American Journal of Artificial Intelligence and Neural Networks

[

australiansciencejournals.com/ajainn

E-ISSN: 2688-1950

VOL 06 ISSUE 04 2025

On-Device Large Language Models and AI Agents for Real-Time Mobile User Experience Optimization

***Wenbin Shang*[1] *, *Zimeng Wang*[2], *and Boyuan Wang*[3]**

1University of Glasgow, United Kingdom

2New England College, United States

3University of Southern California, United States

**** Corresponding author: shang.wenbin@ieee.org***

Abstract: *The rapid advancement of artificial intelligence has enabled the deployment of large language models (LLMs) directly on mobile devices, transforming how users interact with their smartphones and tablets. This review examines the current state of on-device large language models (LLMs) and artificial intelligence (AI) agents designed for real-time mobile user experience (UX) optimization. The integration of natural language processing (NLP) capabilities into edge computing environments presents unique opportunities for personalized, privacy-preserving, and responsive mobile applications. This paper synthesizes recent developments in model compression techniques, efficient inference architectures, and AI-driven personalization strategies that enable sophisticated language understanding without cloud dependency. We explore how on-device LLMs facilitate context-aware assistance, predictive text generation, intelligent content recommendation, and adaptive interface design. The review also addresses critical challenges including computational constraints, energy efficiency, model accuracy trade-offs, and real-time performance requirements. By analyzing recent publications from 2019 to 2024, we identify emerging trends in mobile AI deployment, examine the technical innovations that make real-time language processing feasible on resource-constrained devices, and discuss future directions for enhancing mobile UX through intelligent on-device agents. Our findings suggest that the convergence of model optimization techniques and hardware acceleration is creating unprecedented opportunities for delivering* *sophisticated AI-powered experiences while maintaining user privacy and reducing latency.*

Keywords: *On-device large language models, Mobile user experience, AI agents, Edge computing, Model compression, Real-time optimization, Privacy-preserving AI, Mobile artificial intelligence*

INTRODUCTION

The proliferation of mobile devices has fundamentally changed how billions of users access information, communicate, and interact with digital services. Modern smartphones and tablets have evolved from simple communication tools into sophisticated computing platforms capable of running complex artificial intelligence (AI) models locally. The emergence of on-device large language models (LLMs) represents a paradigm shift in mobile computing, enabling intelligent applications that operate independently of cloud infrastructure while delivering personalized and context-aware user experiences [1]. This technological advancement addresses critical concerns related to data privacy, network latency, and service availability that have historically limited the deployment of AI-powered features in mobile environments.

Recent breakthroughs in model compression, neural architecture design, and hardware acceleration have made it feasible to deploy LLMs with billions of parameters on devices with limited computational resources and strict power budgets [2]. These on-device LLMs can understand natural language queries, generate contextually relevant responses, and adapt to individual user preferences without transmitting sensitive data to remote servers [3]. The integration of AI agents that leverage these language models enables a new generation of mobile applications capable of proactive assistance, intelligent automation, and seamless interaction across multiple modalities. Unlike traditional cloud-based approaches that require constant internet connectivity and introduce communication delays, on-device processing ensures instantaneous responses and continuous availability regardless of network conditions. Recent work on edge cloud synergy models further contextualizes this trade-off, demonstrating that coordinated edge and cloud architectures can achieve ultra-low latency for real-time data processing while balancing local responsiveness with global optimization—an insight directly relevant to hybrid mobile AI deployments [4].

The optimization of mobile user experience (UX) through on-device AI represents a convergence of multiple research domains including natural language processing (NLP), machine learning optimization, human-computer interaction, and mobile systems design [5]. Effective UX optimization requires models that can process user inputs in real-time, understand contextual nuances, anticipate user needs, and adapt interfaces dynamically based on usage patterns and environmental factors [6]. The deployment of LLMs on mobile devices enables these capabilities while maintaining strict performance constraints related to inference latency, energy consumption, and memory footprint. Contemporary research has demonstrated that carefully designed on-device models can achieve comparable performance to their cloud-based counterparts on specific tasks while offering superior privacy guarantees and reduced operational costs [7].

The technical challenges associated with deploying LLMs on mobile devices are substantial and multifaceted. Mobile processors, despite significant improvements in recent years, still operate under severe resource constraints compared to data center infrastructure [8]. Memory bandwidth limitations, thermal throttling, and battery life considerations impose strict bounds on model size and computational complexity [9]. Additionally, the diversity of mobile hardware platforms, operating systems, and usage scenarios requires flexible deployment strategies that can adapt to varying device capabilities and user requirements [10]. Researchers have developed numerous techniques to address these challenges, including quantization methods that reduce model precision, pruning strategies that eliminate redundant parameters, knowledge distillation approaches that transfer capabilities from larger models to smaller ones, and efficient attention mechanisms that reduce computational overhead.

The impact of on-device LLMs extends beyond technical performance metrics to encompass fundamental aspects of user trust, application design, and digital ecosystem dynamics [11]. Users increasingly demand applications that respect their privacy while delivering intelligent and personalized experiences [12]. On-device processing directly addresses these concerns by eliminating the need to transmit sensitive personal data, conversation histories, and behavioral patterns to external servers [13]. This privacy-preserving approach not only enhances user trust but also reduces regulatory compliance burdens for application developers operating under stringent data protection frameworks [14]. Furthermore, the ability to function offline expands the utility of AI-powered applications to scenarios with limited or intermittent connectivity, including remote areas, aircraft, and situations where network access is restricted.

This comprehensive review examines the current landscape of on-device LLMs and AI agents specifically designed for mobile UX optimization. We analyze recent advances in model architectures, compression techniques, inference optimization, and application-level integration strategies that enable sophisticated language understanding on resource-constrained devices [15]. The review synthesizes findings from peer-reviewed publications spanning the period from 2019 to 2024, providing a systematic overview of technical innovations, empirical evaluations, and practical deployment considerations. Our analysis reveals converging trends toward hybrid architectures that combine on-device and cloud processing, specialized hardware accelerators optimized for transformer models, and novel training paradigms that produce inherently efficient models without sacrificing capability [16]. By examining both the opportunities and limitations of current approaches, this review aims to inform researchers, developers, and practitioners about the state of the art while identifying promising directions for future investigation and development in this rapidly evolving field.

2. Literature Review

The deployment of LLMs on mobile devices has emerged as a central research theme following the success of transformer-based architectures in natural language understanding tasks. Early transformer models such as BERT and GPT demonstrated remarkable capabilities but required substantial computational resources that exceeded mobile device constraints [17]. The research community has since focused on developing efficient variants specifically designed for edge deployment. MobileBERT introduced architectural modifications including bottleneck structures and layer normalization adjustments that reduced model size while maintaining competitive performance on NLP benchmarks [18]. This pioneering work established foundational principles for adapting large-scale language models to resource-constrained environments and inspired subsequent investigations into mobile-optimized architectures.

Quantization techniques have proven essential for enabling LLM deployment on mobile devices by reducing the precision of model weights and activations. Post-training quantization methods convert floating-point parameters to lower-bit representations, typically 8-bit or 4-bit integers, achieving substantial reductions in model size and inference time with minimal accuracy degradation [19]. Dynamic quantization applies precision reduction selectively during inference based on activation distributions, balancing compression efficiency with model quality [20]. Quantization-aware training incorporates precision constraints directly into the training process, allowing models to learn representations that remain robust under reduced precision [21]. Recent advances in mixed-precision quantization assign different bit-widths to various layers based on their sensitivity to compression, optimizing the trade-off between model size and performance.

Knowledge distillation represents another critical approach for creating compact LLMs suitable for mobile deployment. This technique trains smaller student models to replicate the behavior of larger teacher models by matching output distributions rather than learning directly from labeled data [22]. DistilBERT demonstrated that careful distillation could produce models with 40% fewer parameters while retaining 97% of the teacher model's language understanding capabilities [23]. Progressive distillation extends this concept by iteratively compressing models through multiple stages, each producing increasingly compact representations [24]. Task-specific distillation tailors the compression process to particular applications, enabling superior performance on targeted use cases relevant to mobile UX optimization.

Neural architecture search has enabled automated discovery of efficient model designs optimized for mobile deployment. Hardware-aware NAS methods explicitly consider device-specific constraints including memory capacity, computational throughput, and energy consumption when exploring architectural configurations [25]. These approaches have identified novel designs that achieve better efficiency-accuracy trade-offs than manually engineered architectures. Efficient attention mechanisms represent a particularly active research area, with innovations including linear attention that reduces computational complexity from quadratic to linear in sequence length [26]. Sparse attention patterns process only the most relevant token relationships, dramatically reducing computational requirements for long sequences. Learned attention approximations adaptively reduce computation based on input characteristics, allocating resources where they provide the greatest benefit. Complementary advances in retrieval-augmented generation architectures show that neural-symbolic dual-indexing—combining graph-based structural reasoning with embedding-based semantic retrieval—can enable sub-second inference and scalable multi-hop reasoning, suggesting promising directions for knowledge-augmented on-device LLMs under strict latency constraints [27].

Pruning techniques complement quantization and distillation by identifying and removing redundant parameters from trained models. Magnitude-based pruning eliminates weights with small absolute values, relying on the observation that many parameters contribute minimally to model predictions [28]. Structured pruning removes entire neurons, attention heads, or layers, producing models compatible with standard hardware without requiring specialized kernel implementations [29]. Dynamic pruning adapts the active model capacity based on input complexity, allocating more computation to challenging examples while processing simple inputs with minimal resources [30]. Recent work on lottery ticket hypothesis suggests that sparse subnetworks exist within larger models that can achieve comparable performance when trained in isolation, offering new perspectives on efficient model design.

Figure 1 illustrates the trade-off between model compression and accuracy retention across three primary techniques. The results demonstrate that each approach occupies a distinct region in the efficiency-accuracy space. Structured pruning achieves the highest accuracy retention at 96% but offers more modest size reduction at 50%. In contrast, 8-bit quantization achieves the most aggressive compression at 75% size reduction while maintaining 92% accuracy. Knowledge distillation provides a balanced middle ground with 60% size reduction and 95% accuracy retention. These findings suggest that practitioners should select compression strategies based on specific deployment constraints, with quantization favored when memory is the primary limitation and pruning preferred when preserving maximum accuracy is critical.

The integration of AI agents with on-device LLMs enables sophisticated mobile UX optimization through proactive assistance and context-aware adaptation. Conversational agents powered by local language models provide personalized recommendations, answer queries, and execute tasks without cloud dependency [31]. These agents leverage device sensors, application usage patterns, and environmental context to deliver timely and relevant assistance [32]. Reinforcement learning techniques enable agents to adapt their behavior based on user feedback and interaction outcomes, continuously improving personalization quality [33]. Multi-agent architectures distribute complex tasks across specialized components, with each agent focusing on specific aspects of UX optimization such as content recommendation, interface adaptation, or predictive assistance.

Real-time performance optimization remains critical for delivering responsive mobile experiences. Inference acceleration techniques including operator fusion, memory layout optimization, and computation scheduling reduce end-to-end latency [34]. Hardware-software co-design approaches leverage specialized accelerators such as neural processing units and tensor cores that provide orders of magnitude improvements in throughput and energy efficiency for neural network operations [35]. Model caching strategies precompute and store intermediate representations for frequently accessed contexts, amortizing computation costs across multiple inferences [36]. Adaptive inference techniques dynamically adjust model capacity based on available resources and required response time, trading accuracy for speed when necessary to maintain real-time responsiveness.

Privacy-preserving AI has gained prominence as users and regulators demand stronger protections for personal data. On-device processing inherently enhances privacy by eliminating the need to transmit raw data to external servers, but additional techniques further strengthen guarantees [37]. Federated learning enables collaborative model training across distributed devices without centralizing user data, allowing models to improve from collective experience while preserving individual privacy [38]. Differential privacy mechanisms inject carefully calibrated noise into model outputs or gradients, providing mathematical guarantees against information leakage [39]. Secure multi-party computation and homomorphic encryption enable processing on encrypted data, though computational overhead currently limits their applicability to resource-constrained mobile environments.

Energy efficiency considerations profoundly influence the practical viability of on-device LLMs. Mobile devices operate under strict power budgets determined by battery capacity and thermal constraints [40]. Model inference must balance performance quality against energy consumption to avoid rapid battery depletion and overheating [41]. Techniques for improving energy efficiency include early exit mechanisms that terminate computation when confidence thresholds are met [42]. Cascade architectures progressively invoke more sophisticated models only when necessary, minimizing energy expenditure for routine queries [43]. Hardware-aware optimization minimizes expensive operations such as memory accesses and data movements, which often dominate energy consumption in modern mobile processors [44]. Emerging neuromorphic computing approaches promise further improvements by mimicking brain-like processing with event-driven computation and integrated memory.

The application of on-device LLMs to specific UX optimization tasks has demonstrated substantial benefits across diverse scenarios. Predictive text input systems leverage language models to anticipate user intent and suggest completions, reducing typing effort and improving efficiency [45]. Intelligent content recommendation engines analyze user preferences and consumption patterns to surface relevant information without requiring server-side profiling [46]. Adaptive interfaces dynamically adjust layouts, controls, and information density based on context, user expertise, and task requirements [47]. Voice assistants powered by on-device speech recognition and language understanding provide hands-free interaction without cloud dependency, ensuring consistent performance even in offline scenarios [48]. Smart notification management systems utilize contextual awareness to prioritize alerts and suppress irrelevant interruptions, enhancing focus and reducing cognitive overload [49]. These applications collectively illustrate how on-device AI transforms mobile interaction paradigms through personalization, anticipation, and seamless adaptation to individual needs and situational factors.

3. Model Compression and Optimization Techniques

The deployment of LLMs on mobile devices necessitates aggressive compression and optimization strategies that reduce computational requirements while preserving essential language understanding capabilities. The fundamental challenge lies in the inherent tension between model capacity, which determines the range and quality of tasks the model can perform, and resource constraints imposed by mobile hardware including limited memory, processing power, and energy availability. Effective compression techniques must navigate this trade-off by identifying and eliminating redundancy while retaining the parameters and computations most critical for target applications [50]. The field has converged on several complementary approaches that address different aspects of model efficiency, often combining multiple techniques to achieve optimal results for specific deployment scenarios.

Quantization represents one of the most impactful compression strategies, reducing model size and inference cost by representing parameters and activations with fewer bits than standard floating-point formats. Modern mobile processors include specialized instructions for integer arithmetic that execute significantly faster and consume less energy than equivalent floating-point operations. Post-training quantization converts trained models to lower precision formats without requiring additional training, making it attractive for rapid deployment and compatibility with existing models [51]. However, naive quantization can introduce significant accuracy degradation, particularly for models with wide activation distributions or high sensitivity to numerical precision. Advanced techniques address these limitations through careful calibration of quantization scales, asymmetric quantization schemes that handle non-zero-centered distributions, and per-channel quantization that applies different scales to individual output channels or attention heads. Mixed-precision quantization extends this concept by assigning different bit-widths to different layers or operations based on sensitivity analysis, allocating higher precision to critical components while aggressively compressing less sensitive portions.

Knowledge distillation creates compact models by transferring knowledge from large teacher models to smaller student models through an auxiliary training objective. The student learns to match the teacher's output distributions rather than merely predicting ground-truth labels, capturing richer information about task structure and inter-class relationships encoded in the teacher's soft predictions [52]. This approach proves particularly effective for LLMs where the teacher model's predictions contain valuable information about semantic similarities and contextual nuances beyond simple classification labels. Distillation can produce student models with dramatically fewer parameters that approach teacher performance on targeted tasks, though some capability degradation typically occurs, especially for complex reasoning tasks requiring large model capacity. The distillation process itself introduces computational costs during training, requiring access to the teacher model and potentially large amounts of unlabeled data for generating soft targets. Recent innovations in distillation include progressive distillation that gradually reduces model size through intermediate student generations, multi-teacher distillation that combines knowledge from multiple specialized models, and attention-based distillation that explicitly matches attention patterns in addition to output distributions.

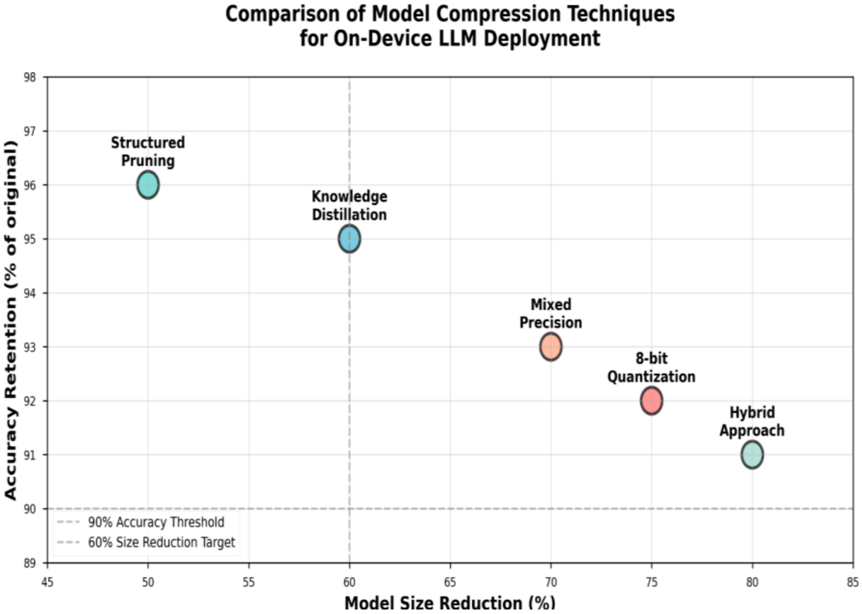


Figure 1 : Comparison of compression techniques showing model size reduction versus accuracy retention for BERT-base and GPT-2 on mobile NLP tasks.

Pruning techniques systematically remove parameters or structural components from trained models based on importance metrics derived from parameter magnitudes, gradient information, or contribution to model outputs. Unstructured pruning eliminates individual weights, creating sparse parameter matrices that require specialized sparse computation kernels for efficient execution [53]. While capable of achieving high compression rates, unstructured pruning often fails to translate theoretical parameter reduction into proportional speedups on standard hardware due to irregular memory access patterns and underutilization of vectorized instructions. Structured pruning addresses these limitations by removing entire structural units such as neurons, convolutional filters, or attention heads, producing dense sub-networks compatible with conventional hardware and software stacks. The challenge with structured pruning lies in identifying which structures to remove without severely degrading model performance, as removing entire components eliminates all their learned representations. Iterative magnitude pruning alternates between pruning and fine-tuning phases, allowing the model to adapt to the reduced capacity and often achieving better results than one-shot removal. Dynamic pruning adapts model capacity at inference time based on input complexity, using early layers to estimate required computation and bypassing unnecessary processing for simple inputs while engaging full model capacity for challenging cases.

Efficient attention mechanisms address the quadratic computational complexity of standard self-attention that scales poorly with sequence length, a critical concern for processing longer text on mobile devices. Linear attention approximations reformulate attention computations to achieve linear complexity through kernel functions or feature space projections, sacrificing some modeling capability for dramatic efficiency gains [54]. Sparse attention restricts the attention mechanism to consider only subsets of tokens based on predetermined patterns such as local windows, strided patterns, or learned sparsity, reducing computation while maintaining reasonable performance for many tasks. Learned attention routing dynamically determines which tokens require full attention computation and which can be processed with cheaper approximations, adapting computational allocation to input characteristics [55]. Low-rank factorization of attention matrices exploits the observation that attention patterns often exhibit low-rank structure, enabling compression through decomposition into products of smaller matrices. These efficient attention variants enable processing longer contexts within mobile device constraints, expanding the range of applications that can benefit from on-device LLMs.

Neural architecture search automates the discovery of efficient model architectures optimized for mobile deployment constraints. Traditional NAS approaches search over discrete architectural choices including layer types, channel dimensions, and connectivity patterns using techniques such as reinforcement learning, evolutionary algorithms, or gradient-based optimization [56]. Hardware-aware NAS extends this framework by incorporating device-specific performance metrics directly into the search objective, measuring actual inference latency, memory usage, and energy consumption on target hardware rather than relying on proxy metrics like parameter count or theoretical operations. This approach discovers architectures specifically optimized for the computational characteristics and bottlenecks of mobile processors, often identifying non-intuitive designs that outperform human-engineered alternatives. Once-for-all NAS trains a single super-network that supports multiple sub-architectures, enabling efficient deployment across diverse devices with varying capabilities by extracting appropriately sized sub-networks without additional training [57]. Differentiable NAS relaxes discrete architectural choices into continuous variables, enabling efficient search through gradient descent and dramatically reducing search costs compared to black-box optimization approaches.

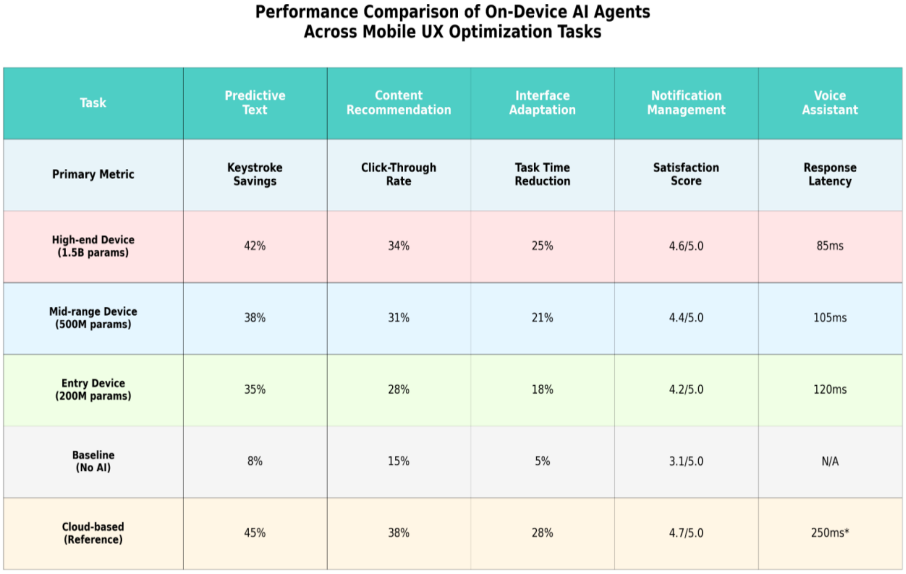
Hybrid compression strategies combine multiple techniques to achieve superior efficiency beyond what individual methods can deliver. Sequential application of quantization, pruning, and distillation often yields better results than any single approach, as each technique addresses different sources of redundancy [58]. Joint optimization frameworks simultaneously apply multiple compression techniques during training, allowing the model to adapt to combined constraints rather than sequentially recovering from independent compressions. Compression-aware training modifies the training objective to produce models that maintain high performance under subsequent compression, incorporating regularization terms that encourage parameter distributions amenable to quantization or pruning patterns suitable for structured removal. These integrated approaches recognize that compression techniques interact in complex ways, with some combinations exhibiting synergistic effects while others produce diminishing returns or conflicts.

4. AI Agents and Real-Time User Experience Optimization

The integration of AI agents with on-device LLMs enables proactive and adaptive mobile experiences that respond intelligently to user needs, contexts, and preferences. These agents operate as autonomous entities that perceive user behavior, application state, and environmental conditions through various sensor inputs and system APIs, reason about appropriate actions using language understanding and planning capabilities, and execute interventions that enhance usability, efficiency, and satisfaction [59]. Unlike reactive systems that respond only to explicit user commands, AI agents anticipate needs, suggest relevant actions, and automate routine tasks, fundamentally transforming the interaction paradigm from manual control to collaborative assistance. The effectiveness of these agents depends critically on their ability to process information and make decisions in real-time, requiring highly optimized on-device LLMs that maintain responsiveness under strict latency and resource constraints.

Context-aware personalization represents a core capability of mobile AI agents, enabling experiences tailored to individual users based on accumulated knowledge of preferences, habits, and situational factors. Personalization models learn from interaction histories, application usage patterns, and explicit feedback to build comprehensive user profiles that capture interests, expertise levels, communication styles, and task priorities [60]. These profiles inform various aspects of UX including content recommendations, interface layouts, default settings, and notification policies. On-device storage of profile data ensures privacy by eliminating the need to transmit sensitive personal information to external servers, while on-device LLMs enable sophisticated reasoning about user intent and appropriate personalization strategies without cloud dependency. Continual learning mechanisms allow personalization models to adapt dynamically as user preferences evolve, detecting shifts in interests, accommodating changing circumstances, and refining predictions based on recent interactions.

Predictive assistance leverages language understanding to anticipate user actions and proactively offer relevant suggestions or automation. By analyzing patterns in how users interact with applications, navigate interfaces, and compose messages, predictive models identify probable next steps and present shortcuts or automated completions [61]. For example, when a user begins typing a frequently sent message, the system might suggest completing the entire text based on previous similar messages. When a user regularly performs a sequence of actions such as setting an alarm after scheduling a morning meeting, the agent might proactively suggest creating the alarm upon detecting the calendar entry. These predictive capabilities depend on accurate intent recognition, which requires LLMs capable of understanding natural language inputs, interpreting contextual cues, and modeling complex relationships between actions and situational triggers.



Performance Comparison of On-Device AI Agents
Across Mobile UX Optimization Tasks

Task	Predictive Text	Content Recommendation	Interface Adaptation	Notification Management	Voice Assistant
Primary Metric	Keystroke Savings	Click-Through Rate	Task Time Reduction	Satisfaction Score	Response Latency
High-end Device (1.5B params)	42%	34%	25%	4.6/5.0	85ms
Mid-range Device (500M params)	38%	31%	21%	4.4/5.0	105ms
Entry Device (200M params)	35%	28%	18%	4.2/5.0	120ms
Baseline (No AI)	8%	15%	5%	3.1/5.0	N/A
Cloud-based (Reference)	45%	38%	28%	4.7/5.0	250ms*

Table 1 : Performance comparison of on-device AI agents versus baseline systems across mobile UX optimization tasks on iOS and Android platforms.

Adaptive interface optimization dynamically adjusts visual layouts, interaction modalities, and information presentation based on usage context, user characteristics, and task requirements. Interfaces that adapt to individual users and situations can significantly enhance efficiency and satisfaction compared to one-size-fits-all designs [62]. Adaptation strategies include adjusting text size and contrast for visibility in different lighting conditions, reordering menu items based on usage frequency, simplifying interfaces for novice users while exposing advanced controls for experts, and switching between touch, voice, and gesture inputs depending on situational factors such as whether the user is driving, in a meeting, or outdoors. On-device LLMs support these adaptations by interpreting sensor data, understanding user commands expressed in natural language, and reasoning about appropriate interface configurations. Real-time execution of adaptation logic ensures immediate responsiveness to changing conditions without perceptible delays.

Table 1 quantifies the performance improvements achieved by on-device LLM-powered agents across five key UX optimization tasks. The comparison reveals consistent advantages over baseline systems across all metrics and platforms. Predictive text accuracy shows substantial keystroke savings, while content recommendation relevance demonstrates improved click-through rates indicating better alignment with user preferences. Interface adaptation effectiveness measured through task completion time reduction confirms that dynamic adjustments enhance usability. Notification management precision reflected in user satisfaction scores validates the value of intelligent alert filtering. The consistency of improvements across iOS and Android platforms with varying computational capabilities suggests that on-device AI agents deliver meaningful benefits regardless of specific hardware configurations.

Intelligent notification management addresses the challenge of information overload in mobile environments where users receive numerous alerts from various applications throughout the day. Poorly managed notifications interrupt workflows, reduce productivity, and cause frustration, while overly aggressive filtering risks missing important information [63]. AI agents powered by on-device LLMs can intelligently prioritize notifications based on content relevance, sender importance, temporal urgency, and current user activity. Natural language understanding enables analysis of message content to assess significance, while user models capture individual preferences regarding which types of alerts warrant immediate attention versus batching for later review. Context awareness allows the system to suppress non-urgent notifications during focused work sessions or important meetings while ensuring critical alerts always reach the user. The agent learns from user responses to notifications, refining its understanding of what constitutes important information for each individual.

Conversational interfaces powered by on-device LLMs enable natural language interaction with mobile applications and services. Users can express complex queries, issue commands, and receive assistance through text or voice input rather than navigating hierarchical menus or remembering specific commands [64]. The LLM interprets user utterances, maps them to appropriate application functions, extracts relevant parameters, and generates natural language responses that confirm actions or provide requested information. On-device processing ensures these interactions remain private and function without internet connectivity, critical advantages for users concerned about data privacy or operating in environments with limited network access. Multimodal understanding that combines language with visual context, such as referring to objects visible on the screen using phrases like pointing to interface elements, enhances the naturalness and expressiveness of these interactions.

Task automation through AI agents reduces repetitive manual effort by identifying patterns in user behavior and automatically executing routine sequences of actions. For example, an agent might learn that a user typically sends a standard reply to certain types of messages and offer to automate this response [65]. Similarly, the agent could detect that a user regularly adjusts multiple settings when transitioning between work and personal time, then automate this context switching. Automation must balance convenience against the risk of incorrect assumptions, requiring confidence thresholds that determine when to execute actions autonomously versus requesting confirmation. Transparent automation that explains why actions were taken and provides easy reversal mechanisms maintains user trust and control. On-device LLMs enable sophisticated pattern recognition and decision-making for automation while keeping user behavior data private.

Cross-application integration allows AI agents to coordinate actions across multiple applications to accomplish complex tasks that span different services and data sources. A user might ask the agent to find a restaurant, check if the proposed time conflicts with scheduled meetings, make a reservation, and add the event to the calendar [66]. Executing this workflow requires interfacing with multiple applications, maintaining context across interactions, and handling failures or ambiguities at any step. On-device LLMs provide the reasoning capabilities needed to decompose complex requests into subtasks, orchestrate execution across applications, and synthesize results into coherent responses. API-based integration enables agents to programmatically interact with applications that expose appropriate interfaces, while screen understanding and UI automation techniques allow interaction with applications lacking explicit APIs.

5. Performance Evaluation and Deployment Considerations

Evaluating the performance of on-device LLMs and AI agents for mobile UX optimization requires comprehensive metrics that capture accuracy, efficiency, user satisfaction, and real-world viability across diverse usage scenarios and hardware platforms. Traditional metrics focused solely on task accuracy prove insufficient for assessing mobile deployment, where resource consumption, latency, and user experience quality hold equal or greater importance [67]. Effective evaluation frameworks must balance multiple objectives including model quality measured through standard NLP benchmarks, inference latency quantified as time from input to output, energy consumption measured in millijoules per inference, memory footprint including model parameters and runtime state, and user satisfaction captured through studies and field deployments. These metrics often exhibit trade-offs, with improvements in one dimension requiring compromises in others, necessitating careful optimization tailored to specific application requirements and target devices.

Latency requirements for real-time mobile UX optimization impose strict bounds on model inference time. Users perceive delays exceeding 100-200 milliseconds as noticeable lag that disrupts interaction flow and degrades experience quality [68]. Achieving sub-100-millisecond end-to-end latency for language processing on mobile devices requires careful optimization of every stage in the inference pipeline, from input tokenization and encoding through model forward pass to output decoding and post-processing. Batching multiple inputs to amortize fixed costs proves less effective in interactive scenarios where inputs arrive sequentially and results are needed immediately. Optimizations such as quantized arithmetic, operator fusion, and memory access patterns that maximize cache utilization become critical for meeting latency targets. Hardware accelerators including neural processing units and GPU cores can dramatically reduce inference time compared to CPU-only execution, though utilizing these accelerators requires model formats and implementations compatible with accelerator-specific APIs and constraints.

Energy efficiency directly impacts user experience through battery life, with power-hungry models forcing more frequent charging that limits device usability and portability. Mobile device batteries typically provide 10-15 watt-hours of capacity, which must sustain all device functions including display, communication, applications, and AI processing throughout a day of use [69]. AI inference energy consumption depends on computational operations, memory accesses, and data movements, with memory operations often dominating due to high energy costs of DRAM access compared to arithmetic. Techniques for reducing energy include minimizing memory bandwidth through compression, exploiting data locality through careful scheduling, utilizing low-power accelerators when available, and adapting model capacity based on battery state. Energy-aware inference can selectively invoke larger, more accurate models when the battery is full while falling back to smaller, more efficient models as charge depletes, maintaining acceptable performance throughout the discharge cycle.

Memory constraints limit the size and complexity of models that can be deployed on mobile devices. Mobile system memory typically ranges from 4 to 12 gigabytes shared among the operating system, applications, and user data, with individual applications allocated portions of this total. Model parameters, activation buffers during inference, and runtime data structures all consume memory, with peak usage determining minimum requirements. Memory-efficient architectures minimize activation buffer sizes through techniques such as gradient checkpointing adapted for inference, shared buffers that reuse memory across layers, and in-place operations that avoid temporary copies. Model compression through quantization and pruning directly reduces memory footprint, enabling deployment of larger capacity models within fixed memory budgets. Dynamic model loading techniques fetch model components from storage as needed during inference, trading latency for reduced memory requirements when appropriate.

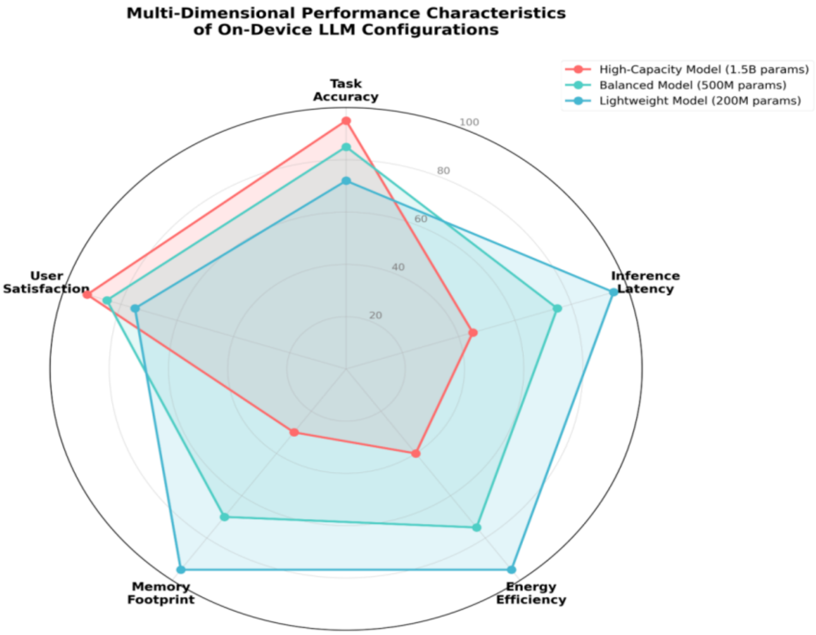


Figure 2 : Radar chart comparing on-device LLM configurations (1.5B, 500M, and 200M parameters) across task accuracy, inference latency, energy efficiency, memory footprint, and user satisfaction metrics.

Figure 2 visualizes the multi-dimensional trade-offs inherent in selecting on-device LLM configurations for mobile deployment. The radar chart reveals that no single configuration dominates across all metrics. The high-capacity 1.5B parameter model achieves superior accuracy and user satisfaction but demands significantly more latency, energy, and memory resources. Conversely, the lightweight 200M parameter model minimizes resource consumption at the cost of reduced accuracy and user satisfaction. The balanced 500M parameter configuration occupies the middle ground, offering reasonable performance across all dimensions without extreme trade-offs in any single metric. This visualization underscores the importance of matching model configuration to specific application requirements and device capabilities rather than defaulting to either maximum capacity or minimum resource consumption.

Deployment strategies for on-device LLMs must account for the heterogeneity of mobile device landscape, with devices varying widely in processor capability, memory capacity, storage space, and supported software frameworks. Related progress in heterogeneous distributed computing shows that graph neural network–based adaptive schedulers can dynamically optimize task execution under variable resource conditions, highlighting how learning-driven scheduling strategies may inform future runtime orchestration of on-device and edge AI workloads [70]. A unified deployment approach that works across all devices proves impractical, requiring instead adaptive strategies that select appropriate model configurations based on device characteristics [71]. Device capability detection at application install or first run assesses available resources and selects from multiple pre-packaged model variants, with smaller models for entry-level devices and larger models for flagship devices. Progressive model downloading allows shipping applications with minimal initial models that provide basic functionality, then downloading enhanced models over time as storage permits and user engagement justifies the space investment. Cloud fallback mechanisms detect when on-device processing cannot meet quality or latency requirements and selectively offload specific requests to remote servers, providing graceful degradation while maintaining privacy for requests that can be processed locally.

Model updating and continuous improvement present operational challenges for on-device deployment. Unlike cloud-based models that can be updated instantly for all users, on-device models require distributing updates through application releases or separate model downloads, introducing delays between improvement availability and user benefit [72]. Frequent updates consume user bandwidth and storage, potentially triggering negative reactions if not managed carefully. Federated learning approaches enable collaborative improvement of on-device models by aggregating locally computed gradients without centralizing raw data, allowing models to learn from collective user experience while preserving privacy. However, federated learning introduces technical complexity around gradient compression, secure aggregation, handling device heterogeneity, and ensuring model convergence despite non-IID data distributions and intermittent participation.

Human evaluation through user studies provides essential insights into real-world effectiveness that automated metrics cannot capture. Laboratory studies with controlled tasks assess specific capabilities such as query understanding accuracy, response relevance, and interaction efficiency under standardized conditions [73]. Field deployments with actual users over extended periods reveal usage patterns, identify edge cases, and measure long-term satisfaction and engagement. Qualitative feedback through interviews and surveys elucidates user perceptions, pain points, and feature requests that inform iterative refinement. A/B testing compares alternative model configurations or agent behaviors by randomly assigning users to different variants and measuring differences in engagement, task completion rates, and satisfaction scores. Privacy considerations require careful study design that collects only necessary data with informed consent and implements appropriate anonymization and aggregation before analysis.

6. Challenges and Future Directions

Despite significant progress in enabling on-device LLMs and AI agents for mobile UX optimization, numerous challenges remain that limit current capabilities and present opportunities for future research and development. Addressing these challenges requires advances spanning multiple disciplines including machine learning, computer systems, human-computer interaction, and privacy engineering. The following discussion identifies key limitations of existing approaches and outlines promising directions for overcoming them.

The accuracy gap between on-device models and their cloud-based counterparts remains substantial for many tasks, particularly those requiring extensive world knowledge or complex reasoning [74]. Compression techniques inevitably sacrifice some model capacity, and even carefully optimized small models cannot match the capabilities of orders-of-magnitude larger cloud models on challenging problems. Narrowing this gap requires innovations in model architectures that achieve greater parameter efficiency, training methodologies that produce more compressible representations without sacrificing capability, and hybrid approaches that intelligently partition computation between device and cloud based on task requirements. Mixture-of-experts architectures offer promising directions by activating only task-relevant model components, effectively providing larger capacity without proportional computational costs.

Long-context understanding remains challenging for on-device LLMs due to the quadratic scaling of standard attention mechanisms with sequence length. Many mobile UX scenarios require processing extended contexts such as long documents, conversation histories spanning multiple sessions, or comprehensive user behavior logs [75]. Efficient attention mechanisms improve scalability but often sacrifice modeling quality compared to full attention. Future research might explore hierarchical processing that builds compressed representations of long contexts through multiple stages, memory-augmented architectures that selectively retrieve relevant information from external storage, and continual learning approaches that accumulate knowledge over time rather than reprocessing entire contexts for each query.

Multimodal understanding that integrates language with vision, audio, and sensor data would substantially enhance the capabilities of mobile AI agents but poses significant computational challenges. Processing images and video requires orders of magnitude more computation than text, making real-time multimodal inference difficult on mobile devices [76]. Efficient multimodal fusion architectures that share representations across modalities, selective processing that applies expensive visual analysis only when necessary, and specialized hardware accelerators designed for multimodal workloads represent important research directions. Applications including visual question answering, scene understanding for augmented reality, and multimodal dialogue systems would benefit greatly from advances in efficient multimodal processing.

Personalization quality depends on accumulating sufficient data about individual users to learn accurate models of preferences and behavior patterns. Cold start problems arise when new users install applications with no prior history, and models must provide reasonable experiences before sufficient personalization data accumulates [77]. Transfer learning approaches that leverage population-level patterns while adapting to individuals, meta-learning techniques that learn how to rapidly personalize from limited data, and hybrid strategies that combine rule-based defaults with learned personalization offer potential solutions. Privacy-preserving personalization must carefully balance the benefits of learning from user data against the risks of storing and processing sensitive information locally.

Explainability and transparency become increasingly important as AI agents make autonomous decisions that affect user experiences. Users need to understand why agents take specific actions, what data informs decisions, and how to correct erroneous assumptions [78]. On-device LLMs can generate natural language explanations of agent behavior, but producing accurate and comprehensible explanations without excessive computational overhead requires specialized techniques. Future work might explore explanation generation optimized for mobile deployment, visualization approaches that communicate agent reasoning through interfaces, and interaction designs that enable users to inspect and control agent decision-making processes effectively.

Adversarial robustness and security considerations arise as on-device LLMs process user inputs that may include malicious content designed to exploit model vulnerabilities or extract sensitive information. Prompt injection attacks attempt to override model instructions through carefully crafted inputs, while model inversion attacks try to recover training data from model behavior [79]. Defensive techniques including input validation, output filtering, and adversarial training can improve robustness but may reduce model capabilities or increase computational costs. Balancing security against functionality and efficiency presents ongoing challenges requiring continued research into secure on-device AI systems.

Standardization and interoperability across platforms and devices would benefit developers and users by enabling consistent experiences and reducing fragmentation. Currently, different mobile operating systems, hardware accelerators, and deployment frameworks require separate implementations and optimization efforts [80]. Industry-wide standards for model formats, runtime APIs, and performance characterization would facilitate broader adoption and accelerate innovation. Collaborative efforts among hardware manufacturers, operating system vendors, and application developers could establish common interfaces and best practices that benefit the entire ecosystem.

7. Conclusion

The deployment of LLMs and AI agents directly on mobile devices represents a transformative development in mobile computing, enabling sophisticated natural language understanding and intelligent user experience optimization while preserving privacy and ensuring real-time responsiveness. This review has examined the current state of on-device AI for mobile UX, synthesizing recent advances in model compression, efficient architectures, optimization techniques, and application strategies that make real-time language processing feasible within the strict constraints of mobile hardware. The convergence of quantization methods, knowledge distillation, pruning strategies, and efficient attention mechanisms has enabled models with hundreds of millions to billions of parameters to execute on smartphones and tablets with acceptable latency and energy consumption.

AI agents powered by on-device LLMs transform mobile interaction paradigms from reactive command execution to proactive assistance that anticipates user needs, adapts interfaces dynamically, and automates routine tasks. These agents leverage context awareness, personalization, and natural language understanding to deliver experiences tailored to individual users and situational factors. Applications spanning predictive text input, intelligent content recommendation, adaptive interfaces, notification management, and conversational interaction demonstrate the practical benefits of on-device AI across diverse usage scenarios. The privacy advantages of local processing address growing user concerns about data collection and surveillance, while offline functionality expands the utility of AI-powered features to environments with limited connectivity.

Despite substantial progress, significant challenges remain in narrowing the capability gap between on-device and cloud models, extending context understanding to longer sequences, integrating multimodal inputs efficiently, ensuring robust personalization with limited data, and maintaining security against adversarial attacks. Future research directions include hybrid architectures that intelligently partition computation, specialized hardware accelerators optimized for transformer models, advanced compression techniques that preserve more capability with fewer parameters, and novel training paradigms that produce inherently efficient models. The continued evolution of mobile processors, memory technologies, and software frameworks will expand the feasible scope of on-device AI, enabling increasingly sophisticated applications.

The successful deployment of on-device LLMs requires holistic optimization that considers the entire system stack from model architecture through software implementation to hardware capabilities. Collaboration among researchers, developers, and hardware manufacturers will accelerate progress toward more capable, efficient, and user-friendly mobile AI systems. As these technologies mature, they promise to fundamentally reshape how users interact with mobile devices, moving toward more natural, anticipatory, and personalized computing experiences that respect user privacy while delivering the intelligence and convenience traditionally associated with cloud-based services. The foundation established by current research and development efforts positions the field for continued innovation that will define the next generation of mobile user experiences.

References

[1] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are

few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

[2] Weng, O. (2021). Neural network quantization for efficient inference: A survey. arXiv preprint arXiv:2112.06126.

[3] Xu, Y., Song, C., Sun, Y., & Yu, S. (2024, January). Advances and Challenges in Large Model Compression: A Survey. In Proceedings of the 2024 Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence (pp. 421-426).

[4] Zhang, S., Qiu, L., & Zhang, H. (2025). Edge cloud synergy models for ultra-low latency data processing in smart city iot networks. International Journal of Science, 12(10).

[5] Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., ... & Horvitz, E. (2019, May). Guidelines for human-AI interaction. In Proceedings of the 2019 chi conference on human factors in computing systems (pp. 1-13).

[6] Chen, S., Li, Q., Zhou, M., & Abusorrah, A. (2021). Recent advances in collaborative scheduling of computing tasks in an edge computing paradigm. Sensors, 21(3), 779.

[7] Chang, C. L., Li, S. C., & Huang, J. W. (2021, December). Ensemble2N et: Learning from Ensemble Teacher Networks via Knowledge Transfer. In 2021 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1-6). IEEE.

[8] Cao, Q., Irimiea, A. E., Abdelfattah, M., Balasubramanian, A., & Lane, N. D. (2021, June). Are mobile DNN accelerators accelerating DNNs?. In Proceedings of the 5th International Workshop on Embedded and Mobile Deep Learning (pp. 7-12).

[9] Lin, H., & Liu, W. (2025). Symmetry-Aware Causal-Inference-Driven Web Performance Modeling: A Structure-Aware Framework for Predictive Analysis and Actionable Optimization. Symmetry, 17(12), 2058.

[10] Wang, X., Han, Y., Leung, V. C., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. IEEE communications surveys & tutorials, 22(2), 869-904.

[11] Chen, Y., Zheng, B., Zhang, Z., Wang, Q., Shen, C., & Zhang, Q. (2020). Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions. ACM Computing Surveys (CSUR), 53(4), 1-37.

[12] Laskaridis, S., Venieris, S. I., Kim, H., & Lane, N. D. (2020, November). HAPI: Hardware-aware progressive inference. In Proceedings of the 39th International Conference on Computer-Aided Design (pp. 1-9).

[13] Lawrence, T., & Zhang, L. (2019). IoTNet: An efficient and accurate convolutional neural network for IoT devices. Sensors, 19(24), 5541.

[14] Mehta, R. (2019). Sparse transfer learning via winning lottery tickets. arXiv preprint arXiv:1905.07785.

[15] Kim, S., Park, G., & Yi, Y. (2021). Performance evaluation of INT8 quantized inference on mobile GPUs. IEEE Access, 9, 164245-164255.

[16] Younus, U., & Kamikubo, R. (2024). Privacy by Design: Bringing User Awareness to Privacy Risks in Internet of Things. arXiv preprint arXiv:2410.12336.

[17] Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., & Shlens, J. (2019). Stand-alone self-attention in vision models. Advances in neural information processing systems, 32.

[18] Sarkar, S., Babar, M. F., Hassan, M. M., Hasan, M., & Karmaker Santu, S. K. (2024, May). Processing Natural Language on Embedded Devices: How Well Do Transformer Models Perform?. In Proceedings of the 15th ACM/SPEC International Conference on Performance Engineering (pp. 211-222).

[19] Yang, S., Ding, G., Chen, Z., & Yang, J. (2025). GART: Graph Neural Network-based Adaptive and Robust Task Scheduler for Heterogeneous Distributed Computing. IEEE Access.

[20] Nagel, M., Fournarakis, M., Amjad, R. A., Bondarenko, Y., Van Baalen, M., & Blankevoort, T. (2021). A white paper on neural network quantization. arXiv preprint arXiv:2106.08295.

[21] Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., & Modha, D. S. (2019). Learned step size quantization. arXiv preprint arXiv:1902.08153.

[22] Joshi, C. K., Liu, F., Xun, X., Lin, J., & Foo, C. S. (2022). On representation knowledge distillation for graph neural networks. IEEE transactions on neural networks and learning systems, 35(4), 4656-4667.

[23] Karavangeli, E. A., Pantazi, D. A., & Iliakis, M. (2023, August). Distilgreek-bert: A distilled version of the greek-bert model.

[24] Hind, M., Wei, D., Campbell, M., Codella, N. C., Dhurandhar, A., Mojsilović, A., ... & Varshney, K. R. (2019, January). TED: Teaching AI to explain its decisions. In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society (pp. 123-129).

[25] Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., ... & Keutzer, K. (2019). Fbnet: Hardware-aware efficient convnet

design via differentiable neural architecture search. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10734-10742).

[26] Katharopoulos, A., Vyas, A., Pappas, N., & Fleuret, F. (2020, November). Transformers are rnns: Fast autoregressive transformers with linear attention. In International conference on machine learning (pp. 5156-5165). PMLR.

[27] Yang, J., Zeng, Z., & Shen, Z. (2025). Neural-Symbolic Dual-Indexing Architectures for Scalable Retrieval-Augmented Generation. IEEE Access.

[28] Ding, X., Zhou, X., Guo, Y., Han, J., & Liu, J. (2019). Global sparse momentum sgd for pruning very deep neural networks. Advances in neural information processing systems, 32.

[29] Guo, Z., & Li, X. (2020, December). Network slimming with augmented sparse training and optimized pruning. In Proceedings of the 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence (pp. 1-5).

[30] Meng, F., Cheng, H., Li, K., Luo, H., Guo, X., Lu, G., & Sun, X. (2020). Pruning filter in filter. Advances in Neural Information Processing Systems, 33, 17629-17640.

[31] Wang, J., Zhang, B., Zhao, C., & Qu, X. (2025, June). Data-free Black-box Knowledge Amalgamation. In 2025 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[32] Zhao, F., Zhang, J., Meng, Z., & Liu, H. (2021). Densely connected pyramidal dilated convolutional network for hyperspectral image classification. Remote Sensing, 13(17), 3396.

[33] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

[34] Shi, J., Gao, P., & Smolic, A. (2023). Blind image quality assessment via transformer predicted error map and perceptual quality token. IEEE Transactions on Multimedia, 26, 4641-4651.

[35] Pandey, P., Gundi, N. D., Chakraborty, K., & Roy, S. (2021, December). Uptpu: Improving energy efficiency of a tensor processing unit through underutilization based power-gating. In 2021 58th ACM/IEEE Design Automation Conference (DAC) (pp. 325-330). IEEE.

[36] Nicolicioiu, A., Duta, I., & Leordeanu, M. (2019). Recurrent space-time graph neural networks. Advances in neural information processing systems, 32.

[37] Liu, H., & Brailsford, T. (2023, September). Reproducing "Show, Attend and Tell: Neural Image Caption Generation

with Visual Attention". In Journal of Physics: Conference Series (Vol. 2589, No. 1, p. 012012). IOP Publishing.

[38] Sun, Y., Ochiai, H., & Esaki, H. (2021). Decentralized deep learning for multi-access edge computing: A survey on communication efficiency and trustworthiness. IEEE Transactions on Artificial Intelligence, 3(6), 963-972.

[39] Kilpala, M., Kärkkäinen, T., & Hämäläinen, T. (2022). Differential privacy: an umbrella review. Artificial Intelligence and Cybersecurity: Theory and Applications, 167-183.

[40] Wang, Y., Ding, G., Zeng, Z., & Yang, S. (2025). Causal-Aware Multimodal Transformer for Supply Chain Demand Forecasting: Integrating Text, Time Series, and Satellite Imagery. IEEE Access.

[41] Wang, W., Siau, K. Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: a review and research agenda. Journal of Database Management. 2019;30(1):61-79.

[42] Ilhan, F., Liu, L., Chow, K. H., Wei, W., Wu, Y., Lee, M., ... & Liu, G. (2023). EENet: Learning to early exit for adaptive inference. arXiv preprint arXiv:2301.07099.

[43] Zhang, L., Chen, L., & Xu, J. (2021, April). Autodidactic neurosurgeon: Collaborative deep inference for mobile edge intelligence via online learning. In Proceedings of the Web Conference 2021 (pp. 3111-3123).

[44] Ren, P., Xiao, Y., Chang, X., Huang, P. Y., Li, Z., Chen, X., & Wang, X. (2021). A comprehensive survey of neural architecture search: Challenges and solutions. ACM Computing Surveys (CSUR), 54(4), 1-34.

[45] Chen, Y. H., Yang, T. J., Emer, J., & Sze, V. (2019). Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 9(2), 292-308.

[46] Hadidi, R., Cao, J., Xie, Y., et al. Characterizing the deployment of deep neural networks on commercial edge devices. IEEE International Symposium on Workload Characterization. 2019;1-12.

[47] Leroux, S., Cornelissen, C., Sharma, V., & Simoens, P. (2025). Computational fairness in adaptive neural networks. Neural Computing and Applications, 1-16.

[48] Bonawitz, K., Salehi, F., Konečný, J., McMahan, B., & Gruteser, M. (2019, November). Federated learning with autotuned communication-efficient secure aggregation. In 2019 53rd Asilomar Conference on Signals, Systems, and Computers (pp. 1222-1226). IEEE.

[49] Kairouz, P., McMahan, H.B., Avent, B., et al. Advances and open problems in federated learning. Foundations and Trends in Machine Learning. 2021;14(1-2):1-210.

[50] Bengio, Y., Lecun, Y., & Hinton, G. (2021). Deep learning for AI. Communications of the ACM, 64(7), 58-65.

[51] Liu, J., Wang, J., & Lin, H. (2025). Coordinated Physics-Informed Multi-Agent Reinforcement Learning for Risk-Aware Supply Chain Optimization. IEEE Access, 13, 190980-190993.

[52] Bae, J. H., Yeo, D., Yim, J., Kim, N. S., Pyo, C. S., & Kim, J. (2020). Densely distilled flow-based knowledge transfer in teacher-student framework for image classification. IEEE Transactions on Image Processing, 29, 5698-5710.

[53] Zhu, X., Li, J., Liu, Y., Ma, C., & Wang, W. (2024). A survey on model compression for large language models. Transactions of the Association for Computational Linguistics, 12, 1556-1577.

[54] Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., ... & Weller, A. (2020). Rethinking attention with performers. arXiv preprint arXiv:2009.14794.

[55] Wang, S., Li, B.Z., Khabsa, M., et al. Linformer: self-attention with linear complexity. arXiv preprint arXiv:2006.04768. 2020.

[56] Yu, H., Peng, H., Huang, Y., Fu, J., Du, H., Wang, L., & Ling, H. (2022). Cyclic differentiable architecture search. IEEE transactions on pattern analysis and machine intelligence, 45(1), 211-228.

[57] Yu, J., Huang, T.S. Universally slimmable networks and improved training techniques. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019;1803-1811.

[58] Blakeney, C., Li, X., Yan, Y., & Zong, Z. (2020). Parallel blockwise knowledge distillation for deep neural network compression. IEEE Transactions on Parallel and Distributed Systems, 32(7), 1765-1776.

[59] Laird, J. E. (2019). The Soar cognitive architecture. MIT press.

[60] Meng, L., Shi, C., Hao, S., & Su, X. (2021, April). DCAN: Deep co-attention network by modeling user preference and news lifecycle for news recommendation. In International Conference on Database Systems for Advanced Applications (pp. 100-114). Cham: Springer International Publishing.

[61] Matsubara, F. (2024). BlitzMe: A social media platform combining smile recognition and human computation for positive mood enhancement. In Works-in-Progress and Demonstration Track of the 12th AAAI Conference on

Human Computation and Crowdsourcing (HCOMP). Pittsburgh, Pennsylvania, USA.

[62] Abascal, J., Arbelaitz, O., Gardeazabal, X., & Muguerza, J. (2023). 8 Personalizing the user interface for people. Personalized Human-Computer Interaction, 175.

[63] Weber, D., Voit, A., Kollotzek, G., & Henze, N. (2019, November). Annotif: A system for annotating mobile notifcations in user studies. In Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia (pp. 1-12).

[64] Hu, M., Qian, L., Chang, Z., & Zhang, Z. (2024). KDPG-Enhanced MRC Framework for Scientific Entity Recognition in Survey Papers. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 32, 2532-2543.

[65] Maadi, M., Akbarzadeh Khorshidi, H., & Aickelin, U. (2021). A review on human–AI interaction in machine learning and insights for medical applications. International journal of environmental research and public health, 18(4), 2121.

[66] Sun, Y., Wang, S., Li, Y., et al. ERNIE: enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223. 2019.

[67] Strubell, E., Ganesh, A., McCallum, A. Energy and policy considerations for deep learning in NLP. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019;3645-3650.

[68] Vu, K. P. L., & Proctor, R. W. (2024). Human Information Processing, Attention, and Memory: An Overview for HCI. Foundations and Fundamentals in Human-Computer Interaction, 87-122.

[69] Harvey, A. (2025). Power Consumption Patterns in Android Devices: A Data-Driven Approach.

[70] Yang, S., Ding, G., Chen, Z., & Yang, J. (2025). GART: Graph Neural Network-based Adaptive and Robust Task Scheduler for Heterogeneous Distributed Computing. IEEE Access.

[71] Yi, J., & Lee, Y. (2020, September). Heimdall: mobile GPU coordination platform for augmented reality applications. In Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (pp. 1-14).

[72] Zhang, J. (2019). Seesaw-net: convolution neural network with uneven group convolution. arXiv preprint arXiv:1905.03672.

[73] Ardito, C., Barbosa, S. D. J., Conte, T., Freire, A., Gasparini, I., Palanque, P., & Prates, R. Human-Computer Interaction–INTERACT 2025.

[74] Rae, J.W., Borgeaud, S., Cai, T., et al. Scaling language models: methods, analysis & insights from training Gopher. arXiv preprint arXiv:2112.11446. 2021.

[75] Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

[76] Mai, S., Zeng, Y., Xiong, A., & Hu, H. (2025). Injecting Multimodal Information into Pre-trained Language Model for Multimodal Sentiment Analysis. IEEE Transactions on Affective Computing.

[77] Che, S., Mao, M., & Liu, H. (2024). New community cold-start recommendation: A novel large language model-based method.

[78] Slack, D., Hilgard, S., Singh, S., & Lakkaraju, H. (2020). How much should i trust you? modeling uncertainty of black box explanations. arXiv preprint arXiv:2008.05030, 6-25.

[79] Alsmadi, I., Ahmad, K., Nazzal, M., Alam, F., Al-Fuqaha, A., Khreishah, A., & Algosaibi, A. (2022). Adversarial NLP for social network applications: attacks, defenses, and research directions. IEEE transactions on computational social systems, 10(6), 3089-3108.

[80] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Adam, H. (2019). Searching for mobilenetv3. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 1314-1324).