}$ close to 0).

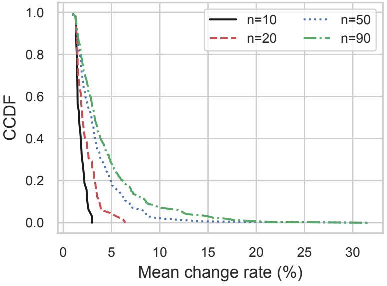
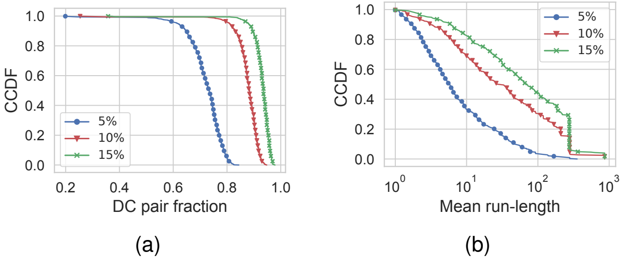


Fig. 6: The distribution of the average change rate of the high-priority traffic for top n% of DC pairs (sorted by traffic volume in descending order).

The above analysis indicates that, although overall inter-DC traffic is relatively stable over time, the traffic exchanged amongst DCs varies significantly. The change rate of the traffic matrix TM may be dominated by those DC pairs with large traffic volumes. Thus, we further investigate the high-priority traffic dynamics of each DC pair and observe a wide difference in dynamic patterns. 50% of the least dynamic DC pairs have an average change rate of below 3.5%, while another 25% (10% resp.) of the most dynamic pairs have an average change rate of above 6.1% (10.5% resp.). We then compare the traffic dynamics of DC pairs with different traffic volumes in Figure 6. We see that the DC pairs with larger traffic volumes show a remarkably stable traffic pattern. Specifically, the average change rates of the top 10% of DC pairs ($n = 10$) are lower than 5%. As more DC pairs with smaller traffic volumes are included, the fraction of pairs with stable traffic patterns decreases significantly. This finding indicates that accurate traffic predictions for the DC pairs with small traffic volumes are particularly challenging.



(a)

(b)

Fig. 7: High-priority traffic stability with three different change rate thresholds. (a) Distr. of the proportion of DC pairs with insignificant traffic change; (b) Distr. of the run-length of insignificant traffic change for individual DC pairs.

Per-DC Pair Traffic Stability. We next explore how the stability of traffic exchanged between DCs changes over time and evaluate the potential of high-priority inter-DC traffic prediction on a 5-minute time scale. To this end, we first compute the proportion of DC pairs that have no significant change in traffic volume at each time interval. For this, we extract the pairs that have a change rate below a predefined threshold, thr. Figure 7a presents the Complementary Cumulative Distribution Function (CCDF), where we experiment with three values of thr: 5%, 10% and 15%.

We see that over 65% of DC pairs remain stable (with $thr = 5\%$) for 80% of 5-minute intervals. This proportion goes beyond 90%, if $thr = 15\%$. This confirms the potential to estimate high-priority traffic demands based on historical data. We thus analyze the persistence of this stability. Specifically, we examine the mean "run-length" of the time sequence: the length of continuous 5-minute intervals where the change in traffic for individual DC pairs remains insignificant (*i.e.*, below thr). Figure 7b shows the distribution of mean run-length across individual DC pairs. We observe that only about 30% of the DC pairs remain stable for more than ten time intervals on average when $thr = 5\%$. This percentage goes up to 70% when using $thr = 10\%$. Only if we can tolerate 15% of change (*i.e.*, $thr = 15\%$), do 80% of the DC pairs remain stable for over ten time intervals. This implies the difficulty for traditional moving average prediction methods to capture the non-linear dynamic temporal patterns of high-priority inter-DC traffic.

Observation 1: While the aggregated high-priority traffic in the DC-WAN remains relatively stable, the traffic of individual DC pairs exhibits greater instability. These variations may challenge existing traffic estimation methods (used in current TE solutions).

5.2 Service-level Traffic Dynamics

As mentioned in §3, Baidu's DCN consists of various types of services, which contribute to complicated aggregate traffic patterns. We thus explore the traffic dynamics at a service level.

In our previous work [15], we discovered different diurnal and dynamic patterns of high-priority WAN traffic

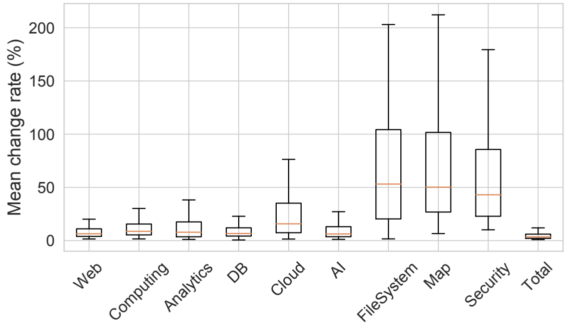
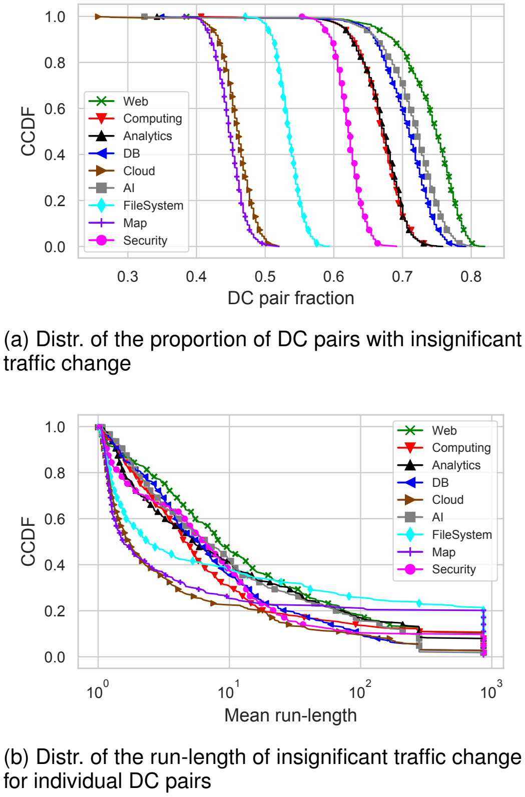


Fig. 8: The distribution of the average change rate of the high-priority traffic across different types of services.

among services. Here, we further inspect the traffic dynamics of each DC pair across different types of services. Figure 8 depicts the change rate distribution. We see dramatically different traffic dynamics across service types. Compared to Web, DB and AI services, the FileSystem, Map and Security services exhibit significantly higher dynamics in high-priority inter-DC traffic. This is possibly because of their unpredictable usage patterns and the relatively low traffic volume.



(a) Distr. of the proportion of DC pairs with insignificant traffic change

(b) Distr. of the run-length of insignificant traffic change for individual DC pairs

Fig. 9: High-priority traffic stability across different types of services with the change rate threshold of 10%.

The traffic exchanged between DCs may be dominated by a single service or a group of different services, which in turn may lead to complicated traffic patterns. We next examine how the stability of high-priority traffic exchanged between DCs varies across services. Figure 9 plots the results with a change rate threshold of 10% (*i.e.*, $thr = 10\%$). The stability indeed varies greatly across services. First, the Web, DB and AI services exhibit very high stability for most DC pairs: for 80% of 5-minute intervals, the traffic change of over 65% of DC pairs remains stable. In contrast, the Map and Cloud services exhibit less stability with the traffic of under half of DC pairs remaining stable for 80% of 5-minute intervals. Second, we see that Web services have the longest run-length of stability. 50% of the DC pairs remain stable for over ten time intervals. The run-length for the Map and Cloud services is much shorter though. Only about 20% of DC pairs remain stable for over ten time intervals.

> **Observation 2:** Different services exhibit various traffic dynamics, which lead to complicated temporal characteristics of high-priority traffic. This makes traffic prediction challenging.

5.3 Understanding Correlations among DC Pairs

The above observations imply that relying on simple moving average predictions [2], [5] may yield poor results for TE solutions, particularly as individual DC pairs often exhibit markedly different levels of stability and host a variety of services with different dynamic patterns. Thus, traditional linear models are not able to capture the high-dimensional temporal characteristics of the high-priority inter-DC traffic.

In addition to the temporal patterns, we observe that the traffic variations of some DC pairs may correlate with each other. We argue that these correlations may provide another dimension for use in traffic prediction.

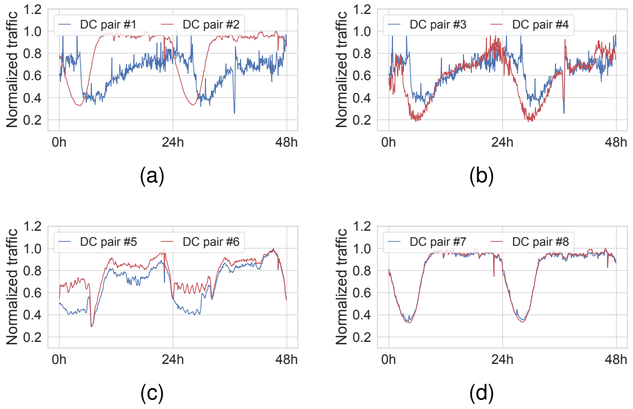


Fig. 10: Examples of high-priority traffic of DC pairs during two days; normalized to the maximum traffic volume. (a) share the same destination DC, not service related; (b) share the same source DC, mostly for Computing services; (c) belong to a bidirectional network link, mostly for Web services; (d) not topologically related, mostly for AI services.

Identifying Traffic Correlations. To better highlight this, we take four groups of representative DC pairs and plot their high-priority traffic during two days in Figure 10. Several clear trends can be observed:

(*1*) In Figure 10a, the traffic of DC pair #1 (mostly for Computing services) is less stable than DC pair #2 (mostly for AI services), yet they tend to have opposite traffic trends during some time periods. That said, the traffic of DC pair #1 and that of #2 are correlated with each other. This is because

DC pair #1 and #2 share the same destination DC, and thus compete for bandwidth, leading to this negative correlation.

(*2*) In Figure 10b, the traffic of DC pair #3 and #4 (both mostly for Computing services) tend to have consistent traffic trends over time. Because DC pair #3 and #4 share the same source DC, their traffic exhibits a positive correlation (due to the intensive traffic demands of Computing services).

(*3*) In Figure 10c, the traffic of DC pair #5 and #6 (both mostly for Web services) also tend to have consistent traffic trends over time. DC pair #5 and #6 belong to a bidirectional network link (*e.g.*, the source of DC pair #5 is the same as the destination of DC pair #6, and vice versa). The data synchronization of Web services between DCs may lead to this positive correlation.

(*4*) In Figure 10d, unlike other examples, DC pair #7 and #8 are not related in network topology, but their traffic exhibit a strong positive correlation. This is because both DC pairs primarily host AI services, and therefore have similar traffic dynamic patterns, leading to the positive correlation.

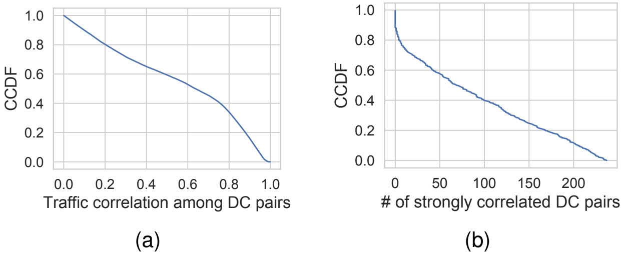
Taken together, the above examples suggest that the knowledge of traffic correlations may be useful in traffic predictions. We identify two potential reasons for these correlations: either the DC pairs are *topologically related* or they are *service related*. Two DC pairs are topologically related if their links are neighbors: (*a*) share the same destination DC; (*b*) share the same source DC; (*c*) belong to a bidirectional network link (*e.g.*, $a \to b, b \to a$) or two cascaded links (*e.g.*, $a \to b, b \to c$). In case (*a*), they may compete for bandwidth (as DC pair #1 and #2 do in Figure 10a), leading to a negative correlation. In case (*b*)-(*c*), they may have positive correlations because of popular traffic demands, data sync or cooperative tasks (as DC pair #3-#6 do in Figure 10b-10c). In contrast, two DC pairs are service related if they carry traffic for similar services. Thus, common events in these services (*e.g.*, a sudden increase in user requests) may simultaneously impact the traffic of both DC pairs (as DC pair #7 and #8 do in Figure 10d). We quantify service similarity between two DC pairs using the Cosine Similarity Coefficient between the traffic distribution of the ten types of services (see §3). Two DC pairs are considered service related if their cosine similarity is over 0.8.

Understanding Traffic Correlations. Next, we study how these two reasons impact the correlation of traffic variations among DC pairs. To this end, we compute the Pearson Correlation Coefficient, ρ, of the traffic time series between each two DC pairs as follows:

$$\rho(p,q) = \frac{cov(T_p, T_q)}{\sigma_{T_p}\sigma_{T_q}} \tag{3}$$

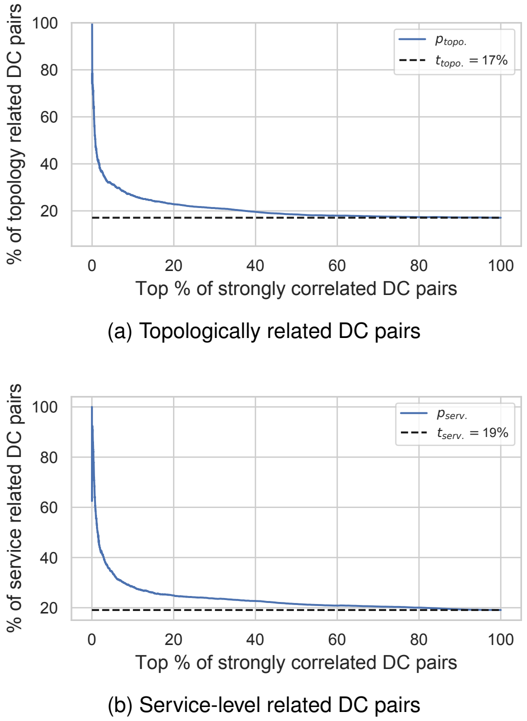
where T_p is the traffic time series of DC pair p, $cov(\cdot)$ is the covariance operator, and σ_{T_p} is the standard deviation of T_p. Note that we consider its absolute value to reflect both the positive and negative correlation: the larger $|\rho(p,q)|$ is, the stronger the correlation between T_p and T_q is.

Figure 11a depicts the distribution of the absolute correlation coefficients among all DC pairs ($|\rho|$) as a CCDF. Over 50% of traffic interaction pairs have strong correlations, with $|\rho| \geq 0.6$. For each DC pair, we count the number of strongly correlated DC pairs and show the CCDF in Figure 11b. We see that about 60% of DC pairs are strongly correlated with at least 50 other DC pairs. This confirms prevalent correlations in traffic variations between DC pairs in DC-WAN.



(a) (b)

Fig. 11: Traffic correlation among DC pairs. (a) Distr. of correlation coefficients among DC pairs; (b) Distr. of the number of strongly correlated DC pairs for each DC pair.



(a) Topologically related DC pairs

(b) Service-level related DC pairs

Fig. 12: Impact of topological and service-level relationship on the inter-DC traffic correlations among DC pairs.

To understand the importance of topology *vs.* service relations, we next sort all the DC pair tuples <DC pair i, DC pair j > ($i \neq j$) in descending order according to their absolute Pearson correlation coefficient, $|\rho|$. This ranks the pairs based on the similarity in their traffic dynamics. Then, for the top $x\%$ strongly-correlated DC pairs, we compute the proportion of pairs that are topology-related (denoted by $p_{topo.}$) and the proportion that are service-related (denoted by $p_{serv.}$).

Figure 12 reports the results. The plots also contain a baseline for comparison: Figure 12a contains $t_{topo.}$, which is the fraction of the topology-related DC pairs out of all tuples (which is 17%); Figure 12b contains the $t_{serv.}$ baseline, which is the fraction of service-related DC pairs out of all tuples (which is 19%). A larger value of $p_{topo.}$ *vs.* the baseline, $t_{topo.}$, indicates it is impacted by a topological relationship. Similarly, a larger value of $p_{serv.}$ *vs.* the baseline, $t_{serv.}$ indicates it is impacted by a service-level similarity. We see from Figure 12 that for the top 1% of strongly-correlated DC pairs, about 47% are topology-related. Furthermore, about 56% of them are service-related, confirming the large combined impact of these two factors.

> **Observation 3:** Some DC pairs exhibit strong correlations with other DC pairs in terms of traffic dynamics. This is driven by either topological or service-level relationships between pairs. We posit that this correlation provides another dimension to improve traffic demand estimation.

6 Inter-Datacenter traffic Prediction

In this section, based on the Observations 1-3, we propose a learning-based model for performing inter-DC traffic capacity prediction: the *Interrelated-Temporal Graph Convolutional Network* (IntegNet). We start by formulating the problem and giving an overview of the IntegNet model. We then describe each of the IntegNet modules in detail.

6.1 Problem Statements

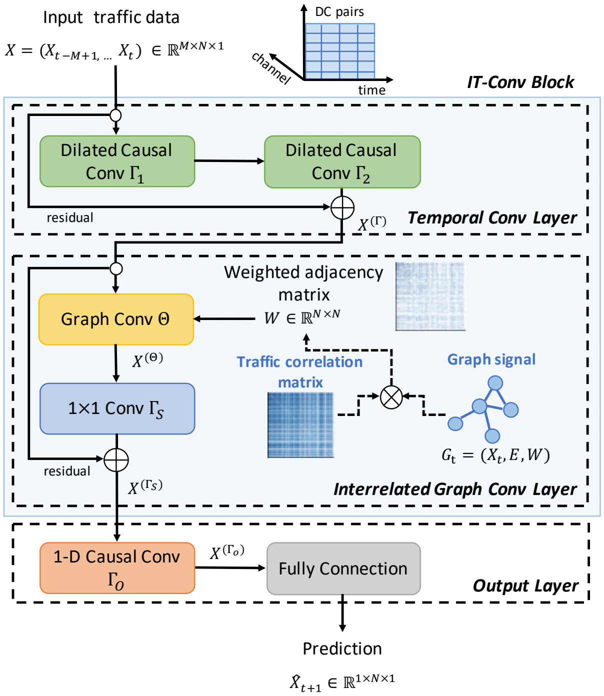
In our network model, we denote by $X_t = (x_t^1, \ldots, x_t^N)$ the high-priority traffic data interacted between DCs in time interval t, where x_t^i is the traffic demand of DC pair $i \in \mathcal{N}$ at time t ($\mathcal{N}$ is the set of N DC pairs in the DC-WAN). We then denote the capacity predicted for DC pair i at time t as $\hat{x}_t^i$, and $\hat{X}_t$ as the set of forecast capacities of all DC pairs. Therefore, the inter-DC traffic capacity prediction problem is to compute $\hat{X}_{t+1}$ based on knowledge of the historical traffic data $(X_{t-M+1}, \ldots, X_t)$ of previous M time intervals.

From Observation 1 and 2, we see complicated temporal characteristics for high-priority inter-DC traffic, as well as a wide range of traffic variations for individual DC pairs. Existing statistical models (*e.g.*, MA, EWMA, ARIMA and SARIMA) are not able to accurately capture such large variations of non-linear temporal features. Therefore, we next utilize neural network models to learn the high-dimensional temporal patterns from historical inter-DC traffic data. Moreover, Observation 3 shows that some DC pairs exhibit stronger traffic correlations with others, driven by either topology- or service-level relationships between DC pairs. Intuitively, the network traffic between DCs can be organized as a graph structure based on these relationships. Accordingly, in our model, we employ a graph convolution on the graph-structured traffic data to directly extract these correlation features (which are overlooked in previous studies).

To summarize, IntegNet utilizes a learning-based neural network model with a customized loss function to capture the combination of temporal features and traffic correlation patterns to predict high-priority inter-DC traffic demands. IntegNet aims to forecast the capacity required to satisfy the high-priority service demands while saving the remaining resource for low-priority bulk transfers, *i.e.*, balancing the QoS losses and resource overprovisioning. Next, we present an overview of IntegNet, its inputs/outputs as well as the loss function we propose.

6.2 Model Overview

IntegNet Structure. Figure 13 presents the overall architecture of IntegNet. It is composed of a Temporal Convolution layer, an Interrelated Graph Convolution layer, and an Output layer. The Temporal Convolution (TCN) layer adopts the dilated causal convolution module for modeling the temporal dependencies of inter-DC traffic. In the Interrelated Graph Convolution (GCN) layer, a graph convolution module followed by a 1×1 convolution module is employed for modeling the traffic correlations among DC pairs.



Input traffic data
$X = (X_{t-M+1, \ldots} X_t) \in \mathbb{R}^{M\times N\times 1}$
DC pairs
channel
time
IT-Conv Block
Dilated Causal Conv Γ_1
Dilated Causal Conv Γ_2
residual
$X^{(\Gamma)}$
Temporal Conv Layer
Weighted adjacency matrix
$W \in \mathbb{R}^{N\times N}$
Graph Conv Θ
$X^{(\Theta)}$
Traffic correlation matrix
Graph signal
1×1 Conv Γ_S
$G_t = (X_t, E, W)$
residual
$X^{(\Gamma_S)}$
Interrelated Graph Conv Layer
1-D Causal Conv Γ_O
$X^{(\Gamma_O)}$
Fully Connection
Output Layer
Prediction
$\hat{X}_{t+1} \in \mathbb{R}^{1\times N\times 1}$

Fig. 13: The architecture of IntegNet network.

First, in the TCN layer, we employ two-layer stacked dilated causal convolution structures [42] on time dimension data to capture the temporal dependencies of the traffic among DC pairs. This module allows for exponentially large receptive fields with increasing layer depth in order to properly handle long-range sequences in a non-recursive manner. Second, in the GCN layer, we adopt the spectral-based graph convolution [43] to learn the correlation patterns of the output temporal features of the traffic of each DC pair (output from the TCN layer). The 1×1 convolution module is used here for adding non-linearity. We further apply residual connections in both the TCN and GCN layers to preserve gradient validity and avoid model degradation. Finally, in the Output layer, we use a 1-D causal convolution module, followed by a fully connection (FC) layer, to integrate the features and produce the prediction. In this way, we obtain the predicted inter-DC traffic demands and use the prediction errors to update the model parameters in model training procedure.

Data Inputs/Outputs. We take the high-priority traffic data of all N DC pairs of the previous M time intervals as the input, which is denoted as $X = (X_{t-M+1}, \ldots, X_t) \in \mathbb{R}^{M\times N\times 1}$; the last dimension stands for the channel feature as shown in Figure 13. To jointly capture the temporal and correlation features of inter-DC traffic, we abstract the input X as the graph-structured data. Since the traffic data of each DC pair x is not independent but interrelated with those of other DC pairs with high traffic correlations, X_t can be regarded as a graph signal at the t-th time interval. We define an undirected graph denoted as $G_t = (V_t, E, W)$, where V_t is a finite set of vertices, corresponding to the traffic volume X_t of N DC pairs, *i.e.*, each node in G_t corresponds to a DC pair x's traffic volume at time t. E is a set of edges, indicating there exists a topological or service-level similarity relationships between the DC pairs with a strong traffic correlation (*i.e.*, $|\rho| \geq 0.6$); and $W \in \mathbb{R}^{N\times N}$ denotes the weighted adjacency matrix of graph G_t, in which the weight is measured by the absolute value of traffic correlation coefficient ρ and can be formed as:

$$w_{ij} = \begin{cases} |\rho_{ij}| & , i \neq j \quad and \quad e_{ij} \in E \\ 0 & , otherwise \end{cases} \tag{4}$$

To help understand this, Figure 14 shows the heat map of weighted adjacency matrix W. When forecasting the traffic of one inter-DC connection, the adjacent nodes in G_t with a high link weight (*i.e.*, the traffic data of other DC pairs with strong correlations) can be used to assist in the prediction.

Based on this graph-structured traffic data, the Temporal Convolution layer first captures the temporal features for each node in G. Then, at each time interval t in M, the Interrelated Graph Convolution layer learns the correlation patterns of the output temporal features for all the nodes in G_t, with weight w_{ij}. Overall, based on the learned temporal dependencies and traffic correlations, IntegNet generates the final single-step capacity prediction $\hat{X}_{t+1} \in \mathbb{R}^{1\times N\times 1}$.

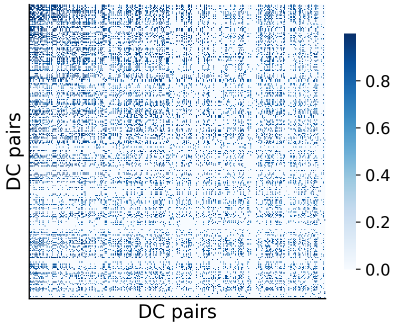


Fig. 14: Heat map of weighted adjacency matrix.

Loss Function. Generally, learning-based prediction solutions evaluate the quality of output (*i.e.*, the prediction accuracy) by using standard loss functions, such as Mean Absolute Error (MAE) or Mean Square Error (MSE). However, these metrics are not well suited for capacity forecasting (see §3). These metrics impose an equal penalty on both the positive loss (*i.e.*, overestimation) and the negative loss (*i.e.*, underestimation). They are therefore not able to achieve a cost balance of QoS losses of high-priority services vs. resource overprovisioning. Therefore, we propose a Weighted Mean Square Error loss function to implement asymmetric cost for overestimation and underestimation (see details in §6.6).

6.3 Temporal Convolution Layer

We next describe each of the IntegNet modules in more detail. First, we introduce the Temporal Convolution network (TCN) layer used for capturing temporal dynamic patterns of inter-DC traffic. We rely on dilated causal convolution structures [44] as the TCN layer to capture the temporal dependencies for DC pairs. As shown in Figure 15, the dilated causal convolution operation slides over inputs by skipping values with a certain step, and applies the standard 1-D causal convolution to the selected values. The receptive field of the convolution can grow exponentially by stacking dilated causal convolution layers with dilation factors in an increasing order. Thus, dilated causal convolution networks are able to capture long-range sequences with fewer layers in a non-recursive manner, which can be parallelized during training and save computation resources.

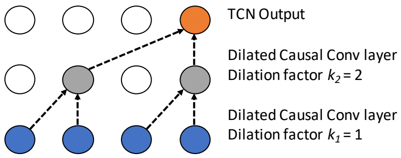

TCN Output
Dilated Causal Conv layer
Dilation factor k_2 = 2
Dilated Causal Conv layer
Dilation factor k_1 = 1

Fig. 15: Visualization of the stacked dilated causal convolution layers with two-layer depth and kernel size 2.

In the TCN layer, for each DC pair (*i.e.*, each node in G), the input is a length-M observed traffic sequence with one channel as $x \in \mathbb{R}^{M\times 1\times 1}$. The dilated causal convolution (defined with the convolution kernel Γ) interactively explores K_t neighbors spaced with a certain distance (determined by the dilation factor k) over x to calculate the temporal features of traffic volume. To generalize the dilated causal convolution to the input traffic data $X \in \mathbb{R}^{M\times N\times 1}$ of all DC pairs (*i.e.*, all nodes in G), we employ the convolution kernel Γ to each node in G equally to capture their dynamic temporal features, and obtain the output as $X^{\Gamma} \in \mathbb{R}^{M\times N\times C_{\Gamma}}$. C_{Γ} is the size of output channels, which is determined by the number of output filters in the dilated causal convolution.

We stack two layers of dilated causal convolutions (defined with the convolution kernel Γ_1, Γ_2 respectively in Figure 15) with the kernel size $K_t = 2$ and the dilation factor $k_1 = 1, k_2 = 2$ (see Figure 13). The size of output channels is designed as $2C_{\Gamma_1} = C_{\Gamma_2} = 64$. We apply residual connections and utilize ReLU as the activation function.

6.4 Interrelated Graph Convolution Layer

We next proceed to model the traffic correlation patterns among DC pairs in the Interrelated Graph Convolution layer. We rely on the spectral-based graph convolution structures in the GCN layer to capture the correlation dependencies among DC pairs. Graph convolution can be used on graph-structured data to extract the highly meaningful features of nodes by aggregating and transforming their neighbor information. At the time interval t, the input of the GCN layer is denoted as $X_t^{\Gamma} \in \mathbb{R}^{1\times N\times C_{\Gamma}}$. We employ the spectral-based graph convolution on X_t^{Γ} to extract its correlation features for all the nodes in G_t. Specifically, the graph convolution can be defined with the convolution kernel Θ as follows:

$$\Theta *_{\mathcal{G}} X_t^{(\Gamma)} = \Theta(L) X_t^{(\Gamma)} \in \mathbb{R}^{1\times N\times C_\Theta} \quad (5)$$

where "$*_{\mathcal{G}}$" is the graph convolution operator, L is the graph Laplacian with $L = D - W$, D is the diagonal matrix with $D_{ii} = \sum_j W_{ij}$, W is the weighted adjacency matrix of graph G, previously defined in the *data inputs*. C_Θ is the size of output channels, which is determined by the number of output filters in the graph convolution. To reduce the number of parameters and lower the computing complexity, we utilize Chebyshev polynomial approximation [43] to restrict the convolution kernel with the kernel size of K_s. That is, the graph convolution recursively computes local convolutions within the radius of K_s from each node in G_t through polynomial approximation.

To generalize the above graph convolution to X^Γ, at each time interval in M, we impose the equal graph convolution operation with the same kernel Θ on all the nodes in G_t to capture their topologically and service-level interrelated features, and obtain the output as $X^\Theta \in \mathbb{R}^{M\times N\times C_\Theta}$. Following the graph convolution, we employ a 1×1 convolution (defined with the convolution kernel Γ_s) to add non-linearity, and obtain the output as $X^{\Gamma_s} \in \mathbb{R}^{M\times N\times C_\Theta}$. In our model, the graph convolution is designed with the kernel size $K_s = 3$ and the output channel size $C_\Theta = 64$. We apply residual connections and utilize ReLU as the activation function.

As shown in Figure 13, by combining the Temporal Convolution layer and Interrelated Graph Convolution layer, the IntegNet can extract both temporal dependencies and traffic correlation patterns for predicting high-priority inter-DC traffic.

6.5 Output Layer

In the output layer, we first utilize the 1-D convolution Γ_o with kernel size of M to map the outputs of Interrelated Graph Convolution layer X^{Γ_s} to a single-step prediction, which is denoted as $X^{\Gamma_o} \in \mathbb{R}^{1\times N\times C_\Theta}$. Here we utilize a Sigmoid as the activation function. Then we use a Fully Connected (FC) layer to map X^{Γ_o} from multi-channels to one-channel. In the FC layer, a linear transformation is applied across C_Θ-channels for the input $X^{(\Gamma_o)}$ as $X^{\Gamma_o} w_f + b_f$, where w_f is a weight vector and b_f is a bias. Finally, we obtain the predicted inter-DC traffic $\hat{X}_{t+1} \in \mathbb{R}^{1\times N\times 1}$ in the $(t + 1)$-th time interval.

6.6 Loss Function

The loss function is an essential part of model training, which determines the penalty incurred when making a prediction error. Inspired by previous work [29], [38], we propose to use a Weighted Mean Square Error loss function, denoted by $l(\cdot)$, to account for the asymmetric cost for: (*1*) overestimation, *i.e.*, forecasting a higher value than the actual load, which leads to resource overprovisioning; and (*2*) underestimation, *i.e.*, predicting a lower value than the actual load, which leads to the QoS loss. Recall that the aim of capacity forecasting in DC-WAN traffic engineering is to save resources for low-priority traffic, while ensuring the QoS of high-priority services. We give higher weights to the cost of underestimation than that of overestimation in the cost model. For each DC Pair, we denote by $\hat{x}_{t+1}$ the predicted traffic demand in the $(t + 1)$-th time interval, and by x_{t+1} the corresponding actual traffic load. The forecast error is then expressed as $s = \hat{x}_{t+1} - x_{t+1}$. Thus, the cost model of capacity forecast is defined as follows:

$$c(s) = \begin{cases} \alpha \cdot s^2 & , s < 0 \\ s^2 & , s \geq 0 \end{cases} \quad (6)$$

where the constant α ($\alpha > 1$) represents the product factor of weight for the higher penalty of underestimation. The loss function used to evaluate the quality of capacity forecast in the $(t + 1)$-th time interval for all the N DC pairs is then:

$$l(\hat{X}_{t+1} - X_{t+1}) = \frac{1}{N} \sum_{x \in X} c(\hat{x}_{t+1} - x_{t+1}) \quad (7)$$

The setting of constant weight α can be obtained from the trade-off between the traffic overprovisioning and unserviced traffic demands. Our current design sets $\alpha = 50$ to balance the resource overprovisioning and QoS loss (see detailed evaluation in §7).

6.7 Practical Concerns

To finish, we briefly discuss some practical considerations that our architecture faces.

Model Extension. We note that the TCN and GCN layers can be combined into the Interrelated-Temporal convolutional (IT-Conv) block, and stacked based on the scale and complexity of traffic data. While more convolution layers may learn higher-order features, more parameters need to be trained (costing more training time), and overfitting may occur. Our current design uses one TCN layer and one GCN layer to balance the performance with training complexity (see detailed evaluation in §7).

Integration with DC-WAN TE. The IntegNet model is trained offline. Given that there are often hundreds of DC pairs communicating, the traffic graph (in which each node corresponds to a DC pair's traffic) is of a relatively small scale (compared to other machine learning tasks). As we show in §7, a round of training lasts just a few minutes on a mid-range server. It can therefore be retrained regularly (*e.g.*, per day if the traffic patterns vary in a time scale of 1 day). The trained model can also be integrated into existing DC-WAN traffic engineering systems (*e.g.*, SWAN, B4) for online prediction of the high-priority DC-WAN traffic demand. Note that the average inference time of all DC pairs at each time step is only 10^{-3} seconds (see §7), indicating that it is feasible for use in online prediction.

7 Evaluation

7.1 Experimental Setup

Data Training. We evaluate IntegNet using our Baidu dataset from §4. Given that the time interval in our dataset is set to 5 minutes, there are 288 data points per day for each DC pair. In our experiments, we set the historical time intervals for input as $M = 12$; that said, we use the prior 12 observed inter-DC traffic matrices (*i.e.,* the prior 60 minutes) to make the one-step prediction for the next 5 minutes.

Specifically, we iteratively split the data for each day into a series of data segments with a stride of one time interval, where each segment consists of 12 traffic matrices used as the input and the following one used for measuring the prediction errors. The training is run on the first 14 days of Netflow data. Therefore, we obtain 3,864 data segments for model batch training. The data of the next 2 days (*i.e.,* 552 data segments) are used for validation, in order to locate the best parameter settings for the trained model. Finally, we use the data of the last 2 days (*i.e.,* 552 data segments) for testing. Note that the traffic data is pre-processed with log-transform and normalized with the Z-Score method. Unless specified otherwise, we train the prediction model once and do not retrain it during the evaluation.

Evaluation Metrics. The cost of resource overprovisioning and QoS loss is determined by the deviation between forecast traffic capacity and the actual traffic demand. We thus leverage the absolute estimation error of traffic capacity to evaluate the forecasting performance. We use $y(i,t)$ and $\hat{y}(i,t)$ to denote the actual and predicted traffic demand of DC pair i at time interval t. For all the N DC pairs over T time intervals, we define the metric of overestimation ($Overcost$) and underestimation ($Undercost$) as follows:

$$\begin{aligned} Undercost &= \frac{1}{T} \times \sum_{t=1}^{T}\sum_{i=1}^{N} |y(i,t) - \hat{y}(i,t)| \\ &\quad \text{if} \quad \hat{y}(i,t) - y(i,t) < 0 \\ Overcost &= \frac{1}{T} \times \sum_{t=1}^{T}\sum_{i=1}^{N} |y(i,t) - \hat{y}(i,t)| \\ &\quad \text{if} \quad \hat{y}(i,t) - y(i,t) \geq 0 \end{aligned} \tag{8}$$

Baselines. We compare our model against four baselines: Moving Average (MA), Exponentially Weighted Moving Average (EWMA), a Long Short-Term Memory (LSTM) model [45], and a Graph Attention (GAT) model [46]. MA and EWMA are the most widely used DC-WAN traffic estimation methods [2], [5], [7], and are performed as univariate time series prediction. LSTM is a popular RNN-based neural network model for time series prediction, often used for network traffic [23], [30]. The GAT model adopts a convolutional architecture with an attention mechanism to learn relative weights between neighboring nodes in graph structured data, which is leveraged by a recent solution for network congestion predictions [47]. We apply these baseline models in capacity forecast problem by the *naive* solution of setting aside "headroom" that is dependent on the historical prediction error. Specifically, we inflate the predicted traffic demand based on the error in past estimations (mean plus 1.96 standard deviations) to guarantee the provision for high-priority traffic demands.

Parameter Settings. The settings of our IntegNet model are presented in §6. In the LSTM model, we deploy an LSTM layer by setting the hidden size as 128 and using ReLU as the activation function. In the GAT model, we deploy an attention layer by using 8 attention heads (8 output features per head) with the Exponential Linear Unit (ELU) activation function, and utilize a linear transformation in the output layer. We choose the optimal settings for individual models according to the validation results. We will discuss the impact of model complexity on the effectiveness of IntegNet in §7.7.

We use mean square error (MSE) as the loss function for training LSTM and GAT models. The optimizer used in LSTM and GAT models are *RMSprop* and *Adam* respectively. We train our IntegNet model by minimizing the Weighted Mean Square Error ($l(\cdot)$) using *RMSprop* as the optimizer. LSTM, GAT, and our IntegNet models are all implemented in *Tensorflow* and trained for 100 epochs with a batch size of 32. All experiments are performed on a mid-range Linux server (Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz, GeForece RTX 2070 GPU).

7.2 Comparison Against Predictors Using Headroom

Overall Evaluation. Figure 16 reports both the *Overcost* (*i.e.,* overprovisioning traffic volume) and *Undercost* (*i.e.,* unserviced traffic volume) of the high-priority traffic prediction over all DC pairs. The traffic volume is normalized to the maximum value of the sum of inter-DC traffic. Apart from the four baselines, we also compare a *naive* version of IntegNet, denoted by *IntegNet_h*, which makes capacity prediction by minimizing the MSE by setting aside "headroom".

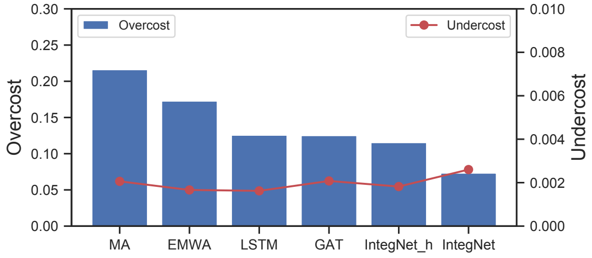


Fig. 16: Comparative evaluation of IntegNet with baseline methods.

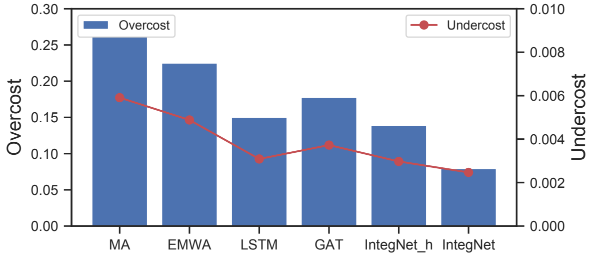
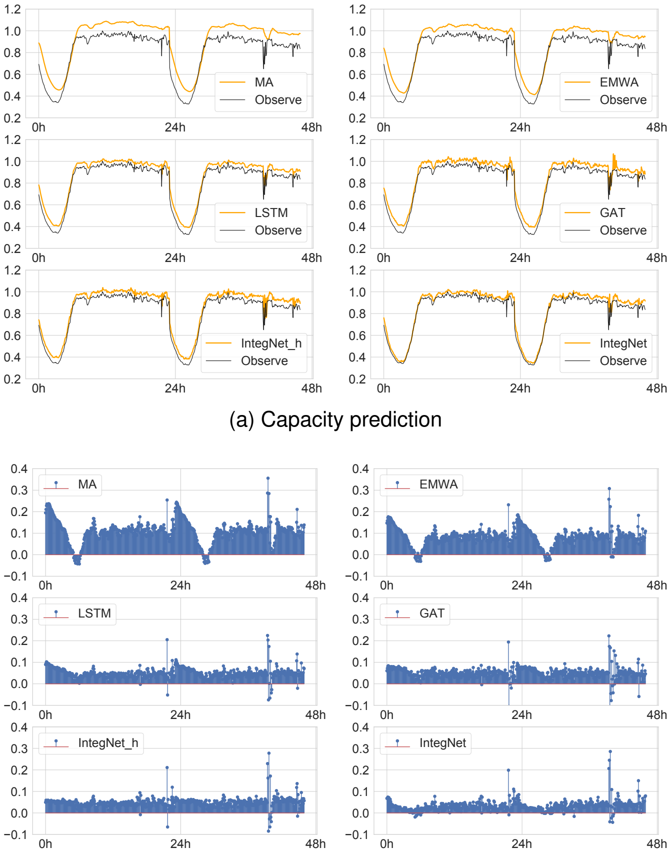


Fig. 17: Performance of IntegNet for DC pairs with high traffic dynamics.

We see that our proposed IntegNet model achieves the lowest overestimation (normalized as 0.073). This constitutes a 66.2%, 57.6%, 42.1%, 41.6%, and 36.5% overprovisioning reduction compared with MA, EWMA, LSTM, GAT, and IntegNet_h respectively. The *Undercost* of IntegNet is less than 0.3% of the total inter-DC traffic volume, which can be tolerated with limited QoS loss. These results confirm the effectiveness of leveraging the Weighted Mean Square Error



(a) Capacity prediction

(b) Estimation error

Fig. 18: Visualization of prediction performance of IntegNet and other comparison methods: (a) capacity prediction results and (b) estimation errors of an example DC pair over two days. The y-axis shows the traffic volume is normalized by the maximum traffic volume.

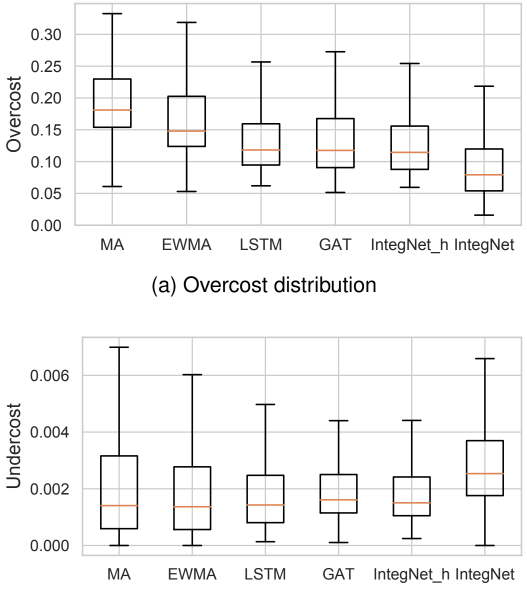
loss function in balancing between resource overprovisioning and QoS losses of high-priority traffic.

Besides, IntegNet_h achieves the lowest *Overcost* as well as quite low *Undercost* compared to the four baselines. This result confirms that incorporating both temporal dynamic patterns and their correlations among DC pairs is important for inter-DC traffic estimation.

Evaluation for Dynamic DC Pairs. To understand how IntegNet performs for DC pairs with high dynamics, we next inspect the estimation cost for DC pairs whose traffic has a change rate of over 10% (see Figure 7a). Note, these are the hardest cases for traffic prediction. Figure 17 plots the normalized estimation cost for these pairs. As expected, the traffic is more difficult to predict, evidenced by the higher overestimation and underestimation than those for all the DC pairs (in Figure 16). Nevertheless, our method IntegNet outperforms the others with both the lowest *Overcost* and *Undercost*, showing its robustness even for the hardest cases.

Figure 18 plots the performance of our IntegNet model against other methods. We present the capacity prediction results compared to the observed ones, showing that IntegNet has the best performance. The figure also plots the estimation cost over time to show that IntegNet outperforms the others with significantly low overprovisioning and unserviced traffic demands for most time intervals.

Evaluation for Individual DC Pairs. We next analyze the traffic prediction performance for all the *individual* DC pairs in Figure 19. For each solution, we present the distribution of the *Overcost* and *Undercost* for individual DC pairs. We note that while the performance of all solutions varies across DC pairs, our model IntegNet shows a significantly lower *Overcost* than others. Further, although IntegNet shows the largest median underestimation, the normalized median *Undercost* is still less than 0.3%, which can be tolerated with limited QoS loss in practice.



(a) Overcost distribution

(b) Undercost distribution

Fig. 19: Distribution of prediction performance over individual DC pairs.

7.3 Loss Function Trade-off Analysis

As aforementioned in §6, the weighted parameter, α, in our proposed loss function can be determined based on the trade-off between overestimation and underestimation. We next evaluate the performance of IntegNet with different settings of α in loss function, $l(\cdot)$, and present the results in Figure 20. We see that, as expected, a higher α reduces unserviced traffic demand (*i.e., Undercost*) at the cost of provisioning more capacity (*i.e., Overcost*). Our current choice of $\alpha = 50$ can achieve a balance between the resource overprovisioning and QoS losses.

7.4 Benefits of GCN

One of the key components of our IntegNet model is the Interrelated Graph Convolution (GCN) layer. As a complement to the Temporal Convolution (TCN) layer, it learns the traffic correlation patterns among DC pairs and helps to improve the prediction accuracy. We further validate the effectiveness of GCN with ablation analysis and show the results in Figure 21.

We compare IntegNet with TCN alone, which denotes the model that only consists of the TCN layer. We also compare IntegNet_h with TCN_h, which denotes the *naive* version of TCN by setting aside "headroom" for capacity

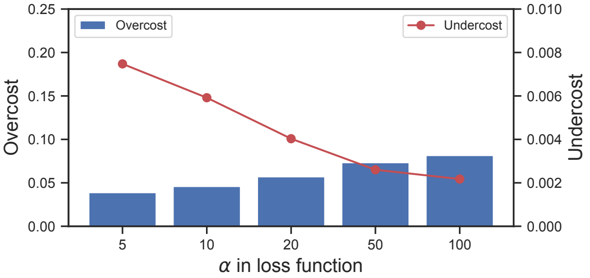


Fig. 20: Performance trade-off of IntegNet with different α in loss function.

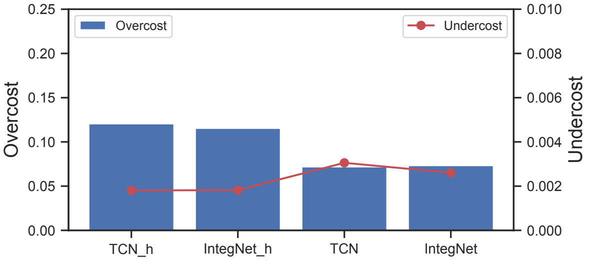


Fig. 21: Benefits of GCN on the performance of IntegNet.

forecast. We find that IntegNet reduces 16% of *Undercost* while keeping nearly the same low *Overcost* compared to TCN. Besides, IntegNet_h achieves lower *Overcost* than TCN_h, which implies the smaller prediction error of IntegNet_h. These results confirms that the GCN layer is of great importance for capturing high-dimensional traffic correlation patterns to improve the forecast performance.

7.5 Effect of Interrelated Features

Our IntegNet model captures the correlation features of inter-DC traffic based on the graph-structured traffic data, which is determined by interrelated features we observed in measurement (see §5). We utilize both topologically related and service-level related features in our design. In Table 1, we evaluate how these two types of features affect learning traffic correlation patterns by using either topologically related features (denoted by IntegNet_topology) or service-level related features (denoted by IntegNet_service). We see that the IntegNet model achieves the best performance, fully characterizing the correlation features. Compared to IntegNet_topology, IntegNet_service achieves lower *Undercost* at the cost of higher *Overcost*.

TABLE 1: Effect of topology and service correlations on the performance of IntegNet.

Method	Overcost	Undercost
IntegNet	**0.0733**	**0.0026**
IntegNet_topology	0.0719	0.0030
IntegNet_service	0.0779	0.0027

7.6 Evaluation of Computational Overhead

We next investigate the time required for model training and inference, as listed in Table 2. Note that we run all the methods on a mid-range Linux server (Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz, GeForece RTX 2070 GPU). We evaluate the training time of running one epoch on the training data lasting for 14 days. Note that MA and EWMA have no model training process. The average training time for IntegNet is 3.12 seconds for a round of training (in comparison to LSTM and GAT, which takes 31.77 and 49.59 seconds respectively). This means that the model can easily be retrained regularly if needed.

TABLE 2: Computation time overhead evaluation.

Method	Training time (s/epoch)	Inference time (s/time step)
MA	—	3.8×10^{-5}
EWMA	—	5.4×10^{-1}
LSTM	31.77	4.6×10^{-3}
GAT	49.59	5.9×10^{-3}
IntegNet	**3.12**	$\mathbf{1.1 \times 10^{-3}}$

We also compute the average inference time for all DC pairs at each time step on the testing data, lasting for 2 days. This overhead will be encountered whenever a prediction is required by the traffic engineering algorithm. As expected, MA has the lowest execution time (*i.e.*, 10^{-5} seconds in our setup) for each prediction. LSTM, GAT and our proposal IntegNet requires more time for inference: the average time to get a prediction of traffic demand is about 10^{-3} seconds in our setup. As traffic engineering is often scheduled on a 5-minute time scale [2], the inference time for these methods are reasonable in practice.

7.7 Impact of Model Complexity

As aforementioned in §6, in IntegNet, the TCN and GCN layer can be combined into the IT-Conv block and stacked by multiple layers to obtain deeper models. While deep convolution layers may capture higher-dimension features, they may lead to overfitting and longer training time. We evaluate the prediction performance and time efficiency of IntegNet at different complexity levels of model settings in Table 3. We denote C_l $(l = 1, 2, 3 \dots)$ as the size of output channels of GCN layer in the l-th IT-Conv block stacked in IntegNet model.

TABLE 3: Impact of model complexity on the performance and time efficiency of IntegNet.

Model setting	Overcost	Undercost	Training (s/epoch)	Inference (s/time step)
$C_1 = 16$	0.0856	0.00365	1.32	6.3×10^{-4}
$C_1 = 32$	0.0719	0.00377	1.88	8.9×10^{-4}
$\mathbf{C_1 = 64}$	**0.0733**	**0.00260**	**3.12**	$\mathbf{1.1 \times 10^{-3}}$
$C_1 = 16$ $C_2 = 32$	0.0716	0.00407	2.61	1.3×10^{-3}
$C_1 = 32$ $C_2 = 64$	0.0656	0.00340	4.25	1.7×10^{-3}
$C_1 = 16$ $C_2 = 32$ $C_3 = 64$	0.0758	0.00293	5.01	2.2×10^{-3}

We see increasing time consumption with deeper IT-Conv blocks and growing GCN output channel size. In our current design, we deploy one IT-Conv block with $C_1 = 64$, which attains a good balance between the performance and time complexity.

8 Conclusion

In this paper, we have explored the characteristics of inter-DC traffic. Using 18 days of data from Baidu's DC-WAN, we have shown that the stability of high-priority traffic varies greatly across DC pairs. The various dynamic traffic patterns of the different services lead to these complicated temporal characteristics. We have also shown that the traffic variations of some DC pairs are highly correlated, and explored how these correlations are driven by topological and service-level similarities.

With these insights, we have devised a novel prediction model (IntegNet) that relies on both temporal and relational information to forecast the capacity needed by high-priority inter-DC traffic. We have also proposed a customized loss function to account for the cost balance between resource overprovisioning and QoS losses desired in DC-WAN traffic engineering. Our evaluation shows that IntegNet significantly outperforms other state-of-the-art methods.

We note that our findings are potentially specific to the Baidu DC-WAN. As one of the world's largest DC-WAN operators, we argue that this still offers powerful insights into global patterns. We have made the implementation of IntegNet publicly available at [48]. In our future work, we wish to test IntegNet in other environments, particularly smaller-scale DC-WANs to understand how our observations generalize.

Acknowledgment

This work was partially supported by Natural Science Foundation of China (U20A20180, 62072437, 62002344), Beijing Natural Science Foundation (JQ20024), and CAS-Austria Joint Project (171111KYSB20200001).

References

[1] P. Pang, Q. Chen, D. Zeng, and M. Guo, "Adaptive preference-aware co-location for improving resource utilization of power constrained datacenters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 441–456, 2020.

[2] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven wan," in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, 2013, pp. 15–26.

[3] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu *et al.*, "B4: Experience with a globally-deployed software defined wan," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013.

[4] A. Kumar, S. Jain, U. Naik, A. Raghuraman, N. Kasinadhuni, E. C. Zermeno, C. S. Gunn, J. Ai, B. Carlin, M. Amarandei-Stavila *et al.*, "Bwe: Flexible, hierarchical bandwidth allocation for wan distributed computing," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 1–14.

[5] S. Kandula, I. Menache, R. Schwartz, and S. R. Babbula, "Calendaring for wide area networks," in *Proceedings of the 2014 ACM conference on SIGCOMM*, 2014, pp. 515–526.

[6] L. Chen, S. Liu, and B. Li, "Optimizing network transfers for data analytic jobs across geo-distributed datacenters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 2, pp. 403–414, 2021.

[7] Y. Zhang, X. Nie, J. Jiang, W. Wang, K. Xu, Y. Zhao, M. J. Reed, K. Chen, H. Wang, and G. Yao, "Bds+: An inter-datacenter data replication system with dynamic bandwidth separation," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 918–934, 2021.

[8] S.-H. Tseng, S. Agarwal, R. Agarwal, H. Ballani, and A. Tang, "CodedBulk: Inter-datacenter bulk transfers using network coding," in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, 2021, pp. 15–28.

[9] Z. Yang, Y. Cui, X. Wang, Y. Liu, M. Li, S. Xiao, and C. Li, "Cost-efficient scheduling of bulk transfers in inter-datacenter wans," *IEEE/ACM Transactions on Networking*, vol. 27, no. 5, pp. 1973–1986, 2019.

[10] H. Zhang, K. Chen, W. Bai, D. Han, C. Tian, H. Wang, H. Guan, and M. Zhang, "Guaranteeing deadlines for inter-data center transfers," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 579–595, 2017.

[11] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," in *Proceedings of the ACM SIGCOMM 2011 conference*, 2011, pp. 74–85.

[12] V. Jalaparti, I. Bliznets, S. Kandula, B. Lucier, and I. Menache, "Dynamic pricing and traffic engineering for timely inter-datacenter transfers," in *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016, pp. 73–86.

[13] Y. Li, H. Liu, W. Yang, D. Hu, X. Wang, and W. Xu, "Predicting inter-data-center network traffic using elephant flow and sublink information," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 782–792, 2016.

[14] A. D'Alconzo, I. Drago, A. Morichetta, M. Mellia, and P. Casas, "A survey on big data for network traffic monitoring and analysis," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 800–813, 2019.

[15] Z. Wang, Z. Li, G. Liu, Y. Chen, Q. Wu, and G. Cheng, "Examination of wan traffic characteristics in a large-scale data center network," in *Proceedings of the 21st ACM Internet Measurement Conference*, 2021, pp. 1–14.

[16] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: a scalable and flexible data center network," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, 2009, pp. 51–62.

[17] Y. Chen, S. Jain, V. K. Adhikari, Z.-L. Zhang, and K. Xu, "A first look at inter-data center traffic characteristics via yahoo! datasets," in *2011 Proceedings IEEE INFOCOM*. IEEE, 2011, pp. 1620–1628.

[18] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 123–137.

[19] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010, pp. 267–280.

[20] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: measurements & analysis," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*, 2009, pp. 202–208.

[21] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 92–99, 2010.

[22] W. Yoo and A. Sim, "Network bandwidth utilization forecast model on high bandwidth networks," in *2015 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2015, pp. 494–498.

[23] A. Lazaris and V. K. Prasanna, "An lstm framework for modeling network traffic," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 19–24.

[24] A. Azzouni and G. Pujolle, "Neutm: A neural network-based framework for traffic matrix prediction in sdn," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–5.

[25] S. Troia, R. Alvizu, Y. Zhou, G. Maier, and A. Pattavina, "Deep learning-based traffic prediction for network optimization," in *2018 20th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2018, pp. 1–4.

[26] Q. Zhuo, Q. Li, H. Yan, and Y. Qi, "Long short-term memory neural network for network traffic prediction," in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*. IEEE, 2017, pp. 1–6.

[27] J. Lee, S. Lee, J. Lee, S. D. Sathyanarayana, H. Lim, J. Lee, X. Zhu, S. Ramakrishnan, D. Grunwald, K. Lee *et al.*, "Perceive: deep learning-based cellular uplink prediction using real-time scheduling patterns," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, 2020, pp. 377–390.

[28] J.-y. Tian, J. Qin, L.-M. Chen, H. Fang, and Z.-m. Wang, "A novel method for network traffic prediction using residual mogrifier gru," in *2021 IEEE World Congress on Services (SERVICES)*. IEEE, 2021, pp. 47–53.

[29] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Deepcog: Cognitive network management in sliced 5g networks with deep learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 280–288.

[30] K. Gao, D. Li, L. Chen, J. Geng, F. Gui, Y. Cheng, and Y. Gu, "Incorporating intra-flow dependencies and inter-flow correlations for traffic matrix prediction," in *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*. IEEE, 2020, pp. 1–10.

[31] P. Le Nguyen, Y. Ji *et al.*, "Deep convolutional lstm network-based traffic matrix prediction with partial information," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 261–269.

[32] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2018, pp. 231–240.

[33] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[34] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, J. Lang, Ed. ijcai.org, 2018, pp. 3634–3640.

[35] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 890–897.

[36] R. Dai, S. Xu, Q. Gu, C. Ji, and K. Liu, "Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3074–3082.

[37] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, S. Kraus, Ed. ijcai.org, 2019, pp. 1907–1913.

[38] B. Krithikaivasan, Y. Zeng, and D. Medhi, "Generalized cost-function-based forecasting for periodically measured nonstationary traffic," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 38, no. 5, pp. 1105–1117, 2008.

[39] Apollo, "Smart transportation solution; autonomous driving solution; intelligent vehicle solution." 2020. [Online]. Available: https://apollo.auto/index.html

[40] B. Claise, G. Sadasivan, V. Valluri, and M. Djernaes, "Cisco systems netflow services export version 9," 2004.

[41] Doris, "A fast mpp database for all modern analytics on big data." 2020. [Online]. Available: http://doris.apache.org/master/en/

[42] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016.

[43] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

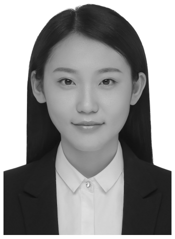
[44] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, vol. abs/1803.01271, 2018.

[45] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

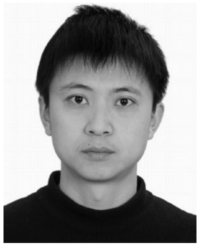
[46] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[47] K. Poularakis, Q. Qin, F. Le, S. Kompella, and L. Tassiulas, "Generalizable and interpretable deep learning for network congestion prediction," in *2021 IEEE 29th International Conference on Network Protocols (ICNP)*. IEEE, 2021, pp. 1–10.

[48] IntegNet, "Interrelated-temporal graph convolutional network." 2023. [Online]. Available: https://github.com/wzhCode/IntegNet

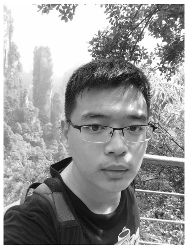


Zhaohua Wang is currently a Ph.D. candidate at the Institute of Computing Technology, Chinese Academy of Sciences. She received her B.S. degree in communication engineering from University of Electronic Science and Technology of China in 2017. Her research interests include Internet measurements and data center networks.



Zhenyu Li received the BS degree from Nankai University in 2003 and the PhD degree in Graduate School of Chinese Academy of Sciences (CAS) in 2009. He is a professor at the Institute of Computing Technology, CAS. His research interests include Internet measurement and Networked Systems.

Heng Pan received the PhD degree in computer science from University of Chinese Academy of Sciences in 2018. He is an associate professor at the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include SDN/NFV, distributed system and in-network computation.



Guangming Liu received the master's degree in computer science from Harbin University of Science and Technology, Heilongjiang, China, in 2015. He is currently a Senior Engineer with Baidu, Inc.

Yunfei Chen received the master's degree in software and Internet application technology from the School of Software and Microelectronics of Peking University, Beijing, China, in 2009. He is currently a Senior Engineer with Baidu, Inc.

Qinghua Wu received the Ph.D. degree from ICT/CAS in 2015. He is currently an Associate Researcher at ICT/CAS. His research interests lie in network transport protocol and Internet measurements.

Gareth Tyson received the PhD degree from Lancaster University in 2010. He is an Assistant Professor at Hong Kong University of Science and Technology (GZ). His research interests include Internet measurements and content distribution.

Gang Cheng is currently the Distinguished Engineer of Baidu Inc., the area technical lead for networking, including physical networking, virtual networking, network services, CDN and edge networking. Prior to joining Baidu Inc., he was an Director of Engineering of Alibaba Group, responsible for next generation cloud network technologies, hybrid network products, and network architecture. From 2010 to 2014, he was a Senior Software Development Engineer with Microsoft, Redmond, WA, USA. He received the B.E. and M.E. degrees in information engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 1997 and 2000, respectively, and the Ph.D. degree in electrical engineering from the New Jersey Institute of Technology, Newark, NJ, USA, in 2005.