Dynamic allocation of microservices for virtual reality content delivery to provide quality of experience support in a fog computing architecture

Derian Alencar [1], Helder Oliveira (Co-Advisor)[2], Denis Rosário (Advisor) [1]

[1]Federal University of Pará (UFPA) - Belém, PA, Brazil

[2]Federal University of ABC (UFABC) – Santo André, SP, Brazil

derian.alencar@itec.ufpa.br, helder.oliveira@ufabc.edu.br,

denis@ufpa.br

Abstract. *Virtual Reality (VR) content is gaining popularity, demanding solutions for its efficient distribution over the Internet. Microservices present an ideal model for deploying services at different levels of a Fog computing architecture for managing traffic and provide Quality of Experience (QoE) guarantees to VR content. However, it is crucial to efficiently find the fog node to allocate the microservices, which directly impact the QoE of VR services. This paper proposes INFORMER, an integer linear programming model aiming to optimize the placement of caching services according to delay, migration time, and resource utilization rate. Based on the insights from INFORMER, the paper presents Fog4VR, a mechanism for the dynamic allocation of content based on a heterogeneous microservice architecture. Results obtained with INFORMER serve as a baseline to evaluate Fog4VR on different scenarios using a simulation environment. Results demonstrate the efficiency of Fog4VR compared to existing mechanisms in terms of cost, migration time, fairness index, and QoE.*

1. Introduction

Virtual Reality (VR) technologies are growing in popularity. For instance, Youtube started supporting 360° content for VR playback, which uses Video-on-Demand (VoD) to partition a VR video into spatially related tiled videos. However, VR services have stringent requirements, such as a latency below *20 ms* and a bitrate greater than that of traditional videos, due to its panoramic nature and high video resolution. Hence, it is challenging to deliver VR video streams with QoE support over the current communication infrastructure [Li et al. 2018].

In this context, fog computing enables to offer VR services closer to mobile users, ensuring the delay and bandwidth requirements for content distribution [Rosário et al. 2018]. A fog infrastructure can expand dynamically according to the user demand for VR video streams, where a microservices architecture is ideal for deploying the components for VR video distribution. In particular, content delivery can benefit from a microservice architecture because it reduces the cost to instantiate and migrate cache instances according tso changes in the behavior of user demand [Tian et al. 2018]. Consequently, the content distribution system can adapt faster to changes in user demand and reduce resources wasted on unnecessary service instances.

Allocating microservices in a fog infrastructure to provide Quality of Experience (QoE) support for VR video streams comprises two steps: (*i*) *decision and allocation of microservice*, and (*ii*) *Content migration*. The first step finds the best node for deploying the microservice in a heterogeneous fog computing architecture. The second step encompasses the transference of content and forwarding of user requests to the allocated fog

node. In this context, the decision step must have an efficient design because the location of caching microservices influences migration time and delay, directly impacting the performance of VR services. Hence, the design for the decision step must take into account different metrics to assess the performance of fog nodes, networks, and users to make efficient decisions.

This paper presents the contributions in the master thesis [Alencar 2022], which tackles the challenge of allocation of microservices for VR content delivery in a fog computing architecture. We also take into account the user side using QoE metrics as a way to evaluate our model and corroborate our proposal. The research conducted and presented in this thesis advances the state-of-the-art in in the following ways: i) First, we design a controller to allocate and migrate microservices in a heterogeneous fog computing architecture called Fog4MS. ii) We provide an optimization model for microservices positioning called INFORMER, which considers transmission delay, content migration time, and resource utilization rate of a fog node to determine the optimal position for allocation. Next, based on the insights from INFORMER, we introduces Fog4VR to distribute VR content with QoE support using the concepts of microservices and heterogeneous fog architectures. It uses the same parameters as INFORMER to identify suitable fog nodes to allocate microservices, improving the QoE of VR contents. iii) Simulated experiments shows the proximity of Fog4VR to the optimal results obtained with INFORMER. For instance, Fog4VR reduces cost in 7% and migration time in 12%, while delivering VR video stream with QoE 50% better than compared to existing mechanisms.

2. Related Works

This section presents a brief summary of the state-of-the-art on dynamic microservices allocation for VR video streaming. Many papers have tried to solve the challenges of this scope, however, many of these works do not consider all the aspects presented in the problem. For example, some works only account the QoS characteristics, which does not characterize the experience obtained by the end-user. Other works take into account the QoE only in part, considering only one metric and not taking into account the nuances related to the video streaming characteristic of the VR application.

Table 1 summarizes the main characteristics of reviewed studies intended to provide content allocation regarding QoE and Quality of Service (QoS) awareness, Video on Demand (VoD) capability, VR aspects, and the allocation mechanism technique. These characteristics prevent users from abandoning the service due to stalls, stalls duration, and the playback start time. A content allocation mechanism must dynamically manage each request, finding the fog node capable of meeting resource demands based on QoS and QoE characteristics of VR streaming services to maximize QoE. However, most works consider only one aspect. The lack of VoD support can lead to problems related to this application's peculiarities, mainly associated with video playback user perception (QoE). Finally, it is preferable to use a heuristic technique as this has less computation time and complexity. To the best of our knowledge, only Fog4VR considers every critical characteristic for existing microservice allocation in a fog computing environment for VR distribution with QoE support. More details about the related works can be found in the Thesis [Alencar 2022].

3. Allocation Architecture and Mechanism

This section describes the fog computing architecture and mechanism for the dynamic allocation of VR microservices.

Table 1. Summary of Characteristics of Related Works

Works	Characteristics				
	QoE	QoS	VoD	VR	Approach
[Rigazzi et al. 2019]			✓	✓	Mathematical Modeling
[Mehrabi et al. 2021]		✓		✓	Heuristic Model
[Apostolopoulos et al. 2020]		✓			Game Theory (PNE)
[Ni et al. 2017]		✓	✓		Petri Network
[Mishra et al. 2020]		✓			AHP-EV
[Yousefpour et al. 2019]		✓			ILP+Greedy
[Mahmud et al. 2019]	✓				Fuzzy
[Lai et al. 2020]	✓	✓			ILP+Heuristic Model
Fog4VR (Proposal)	✓	✓	✓	✓	AHP

3.1. Architecture

We consider fog nodes deployed anywhere in a network organized into tiers between mobile devices (at the bottom) and cloud (at the top) [Rosário et al. 2018]. The cloud keeps the original copy of all VR content. It also distributes VR content for each user request and maintains an overview of each service and node status. Moreover, the cloud runs an allocation mechanism, such as Fog4VR, to select the fog node, allocate the microservice, and distribute VR content adaptively and proactively.

A heterogeneous organization of fog nodes consists of a computational infrastructure with various characteristics to allocate content as closely as possible to the user. Each fog node is represented by $f_i \in F$, which has a unique identity $i \in [1, n]$. For instance, a microservice for VR streaming could be deployed in a fog node f_i to speed up content distribution while improving the QoE. As a result, a fog node f_i could have one or more instances of microservices, which deliver the requested content to the user. Finally, the Client application requests and displays VR content to users.

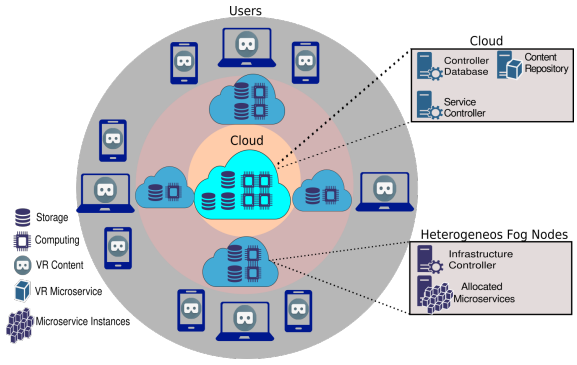


Figure 1. Architecture to deploy virtual reality VoD with microservices

3.2. Mechanism Operation

The Fog4VR mechanism is found in the *Service Controller* module. It manages the decision steps (*i.e.*, positioning of microservices in computing at a given fog node f_i) and content migration (*i.e.*, transferring content and directing requests to that node). Fog4VR receives the microservice request $m \in M$, which is a 3-tuple with content id id, content size s, and location l of the microservice requisition. Based on such information, the mechanism verifies which fog f_i is suitable to allocate the microservice m. To this end, it computes the resource utilization rate u_i based on microservice size m_s, allocated memory Am_i, and total storage available Ts_i in a given fog f_i.

If the fog node f_i has enough storage resources to allocate the microservice m, then Fog4VR must compute the migration time Tm_i to transfer the microservice from the cloud to a given fog node f_i, where W is the TCP window size, and $d_{i,r}$ is the packet transmission time. Lastly, it updates the vector L_i with the values of delay d_{f_i,m_l}, migration time Tm_i, and resource utilization rate u_i.

Fog4VR mechanism computes the cost C_i for allocating the microservice m in given fog node f_i based on Eq. 1. The cost C_i takes into account different metrics (*i.e.*, delay d_{f_i,m_l}, content migration time Tm_i, and resource utilization rate u_i), which have a varying degree of importance.

$$C_i = w_1 \cdot d_{f_i,m_l} + w_2 \cdot Tm_i + w_3 \cdots u_i \tag{1}$$

Fog4VR considers a multi-criteria decision-making method to balance inputs with different degrees of importance, where we argue AHP to compute the best response according to the significance of each parameter to another. Specifically, AHP decomposes a complex problem into a hierarchy of simpler sub-problems, combining qualitative and quantitative factors for analysis. Fog4VR mechanism builds a comparison matrix $V_{j,k}$ for each fog node f_i to compare all pairs of criteria based on Eq. 2.

$$V_{j,k} = \begin{array}{c} \\ d_{f_i,m_l} \\ Tm_i \\ u_i \end{array} \begin{array}{c} \begin{array}{ccc} d_{f_i,m_l} & Tm_i & u_i \end{array} \\ \begin{pmatrix} 1 & 4 & 8 \\ 1/4 & 1 & 2 \\ 1/8 & 1/2 & 1 \end{pmatrix} \end{array} \rightarrow [0.72\ 0.18\ 0.10] \tag{2}$$

As a result, we obtain the eigenvector $P = [0.72,0.18,0.10]$, indicating the weights of metrics, such as 0.72 for delay (d_{f_i,m_l}), 0.18 for migration time (Tm_i), and 0.10 for resource utilization rate (u_i). These weights are used to compute the cost Eq.1 for allocating a cache microservice m in a given fog node f_i. At the end of the process, the fog node with the highest score is chosen, and the microservice is allocated to the node.

4. INFORMER optimization model

We introduce the INFORMER optimization model to perform the dynamic allocation of VR microservices in heterogeneous fog computing environments, minimizing the delay and consequently maximizing the QoE. The results derived by INFORMER can be used as a benchmark to those achieved by other heuristics, due to INFORMER representing the optimal solution for the same scenario.

INFORMER aims to maximize QoE following Eq. 3. Consequently, the equation seeks to minimize the delay and migration time to better QoE. This minimization is essential because VR streaming is very sensitive to delay. In this case, low delay leads to a smaller number and duration of stall events, increasing the overall QoE of VR services. The INFORMER model chooses the fog node with lower delay and minimum migration time to deploy the microservice. Moreover, INFORMER allocates the microservice M in a fog node following the restriction, where each microservice m must be allocated in a fog node according to Eq. 4. Additionally, we cannot overflow the maximum fog storage capacity of Ts_i, as shown in Eq. 5. Finally, we do not exceed the bandwidth B_i limit of each fog f_i, as shown in Eq. 6. Therefore, INFORMER returns a binary variable $\vartheta_{m,f}$, which correlates for each microservice the index of the fog to be allocated.

$$Min \quad D = \sum_{m=0 \in M} \sum_{f_i=0 \in F} (d_{f_i,M_{l_m}} + d_{z,M_{l_m}}) \times \vartheta_{m,f_i} \tag{3}$$

Subject to:

$$\sum_{f_i=0 \in F} \vartheta_{m,f_i} = 1 \; \forall m \in M \tag{4}$$

$$\sum_{m=0 \in M} \vartheta_{m,f_i} \leq Ts_{f_i} \; \forall f_i \in F \tag{5}$$

$$\sum_{m=0 \in M} \vartheta_{m,f_i} \leq B_{f_i} \; \forall f_i \in F \tag{6}$$

5. Experimental Results

Experimental results were implemented in the NS3, and the source code is available on GitHub[1]. Each simulation was performed 33 times with different randomness seeds, and the results show the values with a 95% confidence interval. The scenario considers a varying number of requests to microservices (*i.e.*, 20, 40, 60, 80, and 100), modeled according to a Poisson distribution. The microservice considers a video streaming for VR service based on a MPEG-DASH application. Each user can choose the video from a catalog of 100 different VR content. The selection of video is given by the Zipf distribution with α = 0.7 so that the preference for content is better distributed. Videos have a fixed duration of 30min and have an encoding of 25Mbps for a 4K VR stream. The scenario considers the virtual topology of the FIBRE project network to assign the delays. More simulation parameters and evaluation metrics can be found in the Master thesis [Alencar 2022].

We evaluate our proposal into two use cases; (i) *Fog4MS* aims to allocate VoD microservices in a fog computing architecture using the AHP decision making described in Section 3.2, but it tries to allocate with little migration time while making a load balance across the network. In summary, Fog4MS do not prioritize the elements which return a better QoE for users, its preference give a better service from the service providers perspective, *i.e.*, without consider the QoE. (ii) *Fog4VR* aims to allocate VR microservices in a fog computing architecture with QoE support. Fog4VR considers all steps introduced in Section 3.2.

5.1. Fog4MS Results

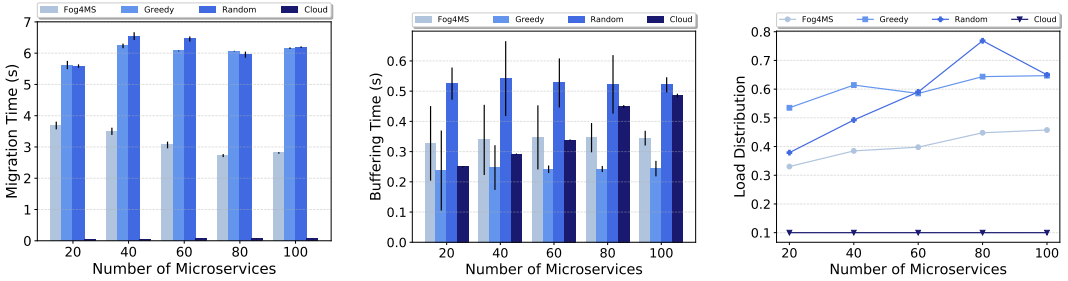
Figures 2(a) and 2(b) present the average time of content migration and video buffering when allocating the microservice in the fog, respectively. The video buffering has a direct impact on the QoE perceived by VoD users, as the shorter the *buffering* time, the higher the user's QoE and the lower the video abandonment rate. Thus, it is essential to observe metrics such as service migration time and buffering time for QoE analysis.

Figure 2(a) shows that the cloud has a non-existent migration time, as the content is already stored, and it is only necessary to instantiate the service. However, it can be seen in Figure 2(b) that the initial buffering time of the video increases with the number of microservices. It occurs because the cloud receives all the requests, so its load increases with the number of microservices, degrading its performance. In turn, the greedy mechanism decides based on the latency between the client and the fog point, so it always has a low buffering time. However, this mechanism has a poor performance concerning content migration time, as it often allocates the video to remote points in the network. The random mechanism has an arbitrary decision that varies significantly and its performance is always unsatisfactory. Finally, Fog4MS has intelligent decision-making that considers

[1]https://github.com/D3F3R4L/Fog4VR

latency and migration time, obtaining the lowest content migration time, even if there is a slight increase in buffering, *i.e.*, the cost/benefit of the mechanism is entirely justifiable. It is essential to highlight that Fog4MS is the only method to reduce content migration time with the increase in the number of microservices.

Figure 2(c) presents the fairness index for the use of fog points resources in the scenario simulation. This index expresses how much the load is being distributed across the network. The worst case is to use only the cloud, *i.e.* when allocating microservices in only one location. The greedy and random mechanisms have a similar distribution. In a random mechanism, the chance for allocation in fog is the same among all possible fogs. The index of Fog4MS mechanism is 33% and 30% lower than the indexes of greedy and random mechanisms, respectively. Furthermore, Fog4MS is still able to distribute the load on the network when necessary, as shown by the increase in the number of microservices. Thus, Fog4MS provides higher efficiency in content distribution since its decision-making prioritizes efficiency in favor of network balancing. The results of Fog4MS is better described on [de Alencar et al. 2020].



(a) Migration Time (b) Buffering Time (c) Load Distribution of Microservices

Figure 2. Simulation results for Fog4MS

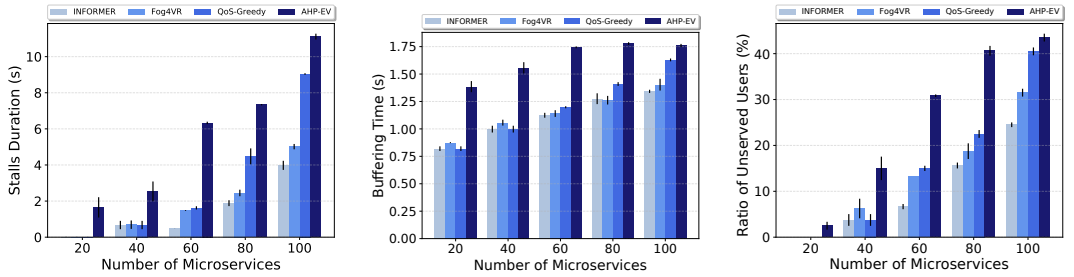
5.2. Fog4VR Results

As Fog4VR focus on the support for QoE, Figure 3 refers to the QoE results in terms of stall durations, buffering time, and percentage of unserved users, provided by different allocation mechanisms for VR distribution according to a different number of microservice requests. The results in Figure 3(a) indicates that Fog4VR provided the lowest number of stalls with the shortest duration, which is an essential behavior since high values may induce users to leave the VR service entirely. This performance is explained by the fact that Fog4VR prioritizes VR microservice allocation on fog nodes with lower delay, taking into account the migration time and resource utilization rate, enabling Fog4VR to choose the best locations for the VR microservice to minimize the number and duration of stalls.

The Fog4VR heuristic provides results close to the best solution available (*i.e.*, INFORMER optimization model). Specifically, Fog4VR increases the number of stalls by 50% in the worst case, and the stall duration is by 5% to 40% compared to INFORMER. Additionally, Fog4VR reduces the number of stall events by 45% to 75% and the duration of stall events by 54% to 74% compared to the AHP-EV mechanism depending on the number of microservice requests. Compared to QoS-*Greedy*, Fog4VR reduces the number of stall events by 33% to 45% and the stall duration by 9% to 45% because QoS-*Greedy* focuses on allocating microservice in fog nodes with a shorter delay. QoS-*Greedy* lacks other relevant metrics, which is crucial for optimal decision-making, leading to bad decisions and overloading fog nodes. This deficiency is more prominent in highly demanding scenarios where other metrics (*i.e.*, utilization rate) are essential to achieving a high QoE.

Figure 3(b) shows that INFORMER and Fog4VR have almost the same buffering time (*i.e.*, initial buffering time). This behavior is because Fog4VR can efficiently determine the fog node to allocate microservice based on multi-criteria metrics combined with different degrees of importance for each metric, leading them to better allocation decisions for users with low delay without overloading the fog nodes. QoS-*Greedy* have similar performance compared to INFORMER for scenarios with up to 60 microservice requests, *i.e.*, low demand. Simultaneously, the buffering time is 20% worst than INFORMER for a scenario with more than 80 microservice requests. The AHP-EV mechanism tends to assign users to more distant locations like the cloud, which gives more delay to users and, consequently, a higher buffering time.

Figure 3(c) shows the ratio of users who will probably be unsatisfied with the VR experience obtained.Results indicate that Fog4VR have similar performance compared to INFORMER since Fog4VR provides a lower number of stalls with a short duration, leading to a lower number of unsatisfied users. In turn, AHP-EV presented the worse ratio in all scenario cases. QoS-*Greedy* have a similar ratio of unsatisfied users compared to INFORMER and Fog4VR in low demand scenarios. Fog4VR have 11% to 22% less unsatisfied users than QoS-*Greedy*, 27% to 56% less unserved users than AHP-EV, and only 20% to 28% more unsatisfied users than INFORMER.



(a) Stalls Duration (b) Buffering time (c) Percentage of Unserved User

Figure 3. Simulation results for Fog4VR

Our performance evaluation analysis identified that Fog4VR have the best overall performance for QoE and service provider perspectives compared to other allocation mechanisms. Fog4VR has a multi-criteria that considers delay, migration time, and resource utilization, which is desirable for the microservice allocation and VR application. Therefore, Fog4VR provides higher efficiency in distributing content since its decision-making prioritizes efficiency in favor of balancing the network, or vice-versa, when necessary, delivering more QoE to users with better resource usage. The results of Fog4MS is better described on [Alencar et al. 2022].

6. Conclusion and Thesis Impact

The necessity to deliver a good quality of experience to the users in VoD services has brought a great challenge to researchers and companies service providers, especially with VR applications. In this sense, we propose the Fog4VR mechanism which takes into account delay, content migration time and utilization rate in the fog for the allocation of microservices VR content in the fog computing infrastructure.

The results obtained showed that the Fog4VR mechanism can reduce stalls and stalls duration by almost half, compared with AHP-EV and Greedy mechanism, while staying closest to the INFORMER optimal solution. For future works, the Fog4VR can be extended to manage the allocation of services in edge mobile computing in conjunction with UAVs to support and increase the QoS of applications in a challenged scenario.

7. Publications

The results of this Master Theses are published on:

Table 2. Summary of Results Published

Works	Qualis	Local	Google Scholar Citations
[Alencar et al. 2022]	A4	SBRC	-
[de Alencar et al. 2020]	A2	TNSM	11
[Santos et al. 2020]	A1	Computer Networks	7

References

Alencar, D. (2022). Dynamic allocation of microservices for virtual reality video delivery to provide quality of experience support in a fog computing architecture.

Alencar, D., Both, C., Antunes, R., Oliveira, H., Cerqueira, E., and Rosário, D. (2022). Dynamic microservice allocation for virtual reality distribution with qoe support. *IEEE Transactions on Network and Service Management*, 19(1):729–740.

Apostolopoulos, P. A., Tsiropoulou, E. E., and Papavassiliou, S. (2020). Risk-aware data offloading in multi-server multi-access edge computing environment. *IEEE/ACM Transactions on Networking*, 28(3):1405–1418.

de Alencar, D., Rosário, D., Cerqueira, E., Both, C., and Antunes, R. (2020). Distribuição de conteúdo sob demanda através da alocação de microserviços dinâmicos na borda e núcleo da rede. In *Anais do XXXVIII SBRC*, pages 575–588, Porto Alegre, RS, Brasil.

Lai, P. et al. (2020). QoE-aware user allocation in edge computing systems with dynamic QoS. *Future Generation Computer Systems*, 112:684–694.

Li, J. et al. (2018). Modeling QoE of Virtual Reality Video Transmission over Wireless Networks. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–7.

Mahmud, R. et al. (2019). Quality of Experience (QoE)-aware placement of applications in Fog computing environments. *JPDC*, 132:190–203.

Mehrabi, A., Siekkinen, M., Kämäräinen, T., and yl Jski, A. (2021). Multi-tier cloudvr: Leveraging edge computing in remote rendered virtual reality. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(2):1–24.

Mishra, S. et al. (2020). Dynamic Resource Allocation in Fog-Cloud Hybrid Systems using Multi-criteria AHP Techniques. *IEEE Internet of Things Journal*.

Ni, L. et al. (2017). Resource allocation strategy in fog computing based on priced timed petrinets. *IEEE Internet of Things Journal*, 4(5):1216–1228.

Rigazzi, G. et al. (2019). An Edge and Fog Computing Platform for Effective Deployment of 360 Video Applications. In *2019 IEEE WCNCW*, pages 1–6. IEEE.

Rosário, D. et al. (2018). Service Migration from Cloud to Multi-tier Fog Nodes for Multimedia Dissemination with QoE Support. *Sensors*, 18(2).

Santos, H., Alencar, D., Meneguette, R., Rosário, D., Nobre, J., Both, C., Cerqueira, E., and Braun, T. (2020). A multi-tier fog content orchestrator mechanism with quality of experience support. *Computer Networks*, 177:107288.

Tian, Y. et al. (2018). A new live video streaming approach based on Amazon S3 pricing model. In *IEEE 8th CCWC*, pages 321–328.

Yousefpour, A. et al. (2019). FogPlan: a lightweight QoS-aware dynamic fog service provisioning framework. *IEEE Internet of Things Journal*, 6(3):5080–5096.