Position/Survey Paper

The Future of Consumer Edge-AI Computing

Stefanos Laskaridis, *Brave Software*
Stylianos I. Venieris, Alexandros Kouris, Rui Li, *Samsung AI*
Nicholas D. Lane, *Flower Labs and University of Cambridge*

Abstract—
In the last decade, Deep Learning has rapidly infiltrated the consumer end, mainly thanks to hardware acceleration across devices. However, as we look towards the future, it is evident that isolated hardware will be insufficient. Increasingly complex AI tasks demand shared resources, cross-device collaboration, and multiple data types, all without compromising user privacy or quality of experience. To address this, we introduce a novel paradigm centered around EdgeAI-Hub devices, designed to reorganise and optimise compute resources and data access at the consumer edge. To this end, we lay a holistic foundation for the transition from on-device to Edge-AI serving systems in consumer environments, detailing their components, structure, challenges and opportunities.

Since their very advent, Deep Neural Networks (DNNs) have been getting larger in their attempt to be more accurate without losing generality. Simultaneously, higher accuracies have also been a result of combining multiple models (ensembles or cascades) or inventing more exotic architectures, manually or automatically, that offer higher capacity, better generalisation or fewer inductive biases [18].

More recently, there have been emerging trends in Artificial Intelligence (AI), generative or discriminative, which are changing the computational landscape quite significantly. On the one hand, the training of hyper-scale models that act as foundations in latent spaces for solving a multitude of downstream tasks in one or multiple modalities has been dominating computation in cloud AI. Prominent examples include Large Language Models (LLMs), text-to-image generation (out-painting) or generative image composition (in-painting). On the other hand, as devices become more capable, an increasing number of DNNs are deployed on-device, oftentimes required to run simultaneously. Furthermore, the advent of fields like Federated Learning (FL) and personalisation introduce on-device training workloads.

Despite the forward-looking use-cases, such workloads have been pushing the compute and memory requirements to unprecedented scales (Fig. 1), along with their data ingestion needs. However, individual edge device capabilities have not scaled at the same pace. While the consumer edge becomes increasingly populated by smart devices, these continue to operate as standalone entities in isolation from their compute environment. Therefore, there are many missed opportunities for shaping a common context to learn and perform higher-level or fidelity tasks under a collaborative environment.

As such, a gap exists between compute requirements and resource availability for deploying intelligence at the consumer edge, which is unlikely to be bridged only through traditional hardware scaling techniques. In this paper, we present a new paradigm for organising resources at the consumer edge when executing emerging AI-tasks. Departing from isolated devices and moving into more capable EdgeAI-Hubs, we argue that the *fluid sharing of compute* and the *among-device sharing of context information* are key ingredients of an architecture that would deliver on the requirements of modern AI-tasks, with *privacy* and *sustainability* as vital components for deploying of state-of-the-art AI at the consumer edge.

Deep Learning Trends

DNNs have traditionally grown in size in their strive for higher accuracies in pursuit of intelligence. Within a few years, we moved from the traditional multi-

Digital Object Identifier 10.1109/XXX.0000.0000000

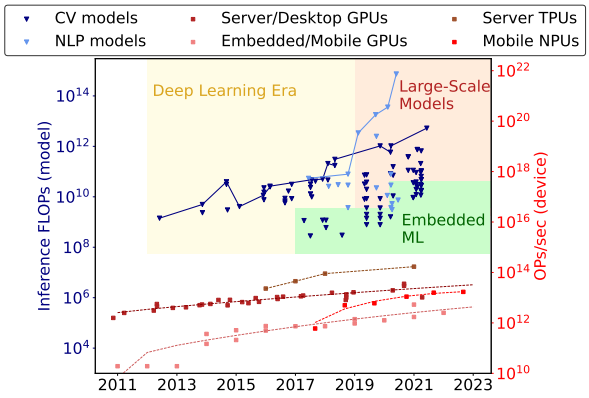

FIGURE 1: Evolution of DNNs operations (FLOPs) and hardware throughput (OP/s).

layer perceptron into deep networks of various forms and architectures. While the inductive bias of previous DNNs offered a convenient "shortcut" for learning in different modality inputs and a native mapping to widely available hardware, it turned out that convolutions were not "all that we needed".

The advent of transformers [18] brought a breath of fresh air into deep learning, and enabled the training of hyperscale models, able to organise and query the knowledge over massive datasets, and facilitated novel applications spanning across the personal and work life of the user. Such include NLP and vision related use-cases, such as next-gen multi-modal chat assistants or meeting and document abstractive and creative tools. Unprecedented accuracies and novel use-cases were accompanied by inflated model footprints, only able to be trained distributedly in large datacenters. Even at inference, these models remain notoriously difficult to deploy, and much more so with real-time performance [10]. Indicatively, running a 4-bit quantised version of Llama-2-7B on an M2 Max SoC (Metal) vs. a Galaxy S23 Ultra (OpenCL), yields 7.2× higher throughput, based on our experiments.

Besides execution time, the large memory footprint of recent models has a direct energy impact on hardware, which becomes the main scaling bottleneck [14]. Memory accesses dominate energy, with more than 100× higher consumption than computation, while on-chip caches often account for 50% of the processor's energy budget. Indicatively, executing TinyBERT (full-precision weights: 255MB) on mobile processors, such as the Edge TPU (8MB cache), requires an excessive amount of off-chip memory accesses and intensive use of the on-chip cache [14]. As a result, even if new mobile processors could execute such models faster, the battery would be drained at an unacceptable rate.

At the same time, privacy of user data has become a top priority. Thus, alternative models of decentralised training have appeared, namely FL and on-device personalisation. While game-changing, since it enables collaborative learning and pushes computation to where data reside, training on device creates a much more memory- and energy-consuming workload that not all mobile devices can support. Indicatively, training SmallBERT on device can consume more that 8GB of peak memory, while inference requires 1/16th of that [4]. Simultaneously, such in-the-wild deployments need to deal with data and system heterogeneity, along with partial availability and dubious robustness of clients, while competing for resources with other co-habitating workloads.

The Evolution of ML Compute

The adoption and continuous upscaling of DNNs would not have been possible without the scaling of compute capabilities. GPUs, and their re-purposing from graphical to neural accelerators, initially paved the way towards training deep networks, followed by specialised ASIC/FPGA accelerators.

On-device compute for DNNs. Nevertheless, DNNs were never intended to stay limited to the premises of the datacenter. AI advancements made their way into the consumer world [1] with smart devices of different shapes and forms, ranging from smartphones and wearables to IoT devices and even robots. Their omnipresence and sensing capabilities enabled the production of unprecedented scales of local data from various modalities, available for harvesting.

While initially fully offloaded to the cloud at the cost of privacy and latency, DNN computation has since progressively been "onloaded" to devices. What made this possible was advances in mobile hardware, which leverage integrated GPUs and Neural Processing Units (NPUs) on the same System-on-Chip (SoC) [7] or external accelerators, *e.g.* Edge TPU. In addition, EfficientML techniques have enabled performant on-device execution for a multitude of AI-tasks.

Still, in contrast to typical cloud-based infrastructure, the computational landscape in-the-wild is much more *heterogeneous* [1] and *failure-prone* [5], while the device still needs to remain responsive for tasks of the respective user. This brings new challenges that need to be tackled by device manufacturers and developers for making the mobile devices truly "smart."

On-device hardware vs. offloading. DNN compute requirements and the respective hardware have not necessarily scaled at the same pace (Fig. 1), as their development cycles are very different. In essence,

developing and deploying highly specialised hardware is a high-cost process and needs to be balanced from a utility-cost standpoint and from a device lifecycle perspective. Regardless, cloud infrastructure still remains much more capable and allows for performant general-purpose compute without the constraints or heterogeneity of the edge. However, this occurs at the expense of client latency, provider cost and privacy. Thus, the ubiquitous question on how much computation to offload or whether to dedicate on-device hardware is posed in the consumer Edge-AI space.

The State of Consumer Edge-AI

Currently, there are emerging applications in the consumer domain, still too costly to be deployed to the edge, or lack a common context that could be shared amongst devices in the ecosystem. In this setting, referred as Consumer Edge-AI 1.0, there is a lot of unfulfilled potential that is currently limited by the way intelligence is organised, deployed and distributed.

Devices are islands

While more and more devices penetrate the realm of the smart edge, they generally remain siloed, integrating standalone hardware to support their narrow requirements and generally do not advertise their capabilities to other local devices. Moreover, separate personal and work devices tend to be a common setup for legal and privacy reasons. In such an inelastic deployment, not only are many compute cycles wasted, but devices remain constrained in their own settings, and replicate similar sensor integration to perform different tasks. As such, the per-device cost is higher, the utilisation remains low and the hardware investment gets retired at the end of the device's lifecycle. While computation *offloading* and *split* computing [11] have been widely proposed before, they remain largely point-to-point and lack horizontal sharing of contextual information amongst user devices, therefore acting as mere remote accelerators. Thus, for personalising one's own GPT-assistant, individual devices would need to reach and offload data to third-party cloud services, as they currently lack a common data and compute fabric.

Shortcomings of previous paradigms

There have been various paradigms of organising ML execution between devices, offering varying levels of success and adoption, presented in Fig. 2. Specifically:

On-device ML [16]. The one end of the spectrum is to run AI-tasks locally on-device. While simple, it is far from simplistic as embedded and mobile devices come with severe constraints across computation, energy and thermals [10]. Thus, specialised hardware integration along with EfficientML techniques become core enablers [1]. Data-sharing remains minimal.

Centralised Learning. The other end suggests that storage and compute are outsourced from the mobile device to more powerful cloud resources. Initially theorised through the Mobile Cloud Computing (MCC) [20] paradigm, it remains today the status-quo method for training and deployment of ML workloads. Latency and privacy remain key issues of this paradigm, along with the need to transfer all data.

Collaborative Learning. Edge computing [20] has been motivated by the movement of compute closer to where the data are produced, *i.e.* the edge. It provides lower latency compared to cloud offloading (*i.e.* MCC) and can be anywhere in between the cloud and the end-device, including base station servers (*i.e.* MEC) and ambient devices (*i.e.* Fog Computing). These paradigms mainly focused on general compute sharing, and while forward-looking, they never found a "killer" use-case driving their adoption, and devices remained isolated with opportunistic collaboration if any. Closer to ML, this paradigm can manifest as Distributed Learning, with use-cases including inference offloading and distributed training.

Consumer Edge-AI 2.0

Given the accelerated rise of new AI-based use-cases and the inability of individual smart devices to scale up their capabilities and context to that dynamic, we propose a new architecture that aims at ML execution, through flexible sharing of compute resources with smarter placement of specialised hardware, so that edge devices' processing power is augmented sustainably without extreme duplication. At the same time and contrary to prior paradigms, their sensing capabilities can also be enhanced via sharing of their situational context to enable collaboration while respecting privacy.

Shared compute

Hardware resource allocation. Specifically, we shift from the paradigm of developing and integrating highly specialised NPUs into every available device to a model where higher-end and more general accelerators can be hosted in *central-device hubs*. These hubs can be standalone devices that only serve this purpose (*e.g.* a home accelerator) or "piggy-backed" into devices with longer life cycles omnipresent at the edge (*e.g.* a router or TV). Thus, hardware can take

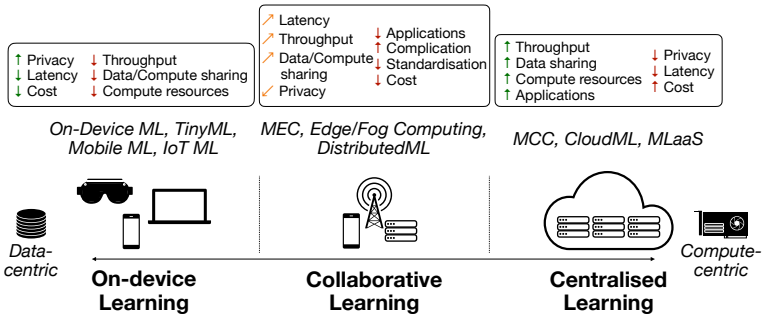

FIGURE 2: Consumer Edge-AI 1.0 paradigms. While not necessarily mutually exclusive, they provide different levels of interaction between the involved entities.

advantage of larger budget in terms of *area*, *cooling* and *power* to provide acceleration to more operations and users than before. In fact, this hardware can even be based on reconfigurable FPGAs [19], so that they can be adjusted to the user needs and available devices. Last, they can be designed in a way that many hubs can be interconnected to scale out capabilities.

We distinguish two levels of compute sharing, both of which act complimentarily to one another and need to be optimised in tandem to achieve resource allocation efficiency: *i) static partitioning* and *ii) dynamic resource sharing*. *Static partitioning* refers to the placement of dedicated compute units to devices. While certain IoT devices can become as thin as a sensor with a network adapter, others still need to maintain some autonomy or mobility. As such, one cannot simply centralise all available compute and strip devices out of any meaningful capabilities. Instead, a balance between the two ends of the spectrum is needed. For example, on-device inference may be a common AI-task, but opportunistic participation[1] in FL has not shown a real need so far. Thus, smartphones may not need train-capable hardware integration. Instead, a training-ready NPU could be integrated to a home hub where training can be offloaded.

Simultaneously, *resource sharing* can happen *dynamically* between device-hub or in a *peer-to-peer* manner, compatibly with Edge/Fog Computing [20]. Workloads can be offloaded on demand, based on either the static capabilities of the device or their instantaneous dynamic load. Effectively, one can think of resource partitioning and allocation as a generalised Knapsack Problem. How the equilibrium of resource partitioning is met is a function of *i)* the *nature* of AI-tasks at hand, *ii)* their *urgency* to complete, and *iii)* the *intersection* of tasks between devices.

[1] Participation in FL usually happens when device is plugged in and connected to WiFi.

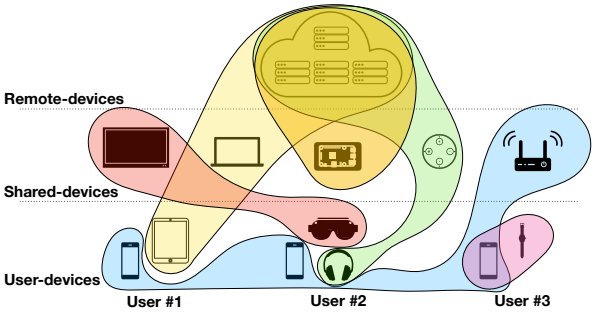

FIGURE 3: Privacy and collaboration zones between devices. These can be supported via rights management through device-owner groups and Access-Control Lists.

Networking & scheduling. Central to this elastic compute is a capable *multi-channel networking* infrastructure on top of which different requests for AI-tasks can be *scheduled* and *prioritised*. Such an infrastructure should offer multi-channel access, as different devices support diverse protocols (*e.g.* Wi-Fi, BLE, Zigbee, LoRa, etc.), load-balancing among available channels and bandwidth slicing for better Quality of Experience (QoE). Advancements in mobile wireless communications (B5G, 6G) can also facilitate the adoption of this paradigm, over sub-THz, THz or visible light communications (VLC) [2]. Additionally, task deadlines with preemption under multi-tenancy are core features for the scheduler to guarantee QoE for all active users. For example, the upscaling of live streaming video for a user would need to be given higher priority than the classification of newly acquired gallery photos on a user's device, which can be done offline. Thus, it is one of the hub's primary objectives not only to provide resources, but also to co-ordinate them.

Shared context

By having interconnected compute, it is also possible to share a common context between sensing devices to fulfill collaborative tasks. As such, a smart speaker may not only serve as a standalone device, but could also be used for recognising a user in a room along with their intent (e.g. work or entertainment) to personalise their experience by preloading their profiles on ambient devices. At the same time, it can serve as a secondary microphone for noise cancelling or even for intrusion detection along with a smart camera. Context sharing can be *i) explicit*, through sensor-data exchange or *ii) implicit*, by embedding subsets of available sensors into a common subspace that can be leveraged for different tasks. Last, different tasks can also share common DNN backends, instead of replicating them per device. For example, the obstacle detection subsystem of a robotic vacuum cleaner could share parts of a DNN with a pet surveillance camera. In fact, both can act as sensors for enhancing the classification result through multi-view perspectives, with one even offering active vision capabilities.

Privacy

In this era of AI, where information is eagerly digested by billion-scale models, consumer data are increasingly important to protect. Therefore, privacy becomes an innate design component of next-generation consumer Edge-AI.

Privacy-preserving ML. Specifically, we differentiate between privacy and robustness at inference and training time. In the former setting, the input and outputs[2] of the DNN are sensitive, rather than the model itself.[3] However, when training or personalising a model, along with the data involved, the model parameters and gradients become also privacy-sensitive, as they may leak user data. Overall, privacy spans in a continuous spectrum, *i.e.* a privacy budget [12], and therefore should be treated accordingly.

Trust zones. Simultaneously, different data have different sensitivity towards different entities/adversaries. For example, there are data that one might want accessible from home-owned devices but private to the public, such as holiday trip photos. Conversely, one could share their browsing preferences with a third-party for ad personalisation, but not with their household. The same applies between work and personal devices, a delineation that might not be straightforward in working-from-home settings. As such, trust zones are formed that shape the data flow of collaboration in Edge-AI, depicted in Fig. 3.

Sustainable-AI

Additionally to user privacy, it is also important to respect the environmental ecosystem in which Edge-AI operates. We capitalise on the fact that not all downstream tasks require overprovisioned DNNs running round the clock and save on energy and transmission costs. Simultaneously, we acknowledge the downsides of onloading more computation to the consumer edge where energy sources and efficiency may be suboptimal compared to large datacenters [14] and propose ways to offset them.

EfficientML. A particularly important component in any embedded deployment of AI is the optimisation of resources. Embedded and mobile devices integrate lower computational capabilities than state-of-the-art servers. To that direction, techniques from EfficientML [16] reduce the computational, memory or bandwidth requirements of AI-tasks, by means of changing the *architecture*, *representation* or *execution* [5], [9] of the DNN in an offline or dynamic manner. These, not only affect the users' QoE, but can drastically reduce energy consumption. For example, early-exit networks can leverage the difficulty or spatio-temporal relatedness [8] of inputs to preempt computation on-demand.

Lower manufacturing costs. Our paradigm also economises on the placement of highly-specific accelerators. By centralising certain components of the edge compute infrastructure, the manufacturing cost of mobile or IoT SoCs can be lower and the effective utilisation rate higher.

Device upcycling. The unprecedented pace of computer systems advancement means that devices become quickly deprecated and turn from useful companions to effectively e-waste. However, old devices still integrate various sensors and oftentimes enough compute power to be useful [17]. We embrace the previous-generation devices co-existence and integrate them in a sustainable manner to the smart edge context, through upcycling and repurposing.

Proposed Architecture

Here, we lay out a reference architecture of the systems implementing our paradigm and how different components interact to accomplish the end goal, *i.e.* supporting and enabling AI-task execution through a common computational and contextual fabric, while

[2]Along with the intermediate representations.

[3]From the user's standpoint. Model privacy may be of business-critical importance to the provider.

preserving user-privacy and sustainability. We select an edge-accelerated solution as on-device resources are clearly not sufficient for the upcoming workloads (Fig. 1) and *fully* offloading to resources beyond the edge hurts user-privacy and racks up provider costs.

Orchestrator

The consumer-edge ecosystem is inherently dynamic, multi-tenant and partially unavailable, mixing mobile and static devices of potentially different owners. While a peer-to-peer approach may initially seem tempting, we would soon be presented with high preemption rates due to device (un-)availability, along with data inconsistency or model staleness issues. Therefore, we adopt a server-client design where a central node (orchestrator) is responsible for the subscription and management of resources in the local edge. The orchestrator should reside in a non-mobile (for availability) device of the edge network (*i.e.* the EdgeAI-Hub) and should be relatively high-end in terms of compute and networking capabilities for fine-grained scheduling and high-fidelity cross-channel data communication, respectively. Finally, it can be co-located with an AI-capable device – such as a high-end TV or a powerful IoT hub, both of which are persistently available at the edge – or can be simply coordinating them. To avoid having a single point of failure or bottlenecks, there can be a secondary orchestrator residing on another capable device if such exists at the edge.

A reference design of the orchestrator is depicted in Fig. 4a. Effectively, the orchestrator is the coordinator of the AI-task processes among device resources, with optional cloud offloading, where Trusted Execution Environments (TEEs) could ensure privacy. To this direction, it comprises a *resource manager* to keep track of available resources and dynamic load of devices and a *scheduler* to allocate jobs. Each *device* maintains its own *queue*, with preemptible tasks based on their relative priority. To facilitate resource-to-task matching, a *performance controller* assesses an AI-tasks's runtime on a certain device through analytical or historical estimators.

Upon scheduling a training or inference task, the orchestrator tracks its execution; the respective *controller* monitors the task progress, and handles data and model access between devices or even parameter aggregation, if applicable. This way, context sharing and collaborative learning are enabled in a robust manner, while access to sensitive data remains controlled.

EdgeAI-Hub

Fig. 4b showcases the technology stack of an EdgeAI-Hub. At the bottom, we have the hardware that is responsible for running the AI-tasks. The underlying SoC would integrate general-purpose and specialised hardware, *i.e.* an NPU, to accelerate common DNN operations, from convolutional and fully-connected layers to Transformer blocks [3]. Contrary to smartphone SoCs nowadays, EdgeAI-Hub hardware should be optimised for *i) a broader set of operations*, *ii) multi-precision support*, *iii) sparsity and dynamic input length support*, *iv)* large *multi-level memory* for training with Direct Memory Access (DMA) support for zero-copy distributed ML, *v) virtualisation* for fast context switching and task preemption under multi-user tenancy, and *vi)* hardware-based *app sandboxing* with TEE [13] for secure processing of sensitive data.

Network-wise, we envision the EdgeAI-Hub to support simultaneous communication over multiple interfaces, both for supporting different devices as well as for load-balancing and higher throughput. Multiple-Input Multiple-Output (MIMO) over multiple antennas could increase communication capacity, along with mesh networking over proxy repeaters for densely covering the deployment setting. Last, device discovery and handshaking should be made possible over any supported channel, *e.g.* BLE, NFC, UWB. Extensibility through removable components could also future-proof the EdgeAI-Hub's hardware and trade energy efficiency for additional bandwidth.

Data are a first-class citizen in AI and, thus, storage becomes important. We propose a hierarchical storage solution with fast caching for iterative operations, like model training or model/context sharing, and traditional drives for long-term persistence. Hardware-level encryption and user-management with Access Control List (ACL) support from the filesystem is important to ensure security and privacy of data. Finally, redundancy could be offered through hardware (*e.g.* RAID) or distributed replication.

Moving up, the operating system (OS) is responsible for local resource management and task scheduling. Low-level ML compilers and BLAS libraries would reside on top of the OS, along with the high-level ML framework interpreters for DNN execution.

The middleware and application layers form the top of the stack. The former is responsible for the coordination of distributed execution of tasks, including the orchestrator, preemptive scheduler and performance monitor of local and remote resources. The application may then span across different use-cases from the following section.

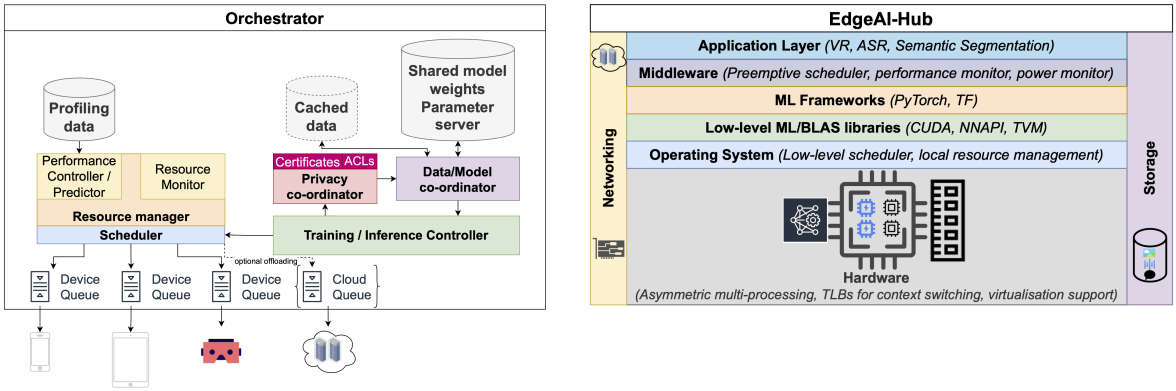

FIGURE 4: Orchestrator reference design (4a) and EdgeAI-Hub reference stack (4b).

Enabling Upcoming Use-Cases

There are various existent or upcoming applications that can benefit from the proposed paradigm.

Virtual assistants & enhanced interaction. Home assistants have been progressively penetrating the consumer-edge. However, they largely remain thinly-provisioned, with most computation happening on the cloud. The proliferation of LLMs that could be – at least partially – hosted on premises could significantly improve privacy, but also enable new use-cases of general or context- and user-specific question answering. Additionally, these models can be combined with other modalities (*e.g.* vision) and repurposed for more complex tasks in the context of smart homes.

Virtual spaces. AR/VR technologies have been undoubtedly getting more capable (*e.g.* Apple Vision Pro) and the promise of virtual worlds like the Metaverse increasingly relevant. Furthermore, the importance of online communication has been further enhanced by the COVID-19 pandemic and alternative models of collaboration, such as telepresence, have emerged. As the physical world enters virtual reality, ML plays a central role and new architectures for scene representation or object mapping (*e.g.* NERF) are utilised, requiring significant computational power and energy to remain mobile. Preparing the consumer ecosystem for such workloads via low-latency multi-device collaborative inference [15] can be the enabler for such next-gen technologies, be it for multiplayer VR gaming, hybrid collaboration or immersive entertainment.

Asynchronous collaboration. Maintaining a work-life balance when working from home can be difficult, especially when collaborating in international and multi-timezone settings. Therefore, tools enabling queryable meeting and document summarisation or multilingual translation (e.g. Microsoft Office co-pilot), along with creative writing or coding assistants (e.g. Github co-pilot) can be valuable tools for asynchronous collaboration. With privacy being a major business requirement, running such models locally becomes crucial. By locally offloading computation to trusted devices, this can be realistically implemented.

Cyber-physical agents. Inversely, computer agents have started having physical presence. Robotic agents, in the form of vacuum cleaners, assistants for the elderly or multi-articulated arms for cooking have appeared and push the envelope of what is possible. In this realm, agents may co-exist with one another or with other sensors that can enhance their perception and spatial awareness. Moreover, mobile robots can act as active learning agents for gathering knowledge and reducing model uncertainty.

e-Health. Edge AI has the potential to democratise healthcare by making services available at the point-of-care via automated or assisted diagnostics and treatments [6], without the need to send patient data to the cloud. Be it through automated diagnostics or treatments, over AI-aided consultations and care, Edge-AI-driven services can improve efficiency and privacy while lowering risk and cost. Additionally, wearables allow for real-time health and fitness feedback and, coupled with external sensors at home, can provide insights and alerts for health issues in a multi-sensory manner. This can be complemented with digital gastronomy tools for more complete well-being assistants.

Challenges

The advent of such new consumer ecosystems brings about several challenges. These remain largely open issues and we anticipate them becoming prevalent

topics of future research.

ML-related challenges

Non-IID data, tasks and annotations. In-the-wild data are highly heterogeneous across clients and may temporally evolve. As such, techniques from domain adaptation, FL and Continual Learning become increasingly relevant to generalisation. Moreover, local data available for training may not be "clean" or annotated with labels, thus paving the way for alternative methods, including semi-supervised, unsupervised and active learning for generalisation.

Model co-habitation. Nowadays smart-devices (*e.g.* smartphones) perform a multitude of tasks concurrently. As such, several DNNs may be cohabitating the device running in parallel. How these are scheduled to be executed efficiently is an open problem [19].

Utility & privacy. FL is becoming a prevalent paradigm for pushing training to the device, the custodian of data. However, there is a trade-off between global and private utility, as optimisation goals can clash. Adding differential privacy noise to the updates adds another degree of freedom, trading utility for privacy. Striking a balance between all these factors can be challenging.

Confidence assessment. As AI-tasks progressively make their way into the physical world, through robotic assistants, it is necessary to not only make decisions in a binary manner, but also assess the confidence of their decisions before acting, both for safety and interpretability.

System-level challenges

System heterogeneity and availability. Contrary to the datacenter, devices in the wild can be very heterogeneous and may arbitrarily become unavailable, due to mobility, power or network connectivity. As such, they should be treated accordingly, as this can affect the overall performance, robustness or even fairness of the (eco)system.

Multi-channel networking. The network interconnect is a core component in the proposed paradigm, acting as a multi-dimensional bus among devices. This interconnect is largely wireless, potentially lossy and heterogeneous in its implementation and qualitative characteristics. Thus, careful task scheduling and channel load balancing is important for ensuring best QoE.

Fault-tolerance & task preemption. As resources may become unavailable, high-priority tasks should be given fault-tolerance margins or be restarted on available resources in a transparent-to-the-user manner.

Interoperability. Another repercussion to extreme heterogeneity is that devices of different manufacturers, generations and tiers will need to co-operate through a common interconnect. The same applies to networking interoperability, which can become a bottleneck across generations of devices. As such, backwards compatibility, standardisation of interfaces and open middleware become requirements to break away from silos and enable device federation.

Incentives. An equally important factor for realisation of such a paradigm is incentivisation of the stakeholders: *i)* Manufacturers to support federated operation of devices and unlocking the potential of old devices; *ii)* Clients to accept on-premise execution of AI-tasks that can potentially benefit more users (*e.g.* FL).

Market transcendence. It is unlikely that many consumers will invest on an expensive device ecosystem at once. Therefore, it is valuable to consider the consumer market in a stateful manner, where devices pre-exist and come up with a way of gradual transcendence. An example could be firstly co-locating AI-Hubs on premium high-end devices (*e.g.* TVs) that progressively get marketed as modular components and standalone devices. This also means that connected devices can gradually get cheaper since they do not need to overprovision for bespoke hardware.

Conclusions

We have just started witnessing the revolutionary capabilities of the latest generation of ML models. However, their resource demands are beyond what the current consumer edge can sustain, making them accessible to only a few. In this article, we envision a new Edge-AI paradigm, built around EdgeAI-Hubs. These hubs do not only enable complex, resource-intensive applications to run at the consumer edge, but also prioritise user-privacy. To achieve this, we have laid the groundwork with a cross-layer blueprint of this architecture and pinpointed the challenges that lie ahead. Ultimately, our work serves as a foundation to a new, more collaborative model in consumer electronics that can set the stage for how our devices operate in the future.

REFERENCES

1. Mario Almeida, Stefanos Laskaridis, Abhinav Mehrotra, Lukasz Dudziak, Ilias Leontiadis, and Nicholas D. Lane. Smart at What Cost? Characterising Mobile Deep Neural Networks in the Wild. In *ACM Internet Measurement Conference (IMC)*, 2021.

2. Shuping Dang, Osama Amin, Basem Shihada, and Mohamed-Slim Alouini. What should 6g be? *Nature Electronics*, 3(1):20–29, 2020.
3. Hongxiang Fan, Thomas Chau, Stylianos I. Venieris, Royson Lee, Alexandros Kouris, Wayne Luk, Nicholas D Lane, and Mohamed S. Abdelfattah. Adaptable Butterfly Accelerator for Attention-based NNs via Hardware and Algorithm Co-design. In *International Symposium on Microarchitecture (MICRO)*, 2022.
4. In Gim et al. Memory-Efficient DNN Training on Mobile Devices. In *International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2022.
5. Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos I. Venieris, and Nicholas D. Lane. FjORD: Fair and Accurate Federated Learning under heterogeneous targets with Ordered Dropout. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
6. Nourah Janbi et al. Imtidad: A reference architecture and a case study on developing distributed ai services for skin disease diagnosis over cloud, fog and edge. *Sensors*, 22(5):1854, 2022.
7. Jun-Woo Jang et al. Sparsity-Aware and Reconfigurable NPU Architecture for Samsung Flagship Mobile SoC. In *International Symposium on Computer Architecture (ISCA)*, 2021.
8. Alexandros Kouris, Stylianos I Venieris, Stefanos Laskaridis, and Nicholas Lane. Multi-Exit Semantic Segmentation Networks. In *ECCV*, 2022.
9. Stefanos Laskaridis et al. Adaptive Inference through Early-Exit Networks: Design, Challenges and Directions. In *International Workshop on Embedded and Mobile Deep Learning (EMDL)*, 2021.
10. Stefanos Laskaridis, Kleomenis Kateveas, Lorenzo Minto, and Hamed Haddadi. Melting point: Mobile evaluation of language transformers. *arXiv preprint arXiv:2403.12844*, 2024.
11. Stefanos Laskaridis, Stylianos I. Venieris, Mario Almeida, Ilias Leontiadis, and Nicholas D. Lane. SPINN: Synergistic Progressive Inference of Neural Networks over Device and Cloud. In *International Conference on Mobile Computing and Networking (MobiCom)*, 2020.
12. H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning Differentially Private Recurrent Language Models. In *International Conference on Learning Representations (ICLR)*, 2018.
13. Fan Mo, Ali Shahin Shamsabadi, Kleomenis Katevas, Soteris Demetriou, Ilias Leontiadis, Andrea Cavallaro, and Hamed Haddadi. DarkneTZ: Towards Model Privacy at the Edge Using Trusted Execution Environments. In *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2020.
14. David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R So, Maud Texier, and Jeff Dean. The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink. *Computer*, 2022.
15. Ella Peltonen, Ijaz Ahmad, Atakan Aral, Michele Capobianco, Aaron Yi Ding, Felipe Gil-Castiñeira, Ekaterina Gilman, Erkki Harjula, Marko Jurmu, Teemu Karvonen, Markus Kelanti, Teemu Leppänen, Lauri Lovén, Tommi Mikkonen, Nitinder Mohan, Petteri Nurmi, Susanna Pirttikangas, Paweł Sroka, Sasu Tarkoma, and Tingting Yang. The many faces of edge intelligence. *IEEE Access*, 10:104769–104782, 2022.
16. Muhammad Shafique, Theocharis Theocharides, Vijay Janapa Reddy, and Boris Murmann. Tinyml: Current progress, research challenges, and future roadmap. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 1303–1306, 2021.
17. Mohammad Shahrad et al. Towards Deploying Decommissioned Mobile Devices as Cheap Energy-Efficient Compute Nodes. In *USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 17)*, 2017.
18. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
19. Stylianos I. Venieris et al. Multi-DNN Accelerators for Next-Generation AI Systems. *Computer*, 2023.
20. Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P Jue. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture*, 98:289–330, 2019.

Stefanos Laskaridis is a Researcher at Brave and visiting researcher at the University of Cambridge. He co-authored the paper while he was at Samsung AI Center in Cambridge, UK (SAIC-C). His research interests include distributed and efficient on-device ML. He received his M.Phil. degree from the University of Cambridge. He is a member of ACM. Contact at mail@stefanos.cc.

Stylianos I. Venieris is a Researcher at SAIC-C, where he leads the Distributed-AI group. His research interests include on-device deep learning for mobile, embedded or custom hardware. He received

the Ph.D. degree from Imperial College London, UK. He is a Member of IEEE and ACM. Contact at s.venieris@samsung.com.

Alexandros Kouris is a Researcher at SAIC-C. His research interests include ML, Embedded Systems and Robotics. He received his Ph.D. degree from Imperial College London, UK. He is a memeber of IEEE. Contact at a.kouris@samsung.com.

Rui Li is a Researcher at SAIC-C. Her research interests include wireless communications and ML. She received her Ph.D degree from the University of Edinburgh. Contact at rui.li@samsung.com.

Nicholas D. Lane is a Professor in the Department of Computer Science and Technology, University of Cambridge and the co-founder and CSO of Flower Labs. His research interest include distributed and on-device ML Systems. He co-authored the paper while he was at SAIC-C. He received his Ph.D. degree from Dartmouth College. Contact at ndl32@cam.ac.uk.