Elastic Collaborative Edge Intelligence for UAVs

***Abstract*—Unmanned aerial vehicles (UAVs), or say drones,**

I. Introduction

II. Conventional Collaborative Edge Intelligence and Its Limitations for UAV Swarms

In this section, we first introduce four conventional CoEI paradigms according to the involved participants, *i.e.*, cloud-device CoEI, edge-device CoEI, cloud-edge-device CoEI, and device-device CoEI, and then discuss their major limitations for UAV swarms.

A. Overview of Conventional CoEI Paradigms

To empower the resource-constrained end device intelligence, huge efforts have been dedicated to how to leverage the external powerful servers such as cloud and edge to relieve the unbearable computation burden at end devices. Specifically, by offloading partial or full computation tasks of end devices to external equipments (*i.e.*, cloud, edge, and other end devices), collaborated edge intelligence is then boosted, while in this article we mainly refer to the partial case as "CoEI".

Cloud-Device CoEI: To avoid the significant amount of data sent to the cloud over wireless links by the cloud-only approach and efficiently leverage the more and more abundant computational resources in mobile end devices, Kang *et al.* [1] proposed "Neurosurgeon", a collaborative intelligence framework between the cloud and mobile edge. To be specific, Neurosurgeon partitions a DNN inference task consisting of the execution of multiple layers into two parts, one executed locally on the device and the other one on the cloud, as shown in Fig. 1(a). By automatically identifying the optimal partition point in DNN and effectively distributing the computation between the cloud and device, it achieves the following benefits: end-to-end latency improvement, device energy consumption reduction, and cloud datacenter throughput improvement. Following [1], Zhang *et al.* [2] also studied how to achieve low-latency inference by cloud-device CoEI.

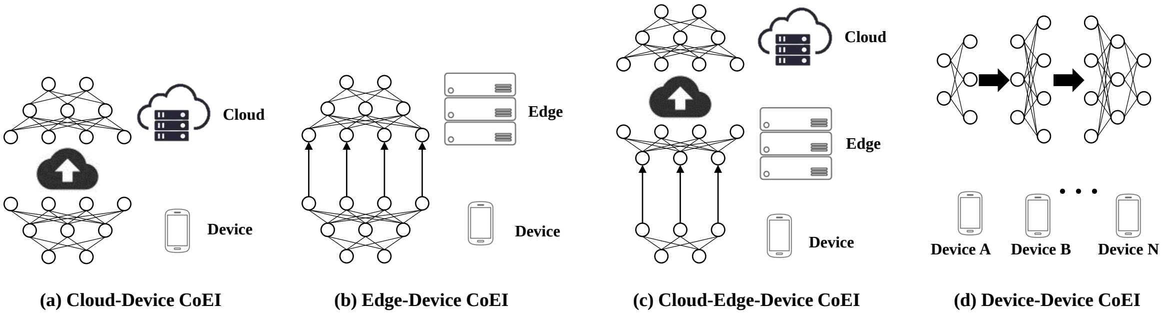
Edge-Device CoEI: With the emergence of edge computing, to avoid the uncontrolled large wide-area network latency for cloud access, Li *et al.* [3] proposed "Edgent", a on-demand DNN co-inference framework with edge-device CoEI. As illustrated in Fig. 1(b), Edgent adaptively partitions the DNN computation between the edge server and mobile devices in light of the available bandwidth, which exploits the computation power of the nearby edge server while reducing the data transfer latency. Note that Edgent considers a simple edge-device CoEI scenario consisting of only one mobile device and one edge server. Some recent works such as [4], [5] further studied the edge-device CoEI under the case with multi-edge server multi-end devices. Compared to the aforementioned cloud-device CoEI, the prominent advantage of edge-device CoEI is similar to that of edge computing, *i.e.*, relatively low end-to-end latency and energy consumption.

Cloud-Edge-Device CoEI: To enhance sensor fusion, data privacy and system fault tolerance for DNN applications, as illustrated in Fig. 1(c), [6]–[8] proposed to partition DNN models over a distributed computing hierarchy from cloud, edge, and end device, *i.e.*, cloud-edge-device CoEI. In addition to more kinds of involved computing nodes, this cloud-edge-device CoEI allows fast inference at local running some shallow portions of the DNN at end devices and the edge, by designing several early exit points in the DNN. Note that this cloud-edge-device CoEI is especially suitable for the inference task running over a prolonged period of time, *e.g.*, certain DNN-based IoT applications including anomaly behavior detection in a huge crowd. Under the cloud-edge-device CoEI, the small model at end devices could fastly perform preliminary feature extraction, as well as classification if the model is pretty confident; otherwise, further processing and eventual classification could be performed by the larger model in the edge and could.

Device-Device CoEI: Although the above CoEI paradigms make full use of the computing power of edge and cloud, the issues of latency-significant and unreliable wide-area network links between end devices and the remote cloud, and users' privacy concerns (*e.g.*, real-time inspection and warning in a smart home) still exist. Thus, [9], [10] proposed to conduct cooperative DNN inference over multiple heterogeneous edge end devices independent of both edge servers and remote cloud. As described in Fig. 1(d), all layers of a complete DNN inference task are partitioned and distributed among those resource-constrained end devices. Recently, few preliminary works [11], [12] studied how to leverage device-device CoEI in UAV swarm networks for UAV-enabled applications such as borders monitoring, surveillance, and forest fires detection, which, however, seldom consider the intrinsic unreliability of UAV swarm networks.

B. Limitations for UAV Swarms

The aforementioned CoEI paradigms have made great efforts to shift from traditional cloud only-based intelligence to cloud/edge/device collaboration-based edge intelligence, to adapt to the fast and accurate intelligence required by end devices. Yet, owing to the specific harsh environment where UAVs usually flow such as military border zones, offshore oil reservers, and forests, the communication link between UAVs and remote ground edge or cloud servers is generally unreliable and unstable, which limits the applicability of the first three CoEI paradigms. What's more, both nodes and links within a swarm of UAVs are also unreliable, *i.e.*, any UAV could be unavailable due to intermittent air-to-air (A2A) links and breakdown of the UAV by hardware/software errors,



(a) Cloud-Device CoEI

(b) Edge-Device CoEI

(c) Cloud-Edge-Device CoEI

(d) Device-Device CoEI

Fig. 1: Illustration of four conventional CoEI paradigms.

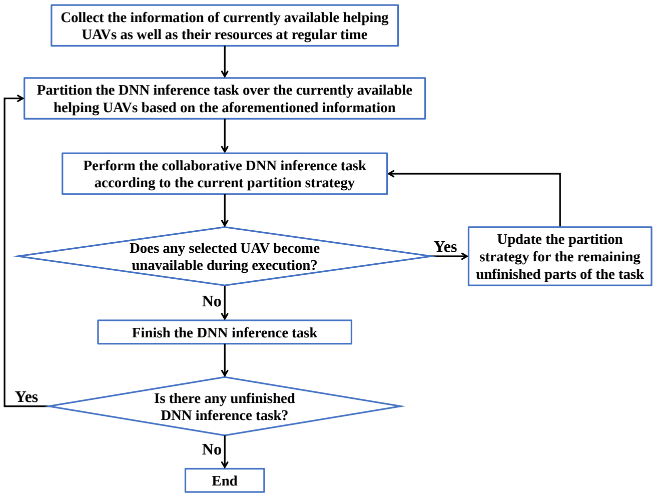

Collect the information of currently available helping UAVs as well as their resources at regular time
Partition the DNN inference task over the currently available helping UAVs based on the aforementioned information
Perform the collaborative DNN inference task according to the current partition strategy
Does any selected UAV become unavailable during execution?
Yes
Update the partition strategy for the remaining unfinished parts of the task
No
Finish the DNN inference task
Yes
Is there any unfinished DNN inference task?
No
End

Fig. 2: Main procedures of the proposed E-CoEI architecture for UAV swarms.

attacks, or being out of on-board battery power. This could result in the uncertain failure of collaborative DNN inference among the swarm, even if applying the device-device CoEI, that is, it is faced with an inevitable single point of failure.

III. Elastic Collaborative Edge Intelligence for UAV Swarms: Architecture, Advantages, and Novelty

In this section, to solve the aforementioned limitation of conventional CI paradigms when applying to UAV swarms, we propose the architecture of Elastic Collaborative Edge Intelligence (E-CoEI). We first present the architecture overview of the E-CoEI, and then introduce its advantages and novelty.

Architecture overview: In general, the proposed E-CoEI architecture enables a UAV swarm to collaboratively perform a DNN inference task in an *on-demand* and *flexible* manner, in face of probably unavailable UAVs due to node failure or A2A link failure, which is so called *"loosely-coupled collaboration"*. The "on-demand" means that all available UAVs in a swarm could be dynamically assembled together to contribute for the DNN inference task. The "flexible" implies that not only any available UAV could help execute any part of the DNN within its capability, but also the inference task distribution could be adaptively adjusted when some predetermined UAV becomes unavailable. Under the guidance of the above thought, the main procedures of the proposed E-CoEI architecture for UAV swarms are as shown in Fig. 2.

To be specific, for a UAV swarm, the information of the currently available UAVs as well as their resources including computation and storage should be sensed and collected at regular time, which is fundamental for the real-time decision of DNN task distribution. Consider there exists a single DNN inference flow consisting of multiple DNN inference tasks to be sequentially completed in a UAV swarm. Firstly, based on the aforementioned information, we partition the chosen DNN inference task over the currently available helping UAVs. Note that probably not all available UAVs could be selected in the collaborated inference, since some UAVs may be too far away or their available resources cannot satisfy the requirements. Secondly, following that partition strategy, each selected UAV executes its corresponding part of the DNN in order. If there exists some selected UAV becoming unavailable during the execution, we update the partition strategy for the remaining unfinished parts of the task. The DNN inference task will be eventually finished no matter whether the availability of any selected UAV changes or not during the collaboration. The next task will be executed according to the above process until no unfinished DNN inference task exists.

To better interpret the E-CoEI, we also construct a toy example in Fig. 3. As shown in the figure, six of the eight UAVs in the swarm are selected to collaboratively conduct the DNN inference task at time t, according to their current status and capabilities, *i.e.*, UAV#xx→UAV#xx→UAV#xx→UAV#xx. At time $t + \Delta t$, two previously selected UAVs (*i.e.*, UAV#xx and UAV#xx) are unavailable while UAV#xx becomes available, which triggers the adjustment of the partition strategy.

Advantages and novelty: The proposed E-CoEI architecture could well adapt to UAV swarms with high dynamics in both network topology and A2A wireless links. This features two main advantages when boosting the urgently needed edge intelligence for UAV swarms. i) The E-CoEI can enable collaborated intelligence with *strong invulnerability* over UAV networks, compared to most existing CoEI paradigms. Specifically, the collaborative DNN inference will not terminate if any previously selected UAV becomes unavailable, since the partition distribution strategy could always be undated based on the current network status. ii) The E-CoEI provides *high agility* and *flexibility* for collaborative AI within UAV

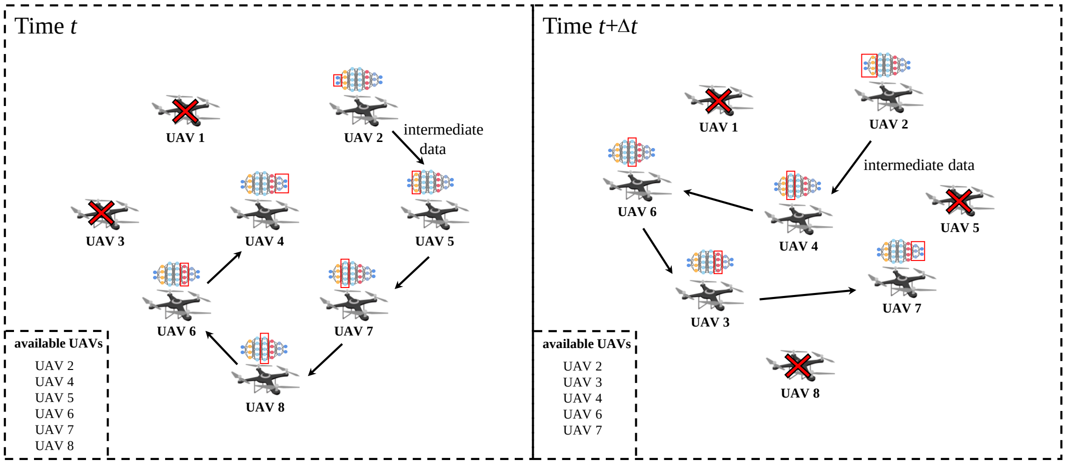


Fig. 3: A toy example of how the E-CoEI works.

swarms. No matter how the UAV network changes owing to the dynamic availability of partial UAVs, the CoEI can be self organized among the remaining UAVs. In a nutshell, the novelty of the proposed E-CoEI lies in that, it proposes a brand new CoEI framework perfectly matching the unique characterizes of UAV swarm networks, which is generally independent of specific CoEI participants.

IV. Performance Evaluation

In this section, we conduct a proof-of-concept evaluation to validate the feasibility as well as effectiveness of the proposed E-CoEI architecture. We designed a prototype system based on proposed E-CoEI architecture and deployed on a set of embedding devices. To reveal the influence of worker numbers to a CoEI system, we first measured the average inference frame rate under different number of devices. Subsequently, we simulated an unstable situation in which a worker's connection was not available to evaluate the elasticity of proposed E-CoEI system.

As Fig. 4 illustrated, the system is composed of two kinds of nodes, one is the master node, and the other is the worker node. The master node is responsible for planning collaborative inference strategy (i.e., tasks for each Worker node and connection relationship of workers), the worker node is the one which performs inference actually together with master node. Each node contains an embedded device and an ad-hoc module. The embedded device is in charge of excuting inference or scheduling task, and the ad-hoc module is used to support data transmission among different nodes. In this prototype, we use four nodes to construct the system, a NVIDIA Jetson Nano developer kit is setted on the master node as embedded device, and a NVIDIA Jetson TX2 as well as two Jetson Nanos are deployed on three worker nodes respectively to represent nodes with multiple performance. For the inference content, we selected object detection application and used Faster R-CNN as terget algorithm. Every time the application is executed, a video stream is transferred to the system and the position as well as classification of targets in frame can be obtained.

A. *Impacts of the number of nodes*

To better reflect the performance of the system, we measured the average frame rate of the inference result, which is a video contains targets' classifications and positions. In the experiment, we deployed a master node and different number of worker nodes, from one to three, to the prototype system. As shown in Fig. 5, with the increase of participating nodes number, the average frame rate of inference result increases gradually. The reason is that the more nodes perform inference collaboratively, the less workload every node is distributed, and the inference process is sped up in this way.

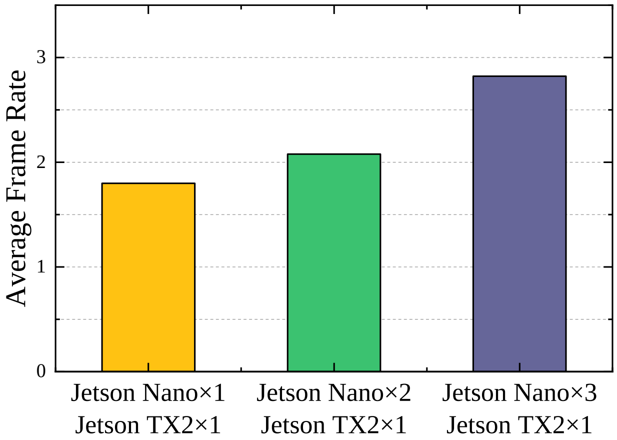


Fig. 5: The average frame rate of collaborative inference vary from different number of nodes.

B. *Performance of proposed E-CoEI architecture*

In order to verify the ability of the E-CoEI architecture to cope with the UAV swarm dynamic network environment, we set up an experiment to simulate the unavailable situation of some UAV network connections. In detail, we seperated this subexperiment into three stages. In the initial stage, we used four nodes (i.e., one master node and three worker nodes) to perform inference collaboratively, as Fig. 6 illustrated, the CoEI system can process the video stream at a speed of $\sim$3

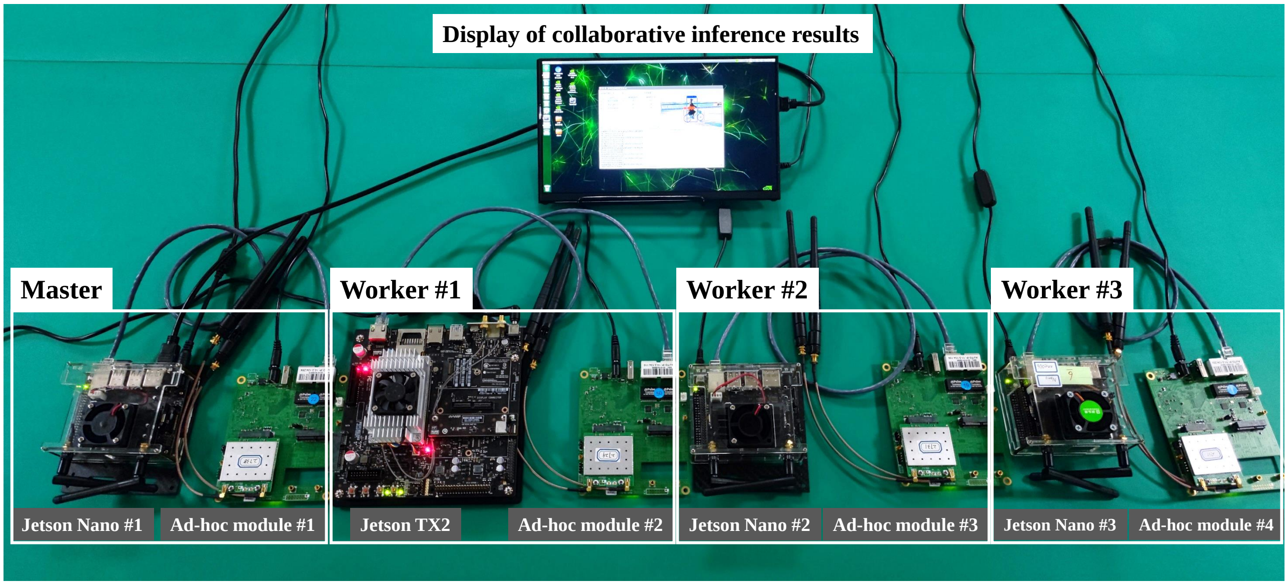


Fig. 4: Real E-CoEI prototype system setup.

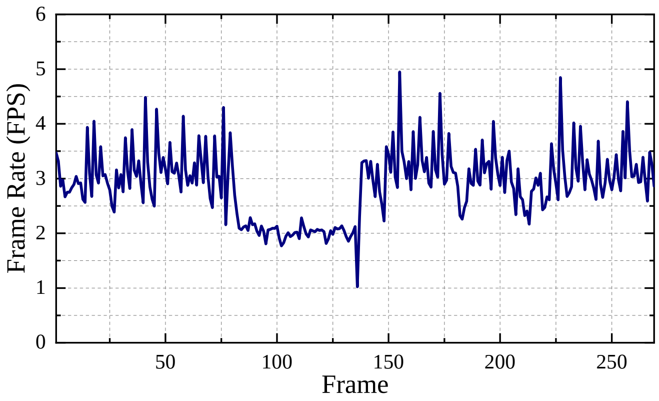


Fig. 6: Performance of proposed E-CoEI architecture under connection fluctuation situation.

FPS. In the second stage, we disabled the communication program of a worker node equipped with a Jetson Nano. In this situation, this worker is not an available node to the system. For the conventional CoEI paradigm, the process of collaborative inference will be terminated, because the workloads belong to the unavailable node cannot be processed. However, the E-CoEI prototype system perform in a different way, the inference process is not interrupted, but in a lower speed, ∼2 FPS in the experiment. In the last stage, we recover the communication program of aforementioned worker node. From Fig. 6, the collaborative inference speed reverted to previous level in the first stage. The reason for the phenomenon in the last two stage is that the information of the currently available nodes can be sensed at regular time by the master node, and when some nodes become unavailable during the execution, the scheduling strategy for the remaining tasks will be updated. In this way, the workload of the unavailable worker node is assigned to other available nodes and the inference task can be finished successfully. The reason why frame rate decreased in stage two is that only three nodes undertook the inference task in the second stage, and the average frame rate increases with the number of nodes, as discovered in Fig. 5.

V. Challenges and Research Directions

A. *Main Technical Challenges*

The proposed E-CoEI architecture could avoid the weakness of a single point of failure in the distributed CoEI within a UAV swarm. We also evaluate the effectiveness of the proposed E-CoEI via preliminary experimental results over the self-built airborne embedded system. Nevertheless, there exist many technical challenges to be carefully addressed in the E-CoEI before fully realizing its broad vision.

How to accurately and timely discover the status of UAV swarm networks? The current status of a UAV swarm including available UAVs as well as their resources is fundamental to the proposed E-CoEI, since it cannot make correct decision without such information. In practical, UAV networks exhibit high dynamics because of various factors, *e.g.*, UAVs' 3D mobility, unstable A2A wireless links, limited battery supply, harsh environments in specific applications such as battlefield, etc. The status of such networks could be sensed by existing sensing techniques such as xx, which inevitably brings non-negligible overhead to resource-constrained UAVs. Therefore, how to accurately and timely discover the status of UAV swarm networks with controllable extra overhead needs to be solved in advance.

How to optimize the distribution of a collaborative DNN inference task within available UAVs? Under CoEI, different UAV executes different part of the DNN model, and the distribution between UAV and corresponding executed part as well as the resource allocation should be optimized, which is challenging to E-CoEI. Firstly, although existing works [11], [12] aim to minimize the overall latency by optimizing the distribution strategy under the constraints of memory, computation, and UAV mobility, they assume that one UAV executes one single layer of the DNN model, which might result in a waste of UAV's on-broad resources. Relaxing such assumption (*i.e.,* one UAV may execute multiple successive layers) could better utilize the network resources of UAV swarm, which also makes the optimization problem harder. Secondly, different from most existing CoEI works considering latency only, in the proposed E-CoEI for UAV swarms, UAVs' energy consumption should be imposed as one of the key quality-of-service (QoS) metrics in the optimization, which increases the problem complexity. Lastly, even when restricting one layer to be executed on one UAV, the distribution problem is already NP-hard [11]. Thus, to be more practical for UAV swarms without extra ground support, the solution should be with low computation complexity and good performance guarantee at the same time.

How to timely respond to the unavailability of selected UAVs during execution? In the proposed E-CoEI architecture, it is envisioned that when some chosen UAV becomes unavailable during execution, the predetermined partition strategy will be updated immediately. Note that this unavailable UAV may be several hops away, which implies large transmission latency. Accordingly, a fast and efficient feedback scheme should be designed to inform that information effectively and efficiently. Besides, similar to backups in multi-hop routing, there should be backups for the DNN inference task partition to avoid unexpected unavailability and respond quickly. Considering the huge search space of the NP-hard partition problem, it is nontrivial to design such backups.

B. Potential Research Directions

Given the aforementioned technical challenges and emerging increasingly diverse applications of UAV swarms, we introduce several promising research directions in the proposed E-CoEI for future study as follows.

Management of collaboration for E-CoEI in hierarchical/clustering UAV swarms: In practical, many UAV swarms are organized in a hierarchical or clustering manner instead of purely ad hoc, where some UAVs compose a small group with a leading UAV in charge of the remaining UAVs. In such UAV swarms, the scope of collaboration for the E-CoEI should be subordinated to the upper organizational structure of the UAVs. To be specific, whether a collaboration inference task could employ those UAVs outside the current cluster or not, may depend on the collective behavior of the UAV swarm as well as the elastic QoS requirement of the task, which deserves further study.

Collaborative DNN inference-driven UAV swarm network protocols design: In the envisioned E-CoEI, there mainly exist two types of data to be transmitted in the network. One is the intermediate data of DNN inference, *i.e.*, output of some layer in the DNN, which is characterized by large volume and has a relatively low requirement on latency and reliability. The other is some control necessary information about network topology, available resources, and routing, which has small data quantities and demands low latency and high reliability. Traditional network protocols such as routing do not differentiate their diverse characteristics and requirements, which thus calls for novel network protocol design for the E-CoEI.

Scheduling of multiple collaborative DNN inference flows: Most existing studies about CoEI focus on optimizing the scheduling of a single DNN inference flow in the collaboration. In practical, there usually exist multiple DNN inference flows at a UAV or multiple UAVs to be scheduled, *e.g.*, a swarm of several UAVs are in charge of monitoring a target area via capturing images at different angles, thereby bringing about several object detection-oriented DNN inference flows. In light of multiple DNN inference flows in the E-CoEI, for the desired elasticity, the optimization of various network resources including participated UAVs, and computation/communication resources, together with execution sequence, is much more complicated than that in the single inference flow case.

Dedicated DNN structure design and efficient training for E-CoEI: Unlike common device-device collaborative DNN inference working in a relatively stable network environment, the failure or unavailability of the participating nodes (*i.e.*, UAVs) should be paid more attention to in the E-CoEI. Specifically, the predetermined UAVs involved in the participation may be unavailable due to attacks/interference, hardware/software failures, and low battery, which happens with high probability in UAV swarms, while how to design a proper DNN against the sudden but frequent node unavailability is less explored. Some recent studies [xx] propose so called "early exit" mechanism in the design of some particular DNN such as xxNet to timely output useful results. Nevertheless, they mainly focus on how to avoid the unavailability of the cloud in the cloud-device CoEI case. Additionally, how to efficiently train that dedicated DNN is also an interesting problem.

Secure collaboration of E-CoEI by blockchain: Similar to conventional CoEI, the E-CoEI contains massive intermediate data transfer among the UAVs therein in the collaborative inference, which could be taken advantage of by malicious participants. In such a distributed collaborative system, blockchain could be exploited to guarantee the secure collaboration of the E-CoEI, in the presence of some malicious UAVs. So far, how to enable the secure collaboration of the E-CoEI for UAV swarms remains an open problem.

VI. Conclusion

In this article, we have proposed the E-CoEI architecture, an invulnerable distributed COEI architecture focus on unstable A2A communication links and UAVs software/hardware breakdown. The preliminary evaluation over the self-built airborne embedded system demonstrated the effectiveness and feasibility of our proposed E-CoEI framework. There still lies a couple of technical challenges to realize its broad vision, such as UAV swarm status discovery, inference task distribution, etc. At the same time, there are promising research directions including UAV swarm management, network protocols design, multiple collaborative inference flows scheduling, dedicated DNN structure design, etc.

References

[1] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. N. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *Proc. ACM ASPLOS*, pp. 615-629, 2017.

[2] S. Zhang, Y. Li, X. Liu, S. Guo, W. Wang, J. Wang, B. Ding, and D. Wu, "Towards real-time cooperative deep inference over the cloud and edge end devices," *Proc. ACM Ubicomp*, vol. 4, no. 2, pp. 69:1-69:24, 2020.

[3] E. Li, Z. Zhou, and X. Chen, "Edge intelligence: On-demand deep learning model co-inference with device-edge synery," *Proc. MECOMM*, pp. 31-36, 2018.

[4] T. Mohammed, C. Joe-Wong, R. Babbar, and M. D. Francesco, "Distributed inference acceleration with adaptive DNN paritioning and offloading," *Proc. IEEE INFOCOM*, pp. 854-863, 2020.

[5] X. Tang, X. Chen, L. Zeng, S. Yu, and L. Chen, "Joint multiuser DNN partitioning and computational resource allocation for collaborative edge intelligence," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9511-9522, 2021.

[6] S. Teerapittayanon, B. McDanel, and H. T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," *Proc. IEEE ICDCS*, pp. 328-339, 2017.

[7] A. Yousefpour, S. Devic, B. Q. Nguyen, A. Kreidieh, A. Liao, A. M. Bayen, and J. P. Jue, "Guardians of the deep fog: Failure-resilient DNN inference from edge to cloud," *Proc. ACM AIChallengeIoT*, 2019.

[8] A. Yousefpour, B. Q. Nguyen, S. Devic, G. Wang, A. Kreidieh, H. Lobel, A. M. Bayen, and J. P. Jue, "ResiliNet: Failure-resilient inference in distributed neural networks," *Proc. FL-ICML*, 2020.

[9] R. Stahl, Z. Zhao, D. Mueller-Gritschneder, A. Gerstlauer, and U. Schlichtmann, "Fully distributed deep learning inference on resource-constrained edge devices," *Proc. SAMOS*, pp. 77-90, 2019.

[10] L. Zeng, X. Chen, Z. Zhou, L. Yang, and J. Zhang, "CoEdge: Cooperative DNN inference with adaptive workload partitioning over heterogeneous edge devices," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 595-608, 2021.

[11] M. Dhuheir, E. Baccour, A. Erbad, S. Sabeeh, and M. Hamdi, "Efficient real-time image recognition using collaborative swarm of UAVs and convolutional networks," *Proc. IEEE IWCMC*, pp. 1954-1959, 2021.

[12] M. Jouhari, A. K. Al-Ali, E. Baccour, A. Mohamed, A. Erbad, M. Guizani, and M. Hamdi, "Distributed CNN inference on resource-constrained UAVs for surveillance systems: Design and optimization," *IEEE Internet of Things Journal*, vol. 9, pp. 2, pp. 1227-1242, 2022.