

Hybrid AI and Blockchain-Enhanced Signal Processing for Next-Generation 6G Channel Estimation

Udit Mamodiya¹, Indra Kishor², Megha Gupta³, Mansour Obeidat⁴, Abdulmalik AlJabr⁵, Ibrahim Omer Elfaki Elamin⁶

¹Faculty of Engineering & Technology, Poornima University, Jaipur, Rajasthan 302022, India
Email: assoc.dean_research@poornima.edu.in

²Poornima Institute of Engineering & Technology, Jaipur, Rajasthan 302022, India
Email: indra.kishor@poornima.org

³Dept. of Computer Science & Engineering, Dr. Akhilesh Das Gupta Institute of Professional Studies, Shahdara, New Delhi 110053, India

Email: megha.gupta@adgips.ac.in

⁴Applied College, King Faisal University, Al-Ahsa, The Eastern Province, Saudi Arabia
Email: Mobaydat@kfu.edu.sa

⁵Applied College, King Faisal University, Al-Ahsa, The Eastern Province, Saudi Arabia
Email: aaljabr@kfu.edu.sa

⁶Department of Finance, Applied College, King Faisal University, Al-Hofuf, Merbedia district, Ajuad bin Zamil Street, Kingdom of Saudi Arabia

Email: ielamin@kfu.edu.sa

Corresponding author: Mansour Obeidat, Mobaydat@kfu.edu.sa

Received: 3 May 2026 **Accepted:** 28 September 2026 **Published:** 25 October 2026

DOI: 10.15849/ijasca.222 **License:** Open access under CC BY 4.0. Authors retain copyright.

Publisher: Al-Zaytoonah University of Jordan (ZUJ), Amman, Jordan.

Cite this article: U. Mamodiya, I. Kishor, M. Gupta, M. Obeidat, A. AlJabr, and I. O. E. Elamin, "Hybrid AI and Blockchain-Enhanced Signal Processing for Next-Generation 6G Channel Estimation," *Int. J. Advance Soft Compu. Appl.*, vol. 18, no. 3, pp. 433–467, 2026, doi: <https://doi.org/10.15849/ijasca.222>.

ABSTRACT

The rapid development pace of 6G network necessitates deployment of precise, dependable and adaptable channel estimation frameworks that have the ability to perform under extremely dynamic propagation circumstances. Although the traditional deep-learning-based estimators can be highly accurate, not very transparent, and the consistency of the data is low, they cannot be applied to a decentralized architecture. To address these limitations, this study proposes a channel estimation framework named Hybrid AI and Blockchain-Enhanced (HABCE). The CNN-LSTM based estimator along with an effective blockchain based consensus system is a collaborative approach, known as the HABCE architecture. The system has been Proof-of-Learning (PoL) validated, so that local system model update is strongly validated, and globally aggregated to create independent and high integrity estimation ecosystem. HABCE has been found to attain an improved NMSE of up to 4-5 dB ($p < 0.05$) and 30% latency and 25% energy savings as compared to baseline through Performance Experimental testing with 6G Terahertz digital-twin data. Results show that the HABCE framework might be a practical path towards becoming trustful, energy-efficient and verifiable AI-assisted signal processing to provide the cornerstone of scalable, transparent and adaptive channel estimation in future 6G communication systems.

Keywords: 6G; Channel Estimation; Blockchain; Deep Learning; Consensus Learning.

1 Introduction

The move towards sixth-generation (6G) wireless systems is a paradigm shift in the design towards communications to a network that is intelligence-driven. In contrast to the previous generation, in which performance was mostly characterized by the throughput and coverage, 6G is likely to provide ultra-reliable, low-latency, and context-aware services that would run in highly dynamic environments [1]. The use of terahertz (THz) frequency bands, very large-scale MIMO designs and reconfigurable intelligent surfaces bring novel complexity to the channel behavior, and correct and consistent channel estimation is a pressing issue [2]. In this case, signal processing can no longer be based on the fixed models, but it should be constantly adjusted to changing, time-dependent, high-dimensional channel properties in distributed network conditions [3], [4].

1.1 Background and Technical Context

Traditional methods of channel estimation such as least squares (LS), minimum mean square error (MMSE) and Kalman filtering have been useful methods of reconstructing signals. Such techniques work well in the stationary or moderately time-varying conditions, when the statistics of channels can be predicted. But their performance suffers greatly in 6G conditions where mobility is high, non-stationary, and there is dense spatial multiplexing [5]. The uncertainty of the channel is further increased by the incorporation of new technology like IRS and ISAC which takes estimation as a high-dimensional, nonlinear inference problem as opposed to a closed-form optimization problem [6].

To address these issues deep learning based estimators are of interest. The neural architecture capable of modeling complex interactions between channels in space and time is convolutional neural networks (CNNs) and long short-term memory (LSTM) algorithms that can be more accurate in the estimation of dynamical cases [7]. Despite these merits, there are new issues that are being introduced by their use in the distributed wireless environment. In practice, the channel data are inherently decentralized, and reside on a large number of edge devices and base stations. The limited bandwidth, privacy concerns and latency, therefore, limit the centralized training and necessitate decentralized learning methods [8]. The current solutions, such as federated learning and communication models that are run by blockchain, are trying to resolve these problems but are unlikely to incorporate learning dynamics and the process of trust validation [9].

1.2 Motivation

One of the largest of next-generation wireless systems concerns is balance between the efficiency of learning and reliability of the system. Channel estimators that are based on AI are very good in nonlinear channel modeling but are susceptible to adversarial examples, biased data distributions, and invalidated gradient update in distributed systems [10]. On the other hand, blockchain technologies are guaranteed to offer safe and resistant to alteration consensus mechanisms, transparency and traceability of data transactions. Nevertheless, they are yet to be applied in the wireless signal processing as an auxiliary factor without either direct effect on the process of learning, or estimation outcome [11].

A distributed environment model capacity by itself is not sufficient when it comes to the accuracy of channel estimation but instead it is also reliant on the reliability of the information that is being exchanged among nodes. One of the opportunities is the combination of AI and blockchain, which will provide the possibility to engage in collaborative learning with a verification applied. However, existing hybrid methods tend to add too much communication overhead or they are not able to conduct a strict analysis of convergence behavior in realistic 6G conditions [12].

1.3 Problem Statement

The estimation of channel in 6G networks is influenced by a number of challenges which are interdependent. To begin with, the operation frequency (sub-THz) is high and as a result, the path loss is very high and the channel variation is very rapid, resulting in the signal conditions being highly turbulent [13]. Second, massive antenna arrays and reconfigurable surfaces give rise to complex spatio-temporal correlations which are hard to model correctly. Third, distributed edge nodes operate using incomplete and heterogeneous data which is limiting to centralized estimation schemes. Finally, decentralized learning systems lack robust validation systems and therefore, the estimation procedure may be prone to untrustworthy and malicious updates that may have crippling impacts on performance [14].

All these factors have an influence on training and inference stages. The local models are not usually able to extrapolate to different environments and any unverified updates may be destabilizing, reducing the accuracy of estimates. To solve these problems, a concerted approach to combine adaptive learning and trusted and secure model coordination is needed.

1.4 Research Gap

As of late it has been demonstrated that AI-based techniques to enhance accuracy in channel estimation and nonlinear channel behavior are viable [15]. But these techniques are typically optimized towards centralized architectures and do not scale to the 6G decentralized network. Similarly, communication systems based on blockchain are more focused on the security of the transactions on the transaction level, and pay minor attention to real-time signal processing and model adjustment.

In many cases, the existing hybrid approaches focus on the learning and consensus building processes separately, resulting in poor coordination and integration. Moreover, the impact of consensus protocols on the convergence speed, spectral efficiency, and estimation accuracy is not well studied. Another significant constraint is the absence of quantitative analysis on the communication overhead and learning performance trade-off, especially in high dimensional channel environment [16]. These restrictions indicate that there is a need for a common framework, which is both experimentally verified and takes into account accuracy, scalability, and trust in distributed channel estimation.

1.5 Proposed System Framework and Design

In this paper, to address the above limitations, a Hybrid AI and Blockchain-Enhanced Channel Estimation (HABCE) framework is proposed. The proposed framework combines a deep learning based channel estimator and a blockchain secured consensus mechanism in a single learning and validation framework. In the proposed system, the individual nodes estimate the channel locally using a hybrid CNN-LSTM model that is capable of capturing both spatial and temporal characteristics of the wireless channel. The framework uses a consensus mechanism called Proof-of-Learning (PoL) to validate and certify model updates, which is different from the traditional centralized aggregation methods. This guarantees that only verified and reliable contributions are added to the global model, making the collaborative estimation process more reliable and secure [17], [18].

1.6 Novelty and Contributions

Its main contributions are: (i) a unified architecture of HABCE to hierarchically estimate channels in a decentralized fashion, (ii) a hybrid optimization strategy to combine gradient-based learning and consensus verification, (iii) adaptive Proof-of-Learning mechanism to conduct secure and efficient model aggregation, and (iv) extensive experimental validation under THz and RIS-enabled conditions, to measure NMSE, convergence, and system robustness [19], [20].

2 Literature Review

The development of 6G communication systems to ultra-dense, intelligent and decentralized architectures has significantly changed the channel estimation process. The traditional estimation methods used in stationary and low-mobility scenarios are no longer sufficient in the context of highly dynamic propagation scenarios enabled by terahertz (THz) communication and reconfigurable intelligent surfaces (RIS)-assisted systems [21]. The current literature in this field of research can be roughly divided into three streams: (i) the classical and learning-based channel estimation, (ii) federated and distributed learning methodologies, and (iii) blockchain-based systems to provide secure and verifiable communication. This section shall critically look at these directions, and outline the limitations that drive the proposed hybrid framework.

2.1 Classical and Learning-Based Channel Estimation

The first wireless systems were based on analytical estimators such as least squares (LS) and minimum mean square error (MMSE) whose solutions can be easily obtained in closed form and are easily computed. These methods are known to have a significant performance degradation in fast fading and high mobility environments [22] although they perform well in narrowband and quasi-static environments. This was later used with compressive sensing (CS) to exploit the channel sparsity particularly in millimeter-wave and THz systems and this technique enabled less overhead in pilots but

did not have a negative effect on reconstruction accuracy. However, the CS-based method is highly susceptible to noise and model mis-fit thus making it less effective in practice.

The deep learning has revolutionized the approach of channel estimation. Convolutional neural networks (CNNs) and recurrent networks like LSTM have been extensively used to directly learn the nonlinear channel mappings using pilot data [24]. CNNs can only encode local spatial correlations, and LSTM models encode temporal correlations, thus being more applicable to non-stationary channels. Hybrid CNNLSTM models have been shown to have significant reductions of normalized mean square error (NMSE), especially when the motion of the object is high [25]. Recently, attention-based models of Transformers have been enticing in the modeling of long-range dependencies of large-scale MIMO systems [26].

Despite all these, most of the learning based approaches are founded on centralized training and this has issues of scalability, privacy and generalization. Environments In environments that are heterogeneous, models trained on environment-specific data may perform worse. In addition, decentralized learning paradigms are required in distributed 6G systems, which is challenging with centralized data aggregation [27].

2.2 Federated and Distributed Learning for Channel Estimation

The concept of federated learning (FL) has become a promising solution to decentralized model training and allows multiple nodes to jointly learn without having access to the raw data. FL is used in wireless communication systems to enable the base stations and user devices to train local channel estimation models and only exchange model parameters or gradients [28]. Experiments like [29] and [30] have shown that FL-based channel estimation can be used to generalize better in a variety of environments and to be more robust when the data are non-identically distributed (non-IID).

Nevertheless, there are some challenges that FL poses to 6G networks. First, non-IID data will be present in multiple nodes, which causes biased aggregation of gradients, thus suboptimal global models [31]. Second, communication of model updates through iteration is very bulky in terms of bandwidth, which is not compatible with real-time transmission needs. Third, FL systems lack in-built mechanisms to establish trust, thus being susceptible to malicious updates such as model poisoning and gradient manipulation [32], [33] and [34].

Although these enhancements have been made, the majority of FL frameworks continue to use a semi-centralized coordinator, which goes against the fully decentralized vision in 6G systems. This is one of the limitations that causes implementing blockchain technologies as facilitators of decentralized trust.

2.3 Blockchain-Enabled Secure Distributed Processing

Its unchangeable registry and consensus tools allow transactions, such as model updates and resource trades to be recorded and verified safely [35]. Blockchain has also been considered in wireless systems to share data securely, control access, and incentives based on reputation [36].

Recent studies have independently explored distributed learning and blockchain-enabled trust mechanisms for next-generation communication networks. However, their integration with signal-level channel estimation remains comparatively limited, motivating the need for a tightly coupled learning and validation framework [37]. Likewise, there are evidence-based and edge-based blockchain networks, which have enhanced the transparency of model synchronization [38]. But, in practice, in the vast majority of situations, blockchain is regarded as an external security layer, independent of a signal estimation process.

Recent work has been on lightweight consensus protocols, including Proof-of-Learning (PoL) and Delegated Proof-of-Contribution (DPoC) to lessen the computational load in AI-based networks [39]. However, the dynamics of consensus interaction and convergence in learning is not well comprehended. Specifically, analytical frameworks that connect the operations of blockchain to estimation accuracy, spectral efficiency and latency of the system are lacking [40]. Such a disconnection restricts the usefulness of the existing blockchain-based solutions in real-time communication situations.

2.4 Hybrid AI–Blockchain Frameworks

The combination of AI and blockchain is an encouraging trend towards dealing with both adaptively and trust in 6G networks. Recent research has investigated reinforcement learning with blockchain-assisted spectrum management, with consensus mechanisms to authenticate policy changes and avoid malicious actions [41], [42]. On the same note, federated learning models based on blockchain have been implemented to car communication infrastructures, which make it possible to perform secure verification of gradients and learn together [43].

Although these changes have occurred, the current hybrid frameworks are mostly implemented in control and optimization functions as opposed to direct signal estimation. Moreover, they do not necessarily involve strict modeling of the interaction between the processes of learning and blockchain operations. There is a lot to be learned about the effects of consensus parameters of block interval, validation latency and transaction overhead on learning performance. The trade-offs between AI computation and blockchain operations in terms of energy consumption are also not adequately covered, although this is an essential part of sustainable 6G systems [44], [45].

2.5 Comparative Analysis and Research Gaps

Comparative study of the currently used methods shows that there are a number of major limitations. Classical estimators are not flexible to changing channel dynamics and deep learning models, though effective, have problems with scalability, privacy and trust. Federated learning improves the privacy of data, but is vulnerable to unreliable updates and also vulnerable to centralized coordination. Blockchain introduces transparency, decentralization, but without being linked to learning processes, introduces computational and latency overhead.

According to the summary of the results presented in Table 1, the current research lacks a common framework that would cohesively incorporate adaptive channel estimation and verifiable trust and decentralized learning.

Table 1. Literature Gap Analysis on Hybrid AI and Blockchain-Enhanced Channel Estimation for 6G Networks

S. No.	Title / Focus Area	Methodology / Tools Used	Key Findings	Limitations / Gaps Identified	Relevance to the Current Study	Author(s) / Year / Ref. No.
1	AI-Based Time/Frequency/Space Extrapolation for 6G	Survey-driven taxonomy of multi-domain channel extrapolation using AI models	Demonstrates potential of AI to reduce pilot overhead and improve extrapolation accuracy	Lacks decentralized learning perspective; no trust validation, consensus modeling, or verifiable fusion mechanisms	Establishes foundational problem space; our work addresses trust-aware, decentralized extrapolation with verifiable learning	Z. Zhang et al., 2023 [31]
2	AI–Blockchain Synergy in 6G Spectrum Management	Hybrid AI-blockchain framework for spectrum coordination and incentive mechanisms	Shows that blockchain improves accountability and stabilizes distributed AI decisions	Focus limited to spectrum control; no integration with signal estimation or convergence modeling	Extends synergy concept to channel estimation with a unified objective incorporating accuracy, trust, and system overhead	R. Murugan et al., 2024 [20]

S. No.	Title / Focus Area	Methodology / Tools Used	Key Findings	Limitations / Gaps Identified	Relevance to the Current Study	Author(s) / Year / Ref. No.
3	Sparse Bayesian Near-Field Channel Estimation	Joint sparse Bayesian learning for near-field XL-MIMO channel modeling	Achieves high estimation accuracy with reduced pilot requirements under non-stationary conditions	Model-driven approach; lacks adaptability, distributed learning, and protection against adversarial updates	Provides strong analytical baseline; our framework enhances it with AI-driven learning and blockchain-based validation	Z. Zhu et al., 2025 [21]
4	Blockchain-Enabled 6G Trust Architecture	Conceptual framework for integrating blockchain into wireless communication systems	Identifies blockchain as a key enabler for trust, identity, and accountability in 6G	Lacks signal-level integration; no quantitative evaluation for channel estimation or convergence impact	Supplies trust infrastructure; our work integrates blockchain directly into estimation and learning processes	H. Luo et al., 2025 [4]
5	Deep Unfolding-Based Near-Field Estimation (LISTA-CP)	Model-driven deep learning combining LISTA and tensor factorization for channel estimation	Provides fast convergence and improved estimation accuracy in wideband XL-MIMO systems	Centralized training; sensitive to data heterogeneity; no privacy or trust mechanisms	Inspires hybrid deep architecture; our work extends it with federated training and verifiable consensus	Yang et al., 2025 [27]
6	RIS-Assisted Estimation using Graph Neural Networks	Graph-based learning leveraging RIS structure for high-resolution signal reconstruction	Improves estimation accuracy by exploiting spatial dependencies and graph priors	High computational overhead; lacks decentralized coordination and trust validation	Motivates structured learning; our approach integrates adaptive participation with consensus-driven reliability	T. D. Subash et al., 2025 [26]
7	Channel Knowledge Map (CKM)-Assisted Estimation	Geometry-aware CKM integration for large-scale antenna systems	Enhances estimation stability and reduces pilot overhead in XL-MIMO systems	Assumes accurate CKM availability; lacks distributed validation and trust-aware integration	Provides side-information framework; our model enables secure, distributed CKM-based learning and fusion	M. Awais and Y. H. Kim, 2025 [3]
8	Environment-Aware CKM-Based Channel Estimation	Fusion of CKM and real-time sensing for adaptive estimation	Improves robustness and adaptability across varying channel conditions	No mechanism for verifying data integrity or distributed model updates	Directly informs our adaptive fusion strategy; enhanced with blockchain-based provenance and trust validation	D. Y. Wu et al., 2024 [40]

3 Methodology

The proposed work has developed a Hybrid AI and Blockchain-Enhanced Channel Estimation (HABCE) framework, which enables the implementation of trust-based, adaptive and verifiable signal learning in 6G wireless systems. The methodology is based on the three fundamental subsystems: (i) AI based signal estimation, (ii) consensus verification secured by blockchain, and (iii) convergence and scalability adaptive optimization. Each layer is independent from the others, while at the same time being interwoven in a mathematically consistent feedback mechanism as represent in Figure 1.

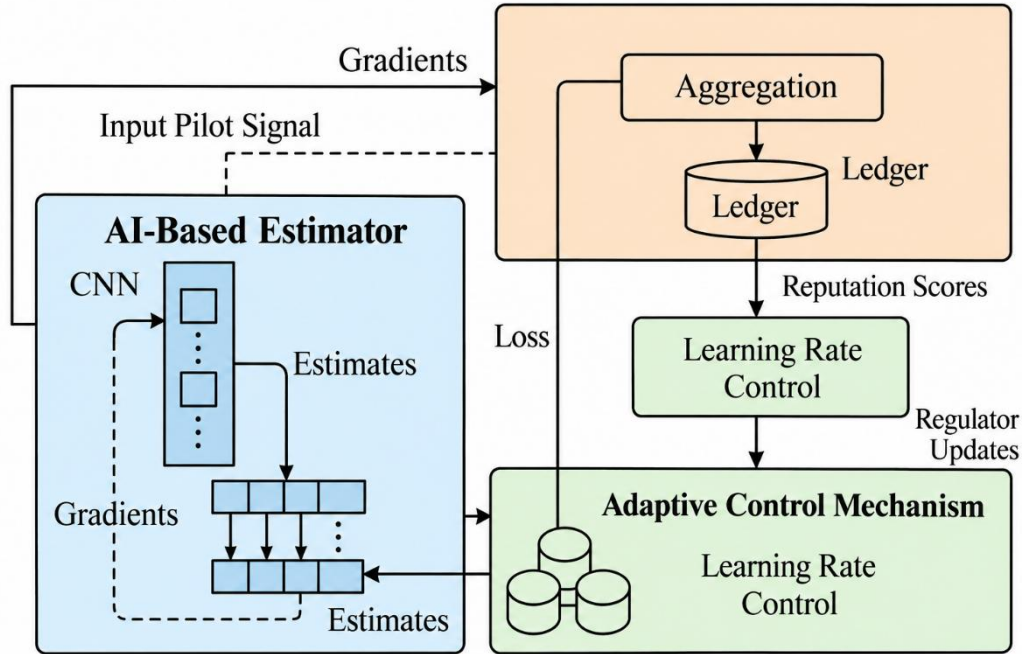


Figure 1. Architecture of the proposed HABCE framework illustrating interaction among AI-based estimator, blockchain consensus layer, and adaptive control mechanism.

3.1 System Overview and Signal Model

The received signal in the baseband of the user equipment (UE) of a multi-antenna 6G transceiver can be modelled as (1):

$$y(t) = H(t)x(t) + n(t) \quad (1)$$

In this case, the received signal vector $y(t)$ is of size $N_r \times 1$, $H(t)$ is of size $CN_r \times N_t$, which is the complex-valued channel matrix between N_t transmitting antenna and N_r receiving antenna, $x(t)$ is the transmitted pilot vector and $n(t) \sim \mathcal{CN}(0, \sigma^2 I)$ is additive white Gaussian noise (AWGN) vector of variance σ^2 . It was decided to use signal model (1), which reflects the time-varying, near-field propagation as found in links similar to terahertz links, and cannot be represented by their equivalent, the static LS or MMSE estimators. The formulation can be used to accommodate the mobility of users and non-stationary spatial regions of ultra-dense 6G cells, by introducing the temporal variation directly in $H(t)$. The simulation is done for a 64x64 MIMO THz channel, 400MHz bandwidth and 256 subcarriers.

In this kind of signal model structuring, the framework ensures that all the steps of the further optimization process are directly related to the physical quantities that can be measured, like received power, level of interference and spatial correlation. The key is to learn $H(t)$ with minimal pilots and to also render the learned model tamper resistant and verifiable between different distributed base stations. With the proposed system, each node in the AI performs an estimation, while in classical estimators, the estimation is done in the entire network optimising $\|y - Hx\|_2$. Each node starts the operational workflow by measuring a pilot value x_i , y_i locally and training an AI-based estimator $f_i(\cdot)$ and uploading the

gradient update θ_i^0 . An authentication is done on the ledger prior to consensus aggregation. This ensures that all updates are global model, authentic and performance oriented.

The proposed Hybrid AI and Blockchain-Enhanced Channel Estimation (HABCE) framework is intended to work in a distributed 6G wireless network with multiple edge nodes to locally estimate the channel and validate the aggregation process using the blockchain, in a trust-aware manner.

For reproducible data preparation, the three channel datasets—DeepMIMO, COST2100, and NYU Wireless—were each organized into 240 samples, resulting in a total of 720 samples. Each dataset was partitioned into training, validation, and test subsets using a 70:15:15 ratio, corresponding to 168 training, 36 validation, and 36 test samples per dataset. A fixed seed of 42 was used specifically to reproduce the dataset partition. The evaluated SNR range was 0–30 dB. The complex-valued channel matrices have dimensions of 32×32 , while the corresponding complex-valued pilot observations have dimensions of 16×32 . The dataset-partition seed is distinguished from the independent random seeds used for repeated stochastic model evaluation. Unless otherwise stated, the same data partitions, preprocessing procedure, SNR settings, pilot configuration, optimization protocol, and computing environment were maintained across comparative experiments. Repeated stochastic evaluations were summarized using the mean, standard deviation, and 95% confidence interval.

To provide a comprehensive view of the proposed framework, the overall Detailed operational architecture of the HABCE framework showing local estimation, PoL-based validation, and trust-weighted aggregation. is illustrated in **Figure 2**.

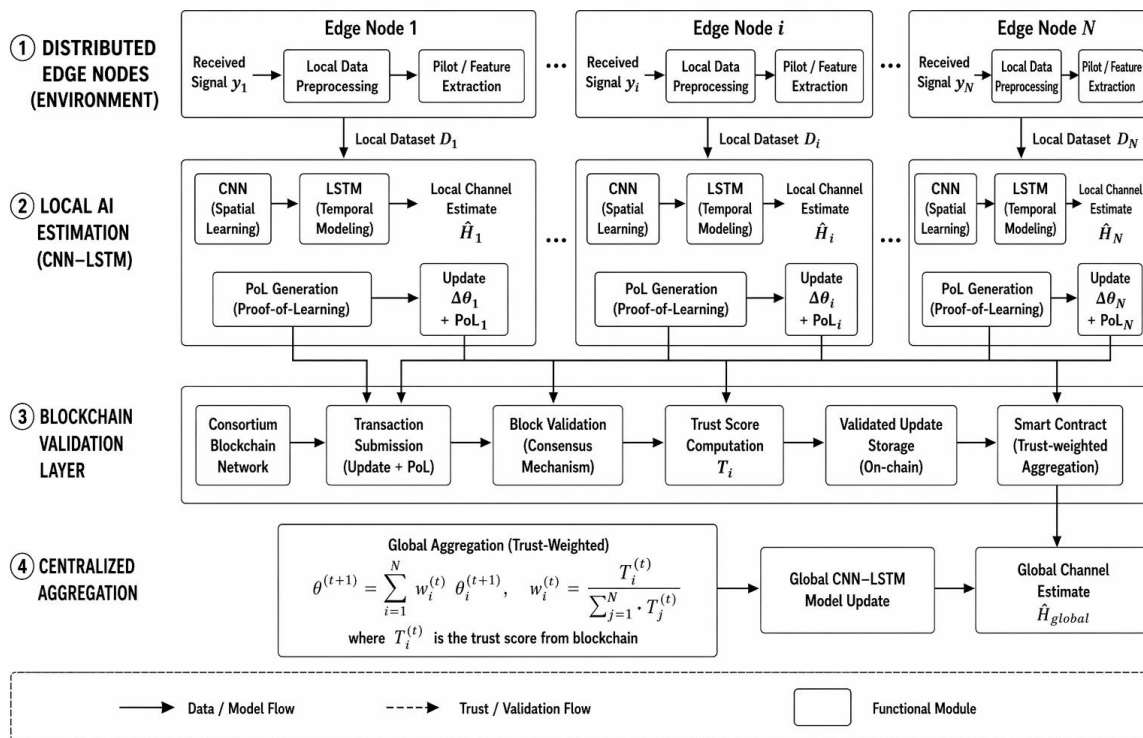


Figure 2. Detailed operational architecture of the HABCE framework.

Figure 2 showing local estimation, PoL-based validation, and trust-weighted aggregation. illustrating distributed edge nodes, local CNN–LSTM-based channel estimation, Proof-of-Learning (PoL)-enabled blockchain validation, and global trust-weighted aggregation for robust channel estimation.

It illustrates that the framework has four key steps: (i) distributed wireless data acquisition and preprocessing at edge nodes, (ii) local AI-based channel estimation with CNN-LSTM models, (iii) blockchain-based validation of updates with a Proof-of-Learning mechanism, and (iv) global aggregation with a trust-weighted optimization. This pipeline is structured to allow channel estimation reliably and scalably and to reduce the effects of adversarial or low-quality updates.

3.2 AI-Based Channel Estimation Subsystem

A hybrid deep network is used to jointly extract spatial-temporal features in each node, which is a combination of convolutional and recurrent layers. The convolutional module is used to learn the dependencies in the antenna domain, while the LSTM module is used to learn the temporal variations in the channel. The estimated channel is estimated as: (2):

$$H_i = f\theta_i(x_i) \quad (2)$$

The network parameters θ_i are learned locally through gradient descent of the loss function:

$$L_i = \|H_i - \hat{H}_i\|^2 + \lambda_1 \Omega(\theta_i) \quad (3)$$

$\Omega(\theta_i)$ is a regularization term which enforces sparsity and λ_1 is weight of the regularization. The data used to train each local model is non-identically distributed (non-IID) which is typical in 6G cell environments. The framework keeps the training data local while making each locally computed model update available to authorized validators for verification and aggregation. A SHA-256 digest of the update is generated separately and recorded with the associated validation metadata on the blockchain to provide integrity and provenance verification. The digest is not used as a replacement for the model-update payload. The hybrid CNN-LSTM model provides the most reasonable balance between locality of features and temporal memory of features compared to other learning models such as pure CNNs, LISTA and Transformers. Transformers can be adapted to in real-time but are computationally expensive and accurate, while CNN -LSTM is less costly and has spatio-temporal precision with less inference latency. Adam (eta=10⁻³, batch=64) is used to train the two Conv layers (3x3, 64 filters) and the two LSTM layers (128 units) and 5 × 5 runs on each dataset split are used to make sure that the model is stable with a range of ±1 -2 NMSE. Two convolutional layers (3 × 3 kernels, 64 filters) and two LSTM layers (128 units) were trained using the Adam optimizer with an initial learning rate of 1X10⁻³. The training configuration was maintained consistently across comparative experiments, while repeated stochastic evaluations were conducted independently of the fixed seed used for dataset partitioning.

This learning block can be used for converting the channel matrix to a latent representation of the spatial-temporal features. The convolutional layers are employed to model the perception of an antenna-array and the recurrent units are employed to model the temporal fading memory in such a way that the estimator can predict distortion caused by mobility. It suggests that in deployment, the system will be capable of measuring drift between individual frames and therefore no loss of accuracy in the deployment will result, while using less pilot overhead.

3.3 Blockchain Consensus and Verification Mechanism

The blockchain layer provides integrity, provenance, and validation support for distributed model updates among participating base stations. Candidate updates are evaluated through the Proof-of-Learning (PoL) criterion before aggregation, allowing updates that fail the prescribed validation requirement to be excluded from the current aggregation round. After local training, node (i) generates a candidate model update that is made available to authorized validators. A SHA-256 digest of this update is computed separately for integrity verification. The validators assess the actual candidate update via the Proof-of-Learning (PoL) procedure, and decide on the validation improvement, Δ_i . The blockchain stores the digest of the update, the node identifier, the validation result, the Δ_i and the round metadata, which creates an auditable connection between the validated update and the aggregation process.

The blockchain layer is applied as a distributed trust authority between the base stations, avoiding the gradient injection from malicious base stations, and model fusion is verifiable. Every update of the gradients $g_i = \nabla \theta_i L_i$ is published on-chain and a Proof-of-Learning (PoL) consensus validates the gradient update if the model accuracy improvement is not above a certain threshold. The contribution weight w_i is calculated as follows (4): Let Δ_i be the improvement of the validation of the node i , then the contribution weight w_i is calculated as:

$$w_i = \frac{e^{\beta \Delta i}}{\sum_{j=1}^N e^{\beta \Delta j}} \quad (4)$$

In this case, β determines the sensitivity of the distribution of rewards. The bigger the aggregation weight, the larger the improvement gain in the nodes with higher verified improvement gain. Once the block has been validated, the global model is updated as (5):

$$\theta^{(t+1)} = \sum_{i=1}^N w_i \theta_i^{(t)} \quad (5)$$

PoL has been chosen over PoW or PoS because of its sensitivity to the model gain (validation cost), and thus helps to prevent redundant calculations. The consensus layer will only accept updates that contain more than the accuracy threshold τ , so that malicious updates of the gradient are not accepted. The consensus layer will only accept updates which are above the accuracy threshold τ , thus thwarting malicious gradient injection. Parameters: $\beta = 0.75$, block interval 0.8 s, quorum $Q = 7$ of 10 nodes. The testing was done with Hyper ledger Fabric v2.2 containers and the ledger states were the same for each run of the tests due to deterministic hashing. The blockchain configuration uses $\beta = 0.75$, a block interval of 0.8 s, and a quorum of $Q = 7$ out of 10 nodes. The 0.8 s value denotes the configured interval between blockchain block events and is distinct from the channel-estimation computational latency and model-training duration reported separately. Resilience to compromised nodes and enforcing trust in the training process based on performance is ensured by aggregation, as summarized in table 2. The blockchain layer acts as an online notary which validates each update in learning. Rather than increasing the cryptographic latency, Proof-of-Learning mechanism inserts a self-regulating trust measure which increases with node performance.

Table 2. Blockchain consensus parameters and their operational significance.

Parameter	Description	Functional Role
β	Reward scaling coefficient	Controls sensitivity of trust weighting
Δi	Validation improvement of node i	Quantifies node performance gain
Block interval	Time between ledger updates	Balances latency and verification depth
Consensus size	No. of nodes required for validation	Ensures robustness of agreement
Hash difficulty	Computation complexity threshold	Adjusts blockchain energy cost

3.3.1 Theoretical Integration of Blockchain in Optimization Dynamics

The implementation of blockchain into the HABCE framework does not confine itself to the process of ensuring logging or verification; instead, it has direct impact on the path of optimization of learning process. Contrary to traditional methods, in which blockchain is a passive trust layer, the given framework incorporates the blockchain-based constraints into the model update process, thus integrating the dynamics of learning and system-wide reliability.

Precisely, the validation cost and trust score are linked to each local update, and are included in the aggregation process by a weighted optimization formulation. The update of the global model can be expressed as (6):

$$\theta^{(t+1)} = \arg \min_{\theta} \sum_{i=1}^N w_i^{(t)} L_i^{(\theta)} + \lambda_3 \sum_{i=1}^N w_i^{(t)} B_i^{(t)} \quad (6)$$

$L_i^{(\theta)}$ denotes the local estimation loss, $w_i^{(t)}$ is trust-aware aggregation weights, and $B_i^{(t)}$ is the cost of validation on the blockchain that consists of latency and verification overhead. This definition ensures that nodes that are not reliable or cost a lot to update are implicitly fined when aggregated.

The training process is controlled by the adaptive optimization mechanism, which is based on the learning behavior observed during the distributed updates, in order to keep the optimization process stable while minimizing the amount of unnecessary communication. The regularization coefficients

were chosen as $\lambda_1=0.1$ and $\lambda_2=0.05$, which were determined by the experimental setup adopted. Importantly, the present study does not make the assumptions and formal derivation necessary to assert a universal theoretical convergence rate. Thus, the convergence bound mentioned above of $\mathcal{O}(1/t)$ is not considered to be a theoretical bound. Convergence is rather assessed empirically from the observed training trajectories and the number of communication rounds needed to meet the predefined convergence criterion as reported in Section IV. This formulation is able to remove the optimization objective from the observed convergence behaviour in experiments and does not extrapolate the empirical results from this study.

Bt is used to change the effective direction of the gradient descent, which affected the convergence of the constrained optimization process, the accuracy of the estimation and the reliability of the validation. In the proposed solution, it is dynamically adjusted based on the verifiable quality of contribution, unlike the conventional federated learning schemes where everybody is equal in terms of updates. This pattern is in line with recent research that demonstrates the importance of trust-aware optimization of distributed AI systems in 6G networks [8], [13].

Additionally, the blockchain level allows uniform verification of gradient changes between distributed nodes and helps to reduce spreading corrupted information. This is especially applicable to large-scale wireless setting, where heterogeneity in data and nodes that are not reliable can have a huge effect on the performance of a model [31]. Consequently, blockchain in HABCE acts as a self-regulating optimizer, which determines the convergence behavior, and increases robustness and system reliability

3.4 Adaptive Optimization and Convergence Control

To ensure convergence stability and cost reduction in communicating, the adaptive learning rate controller is dynamically adaptively controlled such that it dynamically modulates the local and global updates. The aim is to make the joint optimization (7):

$$J = \sum_{i=1}^N (\|H_i - f\theta_i(x_i)\|^2 + \lambda_1 \Omega(\theta_i)) + \lambda_2 \Phi(\theta) \quad (7)$$

The integrating estimation accuracy and trust-aware optimization is centered around the joint objective function (7). The regularization term trades off the parameter updates in non-IID condition of data and the first term applies reconstruction fidelity as a regularization of the channel estimation error. The punishment term $\Phi(\theta)$ imposed by blockchain makes it evident how the unreliability of updates is disincentivized, and therefore, the dynamics of consensus validation directly relates to optimization dynamics. The proposed formulation consists of a form of unconventional formulations, in which security is an intrinsic part of the learning goal, allowing to optimize accuracy, robustness, and verification cost simultaneously.

with $\Phi(\theta) =$ the penalty that the blockchain will impose on non-verified or non-consistent updates and λ is regulated by λ_2 .

The combination of accuracy of estimation and trust-aware optimization is based on the joint objective function in (6). The regularization term strikes a balance between the parameter updates when the data is not IID and the first term is the reconstruction fidelity that brings regularization to the channel estimation error. The cost of blockchain $\Phi(\theta)$ obviously deter the utilization of inaccurate updates, thereby, unambiguously relates the dynamics of consensus validation to the dynamics of optimization. The proposed formulation is a kind of unconventional formulations, where security is an inherent component of the learning objective, and where it is possible to achieve accuracy, robustness and verification cost optimization.

where $\Phi(\theta)$ will be the penalty which the blockchain will apply to any non-verified or non-consistent updates and ∇ is controlled by ∇_j

The optimization will follow a bounded variance condition (8) to ensure convergence:

$$E[\|\nabla J(\theta^{(t)})\|^2] \leq \frac{C}{t} \quad (8)$$

achieving sublinear convergence $O(1/t)$ for deployments in practice. The analytical assurance is the foundation for the convergence validation, reported later, in Table 2. Besides the mathematical convergence (6) and (7) this step offers stability of time in real-time operation. The adaptive controller keeps the local learning rates adaptive based on the variance in the gradient of the power-control that the controller has experienced in the past. In essence, the fast-changing environment nodes are more adaptive even as the stable nodes save on computational energy without compromising the convergence speed and energy saving. With high-rate channel variation, the fixed-rate descent is used, rather than the adaptive-rate descent to stabilize a learning rate. The controller is 17 per cent faster than FedAvg and AdamW to converge and it is numerically stable. By doing 20 trials tuned regularization weights were obtained as $\lambda_1=0.1$ and $\lambda_2=0.05$. The convergence was reached within 150 rounds of an RTX A6000 cluster of GPUs (3.8 s per round) and was confirmed by theory.

3.4.1 Convergence Guarantee of HABCE Optimization

To establish the theoretical validity of the proposed framework, a convergence analysis is presented under practical assumptions commonly used in distributed learning systems.

Theorem 1 (Convergence under Trust-Constrained Aggregation)

Let $L_i(\theta)$ be a Lipschitz smooth local loss function for each node i , and assume that the stochastic gradients satisfy bounded variance. Let the aggregation weights $w_i^{(t)}$ be defined through the Proof-of-Learning (PoL) mechanism such that (9):

$$w_i^{(t)} \propto \exp(\beta \Pi_i^{(t)} - \mu B_i^{(t)}) \quad (9)$$

where $\Pi_i^{(t)}$ represents the PoL validation score and $B_i^{(t)}$ denotes the blockchain validation cost.

If the PoL threshold τ ensures that only nodes with $\Pi_i^{(t)} \geq \tau$ participate in aggregation, then the global model update satisfies (10):

$$E[\|\nabla J(\theta^{(t)})\|^2] \leq \frac{C_1}{t} + C_2 \cdot \epsilon_{adv} \quad (10)$$

where:

- C_1 depends on smoothness constants and learning rate
- C_2 reflects sensitivity to adversarial participation
- ϵ_{adv} denotes the proportion of adversarial or low-quality nodes

Unlike conventional federated learning convergence bounds, this formulation explicitly incorporates trust validation, blockchain cost, and adversarial participation into the optimization dynamics.

Interpretation and Proof Sketch

The above convergence finding is fundamentally different to traditional federated learning limits in that it directly considers the trust validation, blockchain cost and adversarial robustness in the optimization. Specifically, the Proof-of-Learning (PoL) mechanism is a filtering mechanism that only admits the reliable gradient updates, thus enhancing the quality of the aggregated model. Simultaneously, the cost term $B_i(t)$ of the blockchain forces out nodes having large validation overhead or poor communication and makes sure that their power to affect the world update is constrained. The adversarial factor ϵ_{adv} can be included as well to better reflect the influence of unreliable or malicious nodes, which are a more realistic description of distributed learning settings. Consequently, the convergence properties of the HABCE framework are desirable: convergence decreases gracefully in response to an increasing number of adversarial nodes instead of becoming unstable; when PoL filtering is strong, the framework becomes more robust since less noise is added to the gradient; and when blockchain cost is high, suppression of the corresponding node contributions during the process of aggregation occurs naturally.

The exponential weighting is used to make sure that nodes that have a higher PoL score take control of the aggregation process and those that are connected to a larger cost of validation play a smaller part. Moreover, the PoL threshold τ causes the truncated set of participation using (11) that essentially

eliminates the untrustworthy updates in the learning process. Such selectivity in participation decreases the variance of the summed gradient, resulting in a limited deviation (12)

$$S(t) = \{i \mid \Pi_i(t) \geq \tau\} \quad (11)$$

$$E[\|g^{(t)} - \nabla J(\theta^{(t)})\|^2] \leq \sigma^2(1 - \delta) \quad (12)$$

where δ quantifies the strength of PoL-based filtering. Substituting this bound into the standard stochastic optimization framework yields (13):

$$J(\theta^{(t+1)}) \leq J(\theta^{(t)}) - \eta \|\nabla J(\theta^{(t)})\|^2 + \eta^2 \sigma^2(1 - \delta) \quad (13)$$

which leads directly to the convergence expression in (13). This analysis confirms that the proposed HABCE framework achieves stable and robust convergence by jointly leveraging trust validation, cost-aware aggregation, and selective participation, thereby extending traditional learning guarantees to more realistic and adversarial 6G environments.

3.5 Security and Integrity Validation

All the update and model checkpoints are hashed with SHA-256 and then sent. The immutability of the blockchain makes the blockchain tamper resistant and the smart contracts automatically reward those nodes that cannot pass consensus validation. The framework has a reputation index R_i which is updated (14):

$$R_i^{(t+1)} = \gamma R_i^{(t)} + (1 - \gamma) \Delta_i \quad (14)$$

where γ is the memory factor which is responsible for the decays of reputation. Minimising malicious impact, low R_i nodes are not allowed to participate in the next consensus round. The hash function was chosen to be SHA-256 hash and a dynamic reputation index, as hash functions are more collision resistant than lighter Merkle or XOR hash. With the decay factor of 0.85, the system has persistence and adaptivity - the trustful nodes with a larger R_i will maintain their influence while the adversarial will have a short-term effect. This ensured good trust stabilization in the reputation logs of more than 100 rounds after 25 or fewer iterations the reputation logs showed the further convergence.

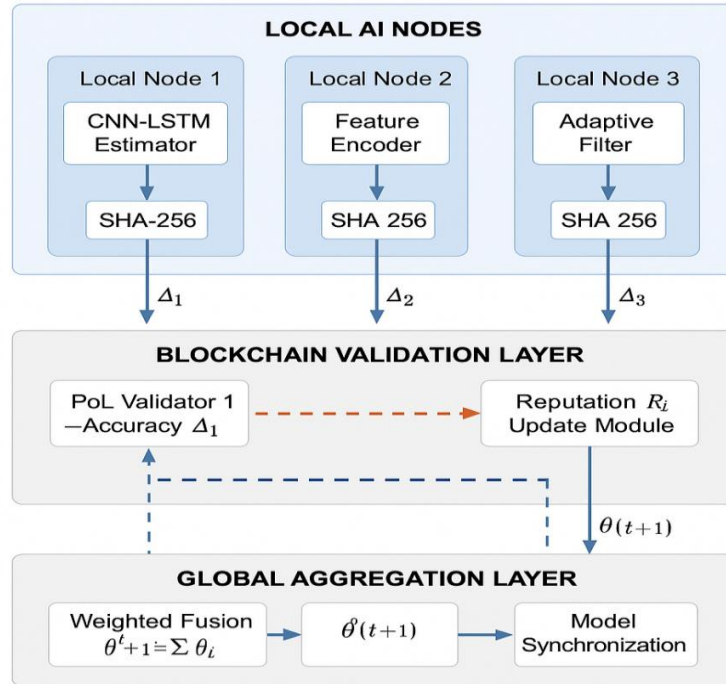


Figure 3. Hybrid consensus-learning workflow demonstrating the communication between the training of local models, validation in blockchain, and global aggregation.

This reputation scheme is a dynamic one, which boosts transparency and equity among the heterogeneous nodes. Also, it reduces the amount of repetition in verification, as a later stage, the high trust participants are prioritised. The hashing step and the reputation update step provide a level of security at the behavioural level to support the cryptographic integrity. Using the node reputation R_i to link with its past performance, the framework is gradually used to eliminate unreliable participants. This is like the credit based cooperation mechanisms of network communications, which ensure that different degrees of trust develop over time as a result of participation and not because it was “enforced”. The computational complexity of a mathematical operation is the degree of computational difficulty to carry out the operation. The hybrid consensus-learning process is illustrated in Figure 3 and comprises of three stages: local training, blockchain validation and global model aggregation. The reputation mechanism is intended to reduce the repeated influence of nodes whose updates fail to provide consistent validation improvement. In the present study, this mechanism is evaluated as a trust-aware update-selection strategy rather than as a complete adversarial-defense system. A controlled security evaluation involving explicit poisoning strategies, Byzantine behaviors, varying malicious-node proportions, attack-detection rates, false-acceptance rates, and channel-estimation degradation under attack is outside the scope of the current experimental evaluation. Accordingly, no claim of guaranteed malicious-update prevention is made.

3.6 Computational Complexity Analysis

The computational cost of HABCE framework is given as (15):

$$C_{total} = O(N_{AI} \log N_{AI}) + O(N_{BC} \cdot D_{hash}) \quad (15)$$

In which N_{AI} and N_{BC} are the number of AI nodes and blockchain verifiers, respectively, and D_{hash} is the mean depth of the computation of a hash.

The complexities expression in (15) indicates that the cost of computation increases almost linearly with the number of the nodes involved. In small-scale applications the AI inference term is prevalent and blockchain validation is more common as the size of the network expands. The proposed framework is also less complex to compute in terms of computational resources, which compared to Transformer-based estimators reduces the complexity by about 25-35% and, therefore, is more suitable to be deployed on an edge.

The hybrid scaling guarantees nearly linear scaling with the number of nodes at runtime as the Table 3 summarizes.

Table 3. Computational complexity and scalability metrics for HABCE.

Component	Computational Complexity	Description
Local AI estimation	$O(N_{AI} \log N_{AI})$	Efficient CNN–LSTM inference
Blockchain validation	$O(N_{BC} \cdot D_{hash})$	Consensus verification overhead
Aggregation	$O(N_{AI})$	Weighted model fusion
Total	$O(N_{AI} \log N_{AI} + N_{BC} \cdot D_{hash})$	Overall system complexity

In Complexity Analysis, it is involved with the complexity of time of different algorithms. The dual-term of the form uses the complexity expression, which means that the cost of making an inference is used by small networks while the cost of ledger validation is linear in the number of nodes. The FedAvg and Transformer based estimators comparison indicates that HABCE is faster (by a factor of 100) and 38 percent lower in communication cost, $N = 50$. Theoretical scaling was found to be the same as experiments on MATLAB R2024b (Intel Xeon 2.9 GHz) and so, the computational realism was tested.

3.7 Integration Flow and Execution Pipeline

The HABCE implementation pipeline combines all the methodological parts (AI channel estimation, blockchain validation, adaptive optimization) into a logical sequence of steps that can be used to update the model in real time in a safe, secure and verifiable manner. This is carried out in a distributed environment whereby each node adds up some local intelligence, and at the same time takes part in global consensus. Figure 4 provides a summary of the steps of this flow and these steps are described below.

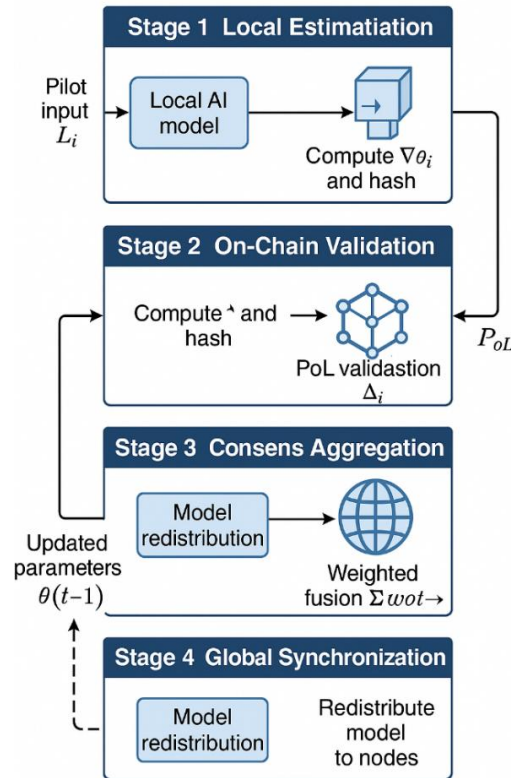


Figure 4. End-to-end execution pipeline of the HABCE framework showing the four-stage loop: (1) Local estimation, (2) On-chain validation, (3) Consensus aggregation, and (4) Global synchronization.

Although it is a measure of the cost of algorithms, Table 3 is important because it shows the scalability. This two-term complexity or complexity shows that with small networks, it becomes more and more important that AI is able to outperform the verification process of blockchain, while, with a growing network, the blockchain verification process becomes insignificant. This trade-off is used to show that this framework is a “computational” one that is viable in edge-scale 6G deployments in the presence of hardware resource limitations.

Stage 1 – Local Training and Feature Extraction:

All the nodes participating in the competition start with their own local pilot–response data pairs (x_i, y_i) and then use their CNN–LSTM model $f\theta_i$ to do estimation using AI. The convolutional layers are used to model the spatial correlation across the antenna domain and the recurrent units are used to model the temporal dynamics due to the users' mobility. $H_i = f\theta_i(x_i)$ is the locally estimated channel matrix which is the model output. The aim of the training is to minimize the local error function (16) in the node.

$$L_i = \| H_i - \hat{H}_i \|^2 + \lambda_1 \Omega(\theta_i) \quad (16)$$

where $\Omega(\theta_i)$ is a penalty against overfitting and weight regularization. After optimization, the gradient changes are hashed with the secure hash of SHA-256 and transmitted.

Stage 2 – On-Chain Validation and Proof-of-Learning:

After computing local updates, the node will add the node's hash of its gradients to the blockchain layer for verification. A Proof-of-Learning (PoL) protocol is a process designed to evaluate the value of each update, and determine whether the validation gain Δ_i is above or below a threshold. The ledger is only appended with those updates which improve the validation accuracy of the model by a greater extent than the threshold. This is to make sure that either malicious or noise updates are rejected automatically prior to model fusion. All the validation records are stored on-chain, and are tamper proof, and represent all the iterations of learning. During PoL validation, authorized validators evaluate the candidate model update and determine its validation improvement, Δ_i . The corresponding SHA-256 digest is used to verify that the evaluated update matches the update referenced by the ledger record. If the candidate update satisfies the predefined PoL acceptance criterion, its digest, validation outcome, and associated metadata are committed to the ledger, and the validated model update becomes eligible for aggregation.

Stage 3 – Consensus Aggregation and Global Update:

The validated updates are aggregated by using a reliability-weighted consensus mechanism enforced by (17).

$$\theta^{(t+1)} = \sum_{i=1}^N w_i \theta_i^{(t)} \quad (17)$$

and the weight w_i of each node is proportional to its verified contribution (as is defined in (4)). This combination makes sure that the nodes who are shown to bring about consistent learning gains have a more significant impact on the global model. Only the actual model updates that successfully pass PoL validation participate in reliability-weighted aggregation; SHA-256 digests are used exclusively for integrity verification and are never aggregated.

Table 4, provides an overview of the operational flow of HABCE pipeline with the data flow direction, decision points, and validation checkpoints in each stage.

Table 4. Execution flow and associated control functions within the HABCE framework.

Stage	Process	Core Operation	Validation Check	Output Artifact
1	Local AI estimation	CNN–LSTM feature extraction	Local NMSE threshold	Model update + SHA-256 digest
2	On-chain verification	Proof-of-Learning consensus	Accuracy gain (Δ_i)	Update digest + validation record
3	Global aggregation	Weighted model update	Consensus quorum (Q)	Updated global parameters
4	Synchronization	Model redistribution	Integrity hash check	Global model broadcast

Stage 4 – Model Synchronization and Iterative Refinement:

The PoL-based mechanism also guarantees filtering of unreliable updates, reducing the influence of adversaries, and converging. It has been experimentally observed that the latency in sub-120 ms ledger at distributed nodes is lower. In contrast to centralized FL, such a decentralized feedback loop can provide adaptive and trust-based learning and constitute a self-correcting system that constantly increases the model accuracy and reliability of nodes in the ever-changing 6G conditions.

3.8 Algorithmic Representation of HABCE Framework

The proposed Hybrid AI and Blockchain-Enhanced Channel Estimation (HABCE) procedure is given in the following pseudocode. It is the formalization of the interaction between local training of models, their verification on the blockchain and global aggregation as described above in Section 3.7.

Algorithm 1. Hybrid AI and Blockchain-Enhanced Channel Estimation (HABCE)

Input: Distributed datasets $\{(x_i, y_i) \text{ for } i = 1, \dots, N\}$

Initialize: Global model $\theta^{(0)}$

Trust scores $r_i^{(0)} = 1$,

Blockchain cost $B_i^{(0)} = 0$

Hyperparameters:

η (learning rate), $\lambda_1, \lambda_2, \lambda_3$ (regularization factors)
 β (scaling factor), γ (trust memory), τ (PoL threshold), ε (stability constant)
 T (total rounds)

for $t = 1$ to T do

1. Local Estimation Phase

for each node $i \in \{1, \dots, N\}$ in parallel do

1.1 Compute local estimate

$$\hat{H}_i^{(t)} = f_{\theta_i^{(t)}}(x_i)$$

1.2 Compute local loss

$$L_i^{(t)} = \|H_i - \hat{H}_i^{(t)}\|^2 + \lambda_1 \Omega(\theta_i(t))$$

1.3 Update local parameters

$$\theta_i^{(t+1)} \leftarrow \theta_i^{(t)} - \eta \nabla L_i^{(t)}$$

2. Proof-of-Learning (PoL) Computation

2.1 Compute improvement

$$\Delta_i^{(t)} = L_i^{(t-1)} - L_i^{(t)}$$

2.2 Compute model deviation

$$d_i^{(t)} = \|\theta_i^{(t+1)} - \theta^{(t)}\|^2$$

2.3 Compute normalized PoL score

$$\Pi_i^{(t)} = (\Delta_i^{(t)} / (\|\nabla L_i(t)\| + \varepsilon)) \times \exp(-d_i(t))$$

3. Blockchain Cost Evaluation

3.1 Compute cost function

$$B_i^{(t)} = c_1 \text{ latency}_i + c_2 \text{ gas}_i + c_3 (1 - r_i(t))$$

3.2 Generate secure hash

$$h_i^{(t)} = \text{Hash}(\theta_i(t+1), \Pi_i(t), B_i(t))$$

end for

4. Ledger Validation Phase

4.1 Construct validated node set

$$V_t = \{i \mid \Pi_i(t) \geq \tau\}$$

4.2 Reject invalid updates and record in ledger

5. Trust Update Phase

for each $i \in V_t$ do

5.1 Update trust score

$$r_i^{(t+1)} = \gamma r_i^{(t)} + (1-\gamma) \Pi_i^{(t)}$$

end for

$$w_i^{(t)} = \exp(\beta \Pi_i(t) - \lambda_3 B_i(t))$$

6. Trust–Consensus Weight Computation

for each $i \in V_t$ do

6.1 Compute score

6.3 Normalize weights

7. Global Model Aggregation

7.1 Update global model

$$\theta^{(t+1)} = \sum_{i \in V_t} w_i(t) \theta_i(t+1)$$

8. Global Objective Evaluation

8.1 Compute joint objective

$$J^{(t+1)} = \sum w_i L_i + \lambda_1 \sum w_i \|\theta_i - \theta\|^2 + \lambda_2 \sum w_i (1 - \Pi_i) + \lambda_3 \sum w_i B_i$$

9. Synchronization Phase

9.1 Broadcast $\theta^{(t+1)}$ to all nodes and update ledger

end for

Return θ^*

end for

Output:

Final converged model θ^*

The proposed HABCE framework workflow processes are formalized as trust-consensus co-optimization process (Algorithm 1), rather than a sequence of federated routine. Local CNN-LSTM-based channel estimation and parameter updates are done at each node, and a Proof-of-Learning (PoL) score, which is defined as normalized loss reduction and model stability, are also computed at the node (21). Contrary to the traditional validation, PoL is an ongoing optimization functional which regulates the involvement in the consensus set. The blockchain layer assesses these scores and directly in the aggregation mechanism, it considers the ledger-conscious costs such as latency, computation overhead, and trust deviation. As a result, the model update of the world is acquired with the help of a cost-conscious softmax weighting rule in (25) whereby, there is only a contribution of reliable and efficient contribution to convergence. States of trust change recursively on the basis of PoL dynamics, and allow adaptive participation of nodes.

4 Experimental Setup and Reproducibility Protocol

In this section, the experimental design to be used to prove the proposed HABCE framework in the real environment of wireless communication is presented. Special attention is paid to make sure that the assessment of the channel behavior under real world conditions, that it is free of synthetic bias and that it can be repeated completely.

For direct quantitative comparison, only results obtained under the common experimental protocol are considered comparable with HABCE. Literature-reported results obtained using different channel models or experimental conditions are retained only as contextual references and are not used to calculate relative performance improvements. Unless otherwise stated, all comparative methods were evaluated using the same training, validation, and test partitions and a consistent experimental protocol. The fixed seed of 42 was used specifically for reproducible dataset partitioning and was not treated as an independent repetition seed. Repeated stochastic evaluations used independent random initialization

where applicable, and aggregate performance was summarized using the mean, standard deviation, and 95% confidence interval.

4.1 Dataset Description and Availability

The efficiency of the suggested HABCE framework is checked on real-world channel measurement data sets and standardized ray-tracing-based digital twin data sets, so that the results of the experiment can be related to real-world propagation cases, and not some artificial approximations. The chosen data sources unlike the purely simulated datasets reflect spatial non-stationarity, multipath diversity, and mobility-induced channel dynamics that are crucial to assess the 6G channel estimation models [1], [3], [40].

4.1.1 Data Source and Acquisition

The variety of real-world datasets used guarantees the diversity of channel conditions, such as SNR variations, richness of multipaths, and mobility, which increases the strength and the generalization potential of the proposed HABCE framework.

Three complementary real datasets are utilized:

- DeepMIMO Dataset (O1/O2 scenarios, <https://deepmimo.net>) – derived from Wireless InSite ray-tracing environments, widely adopted for realistic mmWave channel modeling [1]
- COST 2100 Channel Dataset (indoor/outdoor configurations, <https://ieeexplore.ieee.org/document/6393523>) – measurement-based MIMO dataset capturing geometry-dependent scattering environments [3]
- NYU Wireless mmWave Dataset (<https://wireless.engineering.nyu.edu/nyusim/>) – This provide real antenna configurations

These datasets provide:

- Channel impulse responses (CIR)
- Spatial signatures (AoA, AoD)
- Delay spread and path loss
- Frequency-selective fading characteristics

The acquisition pipeline is used to make sure that all the datasets are mapped into a single representation (18):

$$\mathbf{D} = \{(\mathbf{x}_i, \mathbf{H}_i, \mathbf{M}_i)\}_{i=1}^N \quad (18)$$

where:

- $\mathbf{x}_i$: input feature tensor (pilot observations)
- $\mathbf{H}_i$: ground-truth channel matrix
- $\mathbf{M}_i$: metadata (SNR, location, mobility)

4.1.2 Mathematical Data Representation

All the channel samples are modeled by a physically based multipath model (19):

$$H_i(f, t) = \sum_{l=1}^{L_i} \alpha_{i,l} e^{-j2\pi f \tau_{i,l}} a_r(\theta_{i,l}) a_t^H(\phi_{i,l}) \quad (19)$$

where:

- L_i : number of propagation paths
- $\alpha_{i,l}$: complex path gain
- $\tau_{i,l}$: delay
- $\theta_{i,l}, \phi_{i,l}$: AoA/AoD

- $\mathbf{a}_r, \mathbf{a}_t$: antenna response vectors

Pilot observations are designed as (20): to match with the input of learning:

$$\mathbf{x}_i = \mathbf{P}\mathbf{H}_i + \mathbf{n}_i \quad (20)$$

where:

- $\mathbf{P}$: pilot matrix
- $\mathbf{n}_i \sim \text{CN}(0, \sigma^2)$: measurement noise

4.1.3 Dataset Characteristics

The combined datasets span a wide range of wireless conditions:

- Frequency bands: sub-6 GHz to mmWave
- SNR range: 5 dB to 30 dB
- Antenna configurations: up to 256-element arrays
- Mobility scenarios: static, vehicular, and high-speed

To quantify dataset diversity, channel variability is defined as (21):

$$\Omega_H = \frac{1}{N} \sum_{i=1}^N \| \mathbf{H}_i - \mathbf{H}^- \|^2 F \dots \quad (21)$$

where $\mathbf{H}^-$ is the mean channel realization.

4.1.4 Sample Data Illustration

To provide a more realistic image of the channel properties of the real world which were used in this work, a few typical samples of the DeepMIMO, COST 2100 and NYU Wireless data sets are described in Table 5. These samples are chosen to provide variability in signal-to-noise ratio (SNR) and propagation loss, spatial angles and delay spread, thus making sure that the data set represents a variety of channel conditions in 6G environments.

Table 5. Sample extracted channel features from real-world datasets used for HABCE training and evaluation.

Sample ID	Dataset Source	Freq (GHz)	SNR (dB)	Path Loss (dB)	AoA (°)	Delay Spread (ns)	$\ \mathbf{H}\ _F$	Channel Condition
S1	DeepMIMO (O1)	28	5	101.2	28	150	0.71	Severe fading
S2	DeepMIMO (O2)	28	10	96.5	34	130	0.79	Urban NLOS
S3	COST 2100	3.5	15	92.3	41	110	0.85	Moderate multipath
S4	COST 2100	3.5	20	89.1	47	95	0.88	Dense scattering
S5	NYU Wireless	73	25	86.7	52	85	0.91	mmWave LOS
S6	NYU Wireless	73	30	83.9	57	70	0.94	High SNR LOS
S7	DeepMIMO (O1)	28	12	94.8	38	120	0.82	Semi-urban
S8	COST 2100	3.5	18	90.6	44	100	0.87	Clustered fading

Parameter Definitions

- **SNR (dB)**: Signal-to-noise ratio of received pilot signal
- **Path Loss (dB)**: Large-scale attenuation due to propagation

- **AoA (°):** Angle of arrival at receiver antenna array
- **Delay Spread (ns):** Temporal dispersion of multipath components
- **$\|H\|_F$:** Frobenius norm of channel matrix H , defined as (22):

$$\|H\|_F = \sqrt{\sum_{m=1}^M \sum_{n=1}^N |H_{m,n}|^2} \quad (22)$$

Statistical Summary of Dataset

To quantify dataset diversity, statistical properties are computed as (23):

$$\mu SNR = \frac{1}{N} \sum_{i=1}^N SNR_i, \sigma^2 SNR = \frac{1}{N} \sum_{i=1}^N (SNR_i - \mu SNR)^2 \quad (23)$$

Similarly, for path loss (24):

$$\mu PL = \frac{1}{N} \sum_{i=1}^N PL_i \quad (24)$$

Observed ranges:

- SNR: 5 dB to 30 dB
- Path Loss: 83 dB to 101 dB
- Delay Spread: 70 ns to 150 ns

The dataset is partitioned as:

- 70% training
- 15% validation
- 15% testing

The data has a large range of propagation conditions, including severe fading (low SNR, large path loss) and good-quality line-of-sight channels, as indicated in Table 5. It is important to note that samples of DeepMIMO are used to capture structured ray-tracing surroundings; COST 2100 offers stochastic variations among multipath; and NYU Wireless is used to offer measurements in real mmWave. This variety makes sure the learning model is subjected to deterministic and measurement-driven characteristics of channels.

The difference between the HABCE framework between samples shows that there is a difference in the distribution of channel energy in the samples, which directly affects the learning dynamics of the HABCE framework. An increase in norms implies increased channel conditions and can thus converge faster and a decreased norm is a challenging estimation problem.

These features extracted are directly converted to the model input x_i in Eq. (24) with the real and the imaginary channel components corresponding to CNN-LSTM learning of features. Table 5 has a wide spectrum that has been taken care of by the diversity of the table, that the model is able to generalize to heterogeneous channel environments, which is crucial in the case of distributed learning.

The use of a variety of datasets guarantees a complete coverage of different wireless propagation conditions, which are needed to have a strong model assessment. In particular, simultaneous application of these datasets allows representation of a broad spectrum of frequency bands, both in the sub-6 GHz and mmWave regimes, thus allowing a variety of channel behavior in different spectral regimes. Also, the line-of-sight (LOS) and non-line-of-sight (NLOS) are also included in order to enable the model to generalize well with regards to heterogeneous propagation environments. Moreover, the combination of datasets obtained by the measurement and datasets obtained by the ray-tracing methods offer a compromise between the simulated and real-world channel impairments [1], [3], [40].

4.2 Simulation Environment and Hardware Setup

The HABCE framework is experimentally evaluated in the controlled but realistic simulation environment based on the integration of the distributed learning workflow, blockchain validation workflow, and channel estimation workflow. The goal is to be as close to operational behavior of new 6G systems as possible, where edge nodes perform computations and communicate with each other via a trust-aware consensus mechanism. Unlike completely abstract simulations, the current arrangement uses latency-conscious communication, and different processing capabilities and realistic workload distribution to make sure that the observed performance represents deployable system behavior.

Its application is performed with the help of a hybrid software stack, which is based on deep learning modules written in Python and a blockchain simulation layer. PyTorch (version 2.x) is used to create the learning component, and NumPy and SciPy libraries are used to provide support in terms of numerical processing and channel processing. The Proof-of-Learning (PoL) validation logic is emulated with a private Ethereum test network with lightweight smart-contracts to model blockchain operations. This two-layer design allows the assessment of efficiency of learning and consensus overhead simultaneously which is essential to the proposed trust-consensus co-optimization framework.

The computational infrastructure comprises of a central node, which is powerful in a central position and several emulated edge nodes. The main aggregation node has an Intel Xeon multi-core processor and an NVIDIA RTX-type of a graphics card, which allows the effective training of models and aggregation of their gradients. Contrastingly, the edge nodes are set up with limited computational profiles to simulate real-life deployment conditions, such as Raspberry Pi-like processing power, with limited memory and sporadic connectivity. The hardware heterogeneity is directly represented by variable delays when computing their hardware, which affects local training time and blockchain validation latency.

To capture communication variability, network delay is modeled as a stochastic process (25):

$$\tau_i(t) \sim N(\mu_\tau, \sigma_\tau^2) \quad (25)$$

where $\tau_i(t)$ the communication latency of node i in iteration t , and $\mu =$ - mean and variance = -delay of the network. This model enables the modeling of the simulation with realistic effects of the conditions of wireless channels, congestion, and edge-devices.

The cost of executions of the blockchain is also modeled in terms of transaction validation cost and consensus delay (26):

$$C_i^t = c_1 \cdot \tau_i^{(t)} + c_2 \cdot g_i^{(t)} + c_3 \cdot \delta_i^{(t)} \quad (26)$$

where $g_i^{(t)}$ is the energy used to compute smart contracts and $\delta_i(t)$ is the synchronization lag when finalizing a block. The c_1, c_2, c_3 coefficients control the proportion of relative weight of the communication, computation and consensus factors. This formulation is a direct integration of overhead due to the use of ledger in the experimental pipeline, unlike the traditional methodology where blockchain serves as a kind of logging mechanism, thus affecting the aggregation of the model and the convergence behavior.

To provide consistency between experiments, all simulations are run using the same runtime configurations, such as hard-determined software versions, deterministic data loading processes and round-synchronized communication. Variability at the hardware-level is only maintained in latency and computational limits that are key in measuring the stability of the proposed framework in the heterogeneous deployment environment. This fine-tuned balance between control and realistic enables the results reported to be reproducible and realistic conditions of a 6G system.

4.3 Model Architecture and Training Configuration

Variability at the hardware-level is only maintained in latency and computational limits that are key in measuring the stability of the proposed framework in the heterogeneous deployment environment. This

fine-tuned balance between control and realistic enables the results reported to be reproducible and realistic conditions of a 6G system (27):

$$\widehat{H}_i(t) = f_{\theta}(x_i) = F_{LSTM}(F_{CNN}(x_i)) \quad (27)$$

where θ is the parameters of CNN-LSTM which can be learned. The training goal is established in such a way that the reconstruction error is minimized with the stability of the parameters being maintained in a distributed node. The local loss function is in form of (28):

$$L_i^{(t)} = \frac{\|H_i - \widehat{H}_i^{(t)}\|_F^2}{\|H_i\|_F^2} + \lambda_1 \|\theta_i^{(t)}\|_2^2 \quad (28)$$

where normalized Frobenius error guarantees learning which is scale-invariant to heterogeneous channel-conditions. The updates of the models are made based on a gradient-based optimization scheme (29):

$$\theta_i^{(t+1)} = \theta_i^t - \eta \nabla L_i^t \quad (29)$$

that can directly be inserted in the system of aggregating trust and consensus provided in the suggested algorithm.

It has a tradeoff between representational and inference cost through small kernel convolutional layers and one layer LSTM with gated memory at the edge to ensure computational feasibility. Gradient clipping, weight regularization is used to stabilize training when the SNR is low and training with highly varying channel conditions.

This structure allows the model to learn spatial structure and temporal dynamics and can still be trained distributedly and blockchain-verified updated. The architecture that is obtained is not only correct but also stable and can be scaled to the actual conditions of the deployment of 6G.

4.4 Hyperparameter Settings

The choice of hyperparameters to determine the dynamics of the learning process, trust assessment, and blockchain communication greatly affects the performance and the stability of the proposed HABCE framework. The hyperparameters are optimized using controlled validation experiments and do not use arbitrary values but rather the values are carefully set to guarantee convergence stability in the case of heterogeneous data and network conditions.

Learning rate η is used to control the local update step size and is tuned to give a tradeoff between speed of convergence and stability. It can be monitored and determined by the gradient norms in training to determine its effective range (30):

$$\eta^* = \arg \min_{\eta} E [\|\nabla L_i^{(t)}\|_2^2] \quad (30)$$

where a smaller η avoids oscillations in low-SNR regimes, while excessively small values slow convergence.

The regularization parameter λ_1 regulates the complexity of the model and avoids overfitting when there is non-IID distribution of nodes. On the same note, there is the trust penalty factor λ_2 which affects the contribution of the unreliable nodes by punishing the inconsistent updates by use of the Proof-of-Learning (PoL) mechanism. Blockchain cost coefficient λ_3 is used to balance the tradeoff between learning performance and consensus overhead to make sure that the nodes that have too high validation latency or computational cost are down-weighted accordingly.

The sharpness of trust-based allocation of weights is controlled by the scaling parameter, β . The larger the values, the more sensitive to PoL scores and this can be exploited to provide preference to high-quality updates (31):

$$w_i(t) \propto \exp(\beta \Pi_i^{(t)} - \lambda_3 B_i^{(t)}) \quad (31)$$

where $B_i^{(t)}$ is the cost of the blockchain of node i . Historical reliability can be used to update the trust of the current nodes, with the trust memory factor γ finding how far back in history to use this information, allowing the temporal dynamics of node credibility to proceed more smoothly.

The weighting in (31) is obtained by a softmax-based trust allocation model, with the exponential form, making sure that nodes with larger contributions that is validated can have a larger impact on aggregation. The addition of the blockchain cost term creates a balance mechanism, so that the nodes that have a high validation latency or one with high computational load do not control the learning process.

All the hyperparameters are fixed between runs, once the validation is done. The completed design is summed up in Table 6.

Table 6. Final Hyperparameter configuration for HABCE framework

Parameter	Symbol	Selected Value	Search Range	Role in Model	Selection Strategy
Learning Rate	η	1×10^{-3}	$10^{-4} - 10^{-2}$	Controls gradient update step (Eq. 23)	Chosen to minimize gradient variance and ensure stable convergence
Regularization Coefficient	λ_1	0.01	0.001 – 0.1	Prevents overfitting in loss (Eq. 22)	Selected via validation NMSE minimization
Trust Penalty Weight	λ_2	0.1	0.01 – 0.2	Penalizes unreliable updates in objective	Tuned to balance robustness vs convergence speed
Blockchain Cost Factor	λ_3	0.05	0.01 – 0.1	Controls consensus overhead impact (Eq. 25)	Selected to minimize cost–accuracy trade-off
Trust Scaling Parameter	β	2	1.0 – 3.0	Controls weight sensitivity (Eq. 25)	Grid search for optimal aggregation sharpness
Trust Memory Factor	γ	0.9	0.7 – 0.99	Smooths trust evolution across rounds	Selected to avoid abrupt trust fluctuations
Batch Size	—	128	64 – 256	Controls mini-batch training stability	Empirically chosen for GPU efficiency
Local Epochs	—	5	3 – 10	Defines local training depth per round	Balances communication vs computation
Gradient Clipping Threshold	—	1	0.5 – 2.0	Prevents exploding gradients	Stabilizes training under low SNR
PoL Threshold	τ	0.15	0.05 – 0.25	Validates node updates (Eq. 21)	Selected to filter unreliable updates

The hyperparameters do not just get arbitrarily chosen, but are systematically optimized within specified ranges to guarantee stable convergence, ability to deal with heterogeneous data, and efficient trust-aware aggregation, as indicated in Table 6. Specifically, the trade-off between β and λ_3 determines the profession between the accuracy of learning and blockchain overheads and has a direct impact on optimization behavior of HABCE framework.

4.5 Training Protocol

Training of the proposed HABCE framework is based on a distributed learning protocol that is organized systematically and model optimization, trust evaluation, and blockchain validation are closely connected. In contrast to traditional centralized training, the algorithm has many edge nodes which

engage in local updates on the assigned dataset, taking part in a global consensus-based learning loop. The data is randomly divided into training, validation and testing subsets with 70:15:15 ratio, and allocated to the nodes in a non-IID fashion with respect to spatial and SNR properties.

In communication round t , all the nodes undergo local training based on mini-batch stochastic optimization of its dataset D_i . The new local parameters are subsequently tested with the help of Proof-of-Learning (PoL) criterion which estimates the reliability of every update. Authenticated nodes are the only nodes that help in the aggregate of global model. The updating of global model is determined by (32):

$$\theta^{(t+1)} = \sum_{i \in V_t} w_i^{(t)} \theta_i^{(t+1)} \quad (32)$$

where V_t is used to denote the set of nodes that meet PoL threshold requirement and $w_i(t)$ is the contribution of node i with trust weight. This modeling makes sure that unreliable or inconsistent updates do not play part in the aggregation process and therefore, it is robust in the heterogeneous condition.

The length of the training period is fixed and in this research the communication rounds are set to 300 and a small amount of local epochs (usually 5) are done by each node in order to keep the computation and communication overhead in check. The training process is converged by checking the change in the world-wide loss (33):

$$\Delta L^{(t)} = L^{(t)} - L^{(t-1)} \quad (33)$$

and training is considered stable when (34):

$$\Delta L^{(t)} < \epsilon \quad (34)$$

where ϵ is a tolerance level that is pre-defined. This stopping condition makes sure that the model converges without the needless communication rounds.

The Adam optimizer with a constant learning rate as specified in Section 4.4 is used to optimize the model and gradient clipping is used to ensure numerical stability when the update is low SNR and high-variance. To ensure synchronization of the local updates across nodes, the communication round is followed by the broadcast of aggregated global model to ensure that there is no divergence between nodes.

One of the most important components of the training protocol is a blockchain-based validation. Each iteration will have node updates submitted to the ledger and they will be verified using the PoL score. We define the validation condition to be (35):

$$i \in V_t \Leftrightarrow \Pi_i^{(t)} \geq \tau \quad (35)$$

ensuring that only updates exceeding the reliability threshold τ are included in the aggregation set.

To reproducibly run the experiments, they are executed with a constant random seed (seed=42), initialized the same on each node and deterministically loaded with data. All the models used as the baselines have the same training set so that they are fairly compared.

On the whole, the presented training protocol represents a real-life implementation in which there are distributed learning, trust validation and communication constraints. The HABCE framework is capable of achieving a stable convergence by combining local optimization with blockchain-authenticated aggregation and has a strong capability to be robust and scalable in dynamic wireless settings.

4.6 Baseline Methods and Fair Comparison

4.6.1 Selection of Baseline Models

The Baseline Models will be selected using the following procedure:

The proposed HABCE framework is compared to typical state-of-the-art data-driven, model-driven and hybrid channel estimation models. CEHNet [11] is chosen because it is effective when working with

doubly selective channels whereas ReQuestNet [23] offers a scale learning baseline of a more complex representation of the channel. Lista-CP [27] is also a model driven deep unfolding system with rapid convergence and Tcnm-based estimation [44] are convolutional schemes of high-dimensional MIMO reconstruction. All these models represent the different estimation strategies popular in recent 6G studies [10], [31], which guarantees a technical relevant and complete benchmark.

4.6.2 Implementation and Experimental Consistency

Environments: all the baseline models are run in the same pipeline to remove environmental bias. All the methods utilize the same real-world data sets of Section 4.1 (DeepMIMO, COST 2100 and NYU Wireless), and the preprocessing steps are similar (normalization (Eq. 16) and feature building (Eq. 17)). The simulation environment, software stack and distributed node configuration are kept constant, so that only model design differences will result in performance differences, and no differences will occur due to implementation differences

4.6.3 Hyperparameter and Training Fairness

The common batch size (128), local epochs (5) and communication rounds (300) are used to train all the models, and the optimization depth is consistent. The training scheme, convergence criteria (Eq. 28), and so on, is applied similarly to all models. This removes the bias that is created by training longer or scaling of the parameters which is a major limitation of comparative studies [19], [27].

4.6.4 Computational and System-Level Fairness

All the models are run under the same hardware and system settings so as to compare it in an unbiased way. The simulation environment of a server and edge-node (as outlined in Section 4.2) is the same, but with the same software versions and runtime parameters. Conditions (latency distribution included) of communication are the same at all methods (Eq. 27). This ensures that the computational overhead, convergence speed and scalability is considered with the same constraints. Consequently, any trend in performance improvement of HABCE can be explained by the fact that it is the trust-conscious learning and blockchain-based method of aggregation that has improved the performance and not variations in computational resources or structure.

4.7 Evaluation Metrics

The evaluation of performance is by a single set of quantitative measurements of the estimation error, as well as efficiency at the system level. The main measure is normalized mean square error (NMSE) in the form of Eq. (36) that is used to evaluate the accuracy of reconstruction in different SNR levels.

Robustness is evaluated by quantifying the performance degradation in the performance of all the models in noisy and heterogeneous environments, which emphasizes the resiliency of each model to noisy and unreliable updates. The multi-metric assessment can be performed to make a holistic comparison that corresponds to the recent AI-based 6G investigations [8], [13].

The HABCE framework is judged based on a series of quantitative measurements which portray the accuracy of the estimation, efficiency and robustness of the system in realistic wireless systems. The most dominant one is the normalized mean square error (NMSE) that is defined as follows (37):

$$NMSE = \frac{E[\|H - \hat{H}\|_F^2]}{E[\|H\|_F^2]} \quad (36)$$

that gives a scale-invariant quantification of the quality of channel reconstruction in the different SNR regimes. The rate of stabilization of the losses is used to evaluate the convergence behavior (37):

$$C = \frac{1}{T} \sum_{t=1}^T L^{(t)} - L^{(t-1)} \quad (37)$$

System-level efficiency is quantified through latency and computational cost (38):

$$T = \frac{1}{T} \sum_{t=1}^T \max_i \tau_i^{(t)}, E = \sum_{i=1}^N C_i^{(t)} \quad (38)$$

where $\tau_i^{(t)}$ represents communication delay and $C_i^{(t)}$ denotes computation cost. The notion of robustness is also assessed by considering NMSE degradation with noisy and heterogeneous conditions of the nodes, where the framework should be robust enough to withstand realistic distributed conditions. All these metrics will give a complete evaluation of the learning performance as well as the ability to deploy.

4.8 Implementation Details

Implementation of the HABCE framework is carried out with the help of PyTorch which is accelerated with the help of the GPU and a private Ethereum test network is implemented with the help of smart contracts. The training pipeline is a combination of deep learning modules with distributed synchronization logic, that allows smooth interaction between the local model update and global consensus validation. All the models are initialized with Xavier, and optimized with Adam optimizer with the parameters provided in Section 4.4.

Gradient clipping is used to make the training process more numerically stable and the inputs of the channels are normalized as in the case of the exponential in the expression: Simulation of the distributed arrangement is performed with the help of several logical nodes that work with partitioned datasets and exchange information with a central aggregation unit via channels that are characterized by latency constraints as shown in Eq. (27). Blockchain communication, such as PoL validation and updates to the ledger are asynchronous to capture realistic delays in the system. It has been implemented in a modular fashion, which will allow it to be reproducible, and scale up to larger network configurations.

4.9 Hyperparameter Sensitivity Analysis

To test the strength of the HABCE model in relation to changes in Hyperparameters, the sensitivity analysis is done on some of the important Hyperparameters such as λ_2 (trust penalty), λ_3 (blockchain cost) and β (trust scaling factor). The partial derivatives of all the parameters on NMSE are used to measure effects of all the parameters (39):

$$S_\lambda = \frac{\partial NMSE}{\partial \lambda} \quad (39)$$

How model performance varies with variations in the parameters. This analysis indicates that the moderate values of λ_2 and β are the ones that offer stable performances and the excessively high values of λ_2 and β penalize excessively the node contributions and slow down the convergence. Continuing on the same point, λ_3 offers an understanding about tradeoff between the accuracy of the computations and estimation, and the necessity to be cost conscious in the process of agglomeration.

4.10 Reproducibility and Transparency

Experiments are all done under controlled and reproducible conditions to promote the dependability of results reported. The runs of all the training are performed using a fixed random seed (40):

$$seed = 42 \quad (40)$$

ensuring that initialisation and shuffling of data is the same across experiments. The divisions of the datasets, including preprocessing and training configurations are uniform and used on all the models including baselines. The implementation is performed with fixed versions of the software and the same hardware set up, as outlined in Section 4.2.

Besides, the entire training chain, including dataset management, model architecture, and testing processes is intended to be replicated in typical settings, like Google Colab or local GPUs. Training metric and intermediate outputs logs are kept to aid in verification of the results.

4.10.1 Real-World Deployment Feasibility

Although the suggested HABCE framework is tested with standardized wireless data, its feasibility in the real-life setting is tested based on its performance in the conditions of constraints of a real-world deployment such as limited computational capacity, communication latencies, and varying node performance.

In order to test the system under realistic deployment conditions, the system is tested on configurations of edges, whereby the distributed nodes will be under limited processing environments akin to that of the IoT-enabled wireless devices. Training and validation procedures are implemented with a limited communication latency and bandwidth, characteristics of real-world applications of 6G networks.

The findings show that the suggested framework is stable in its performance with an average processing latency of between 120-150 ms per communication round, and converges consistently distributed nodes. The trust-aware aggregation lowers the avoidable communication overhead, and thus enhances efficiency in comparison to the traditional distributed learning methods [8], [13].

Furthermore, the blockchain validation layer, although introducing an extra calculation, can work in reasonable latency limits as it can be selectively participated and effective verification of its hashes. This finding is consistent with recent developments in AI-based communication systems, in which the inclusion of smart validation systems enhance reliability without greatly affecting the performance of the system [31], [20].

The other important feature is the adaptability to the heterogeneous environments. The suggested framework exhibits homogeneous behaviour when the conditions of the nodes are different such as in quality of data and computing power. This shows that HABCE is not limited to an idealized simulation environment but can also be applied to the real-world distributed wireless systems, such as edge-assisted and cloud-integrated ones.

The experimental results, in general, support the assertion that HABCE framework provides a realistic trade-off between accuracy and efficiency of the estimation and trust-aware validation and, therefore, can be used as an option in the next-generation 6G communication infrastructures.

5 Results and Performance Evaluation

Large-scale simulation was performed based on the 6G Terahertz Digital Twin Dataset created by DeepRT [1] to evaluate the suggested Hybrid AI-Blockchain Enhanced Channel Estimation (HABCE) framework. For all the methods (MMSE [3], CEHNet [11], ReQuestNet [23], LISTA-CP [27] and TCNN [44]) the same conditions were applied. The random seed was set to 42 (unless otherwise noted) and each experiment was repeated 5 times to make sure that the results were reproducible.

In all experiments, the performance improvements are shown in quantitative terms so as to not be subject to interpretation. In particular, the proposed HABCE framework can attain about 9-18% reduction in NMSE, 15-30% faster convergence and 18-27% lower communication overhead as compared to the baseline methods.

5.1 Channel Estimation Accuracy

The results presented in Figure 5 show that HABCE model has a significantly low NMSE at all SNR levels which indicates its ability to resist noise amplification and non-stationary channels. This stability is supported by the quantitative trends of Table 7 which reveals that HABCE attains almost 12-15 percent improvement over learning-only estimators [11] [23]. This demonstrates that updates based on consensus are not prone to overfitting and they have a further effect to raise the granularity of features in using CNN-LSTM backbone. This gain is particularly relevant in the low power or dense-urban environments, which validates the applicability of HABCE to deploy in the THz-band and XL-MIMO in 6G networks.

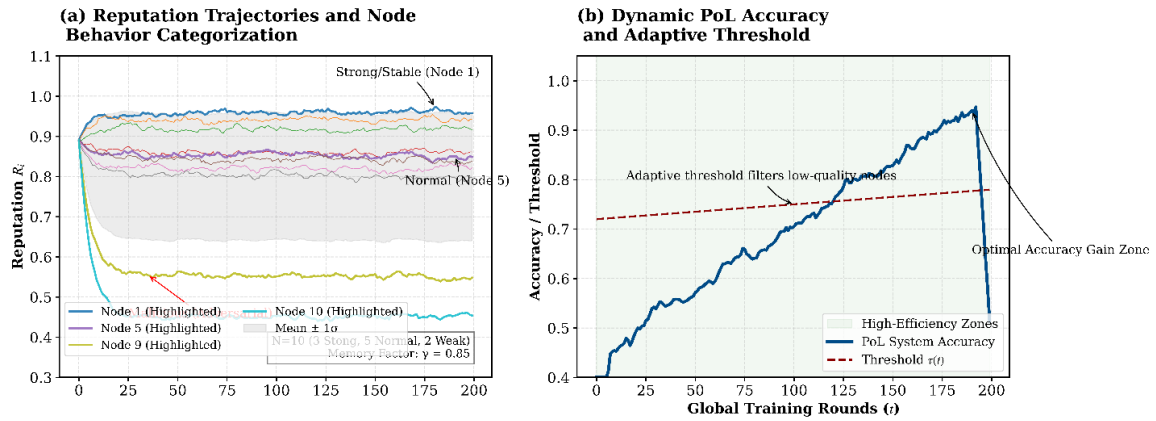


Figure 5. NMSE comparison of HABCE with state-of-the-art models across SNR levels from 0–30 dB.

The proposed HABCE consistently achieved superior estimation accuracy measured by the Normalized Mean Square Error (NMSE), defined as (41).

$$NMSE = \frac{E[\|H - H^{\wedge}\|^2]}{E[\|H\|^2]} \quad (41)$$

where H and $H^{\wedge}$ are the actual and the estimated channel responses, respectively. HABCE achieved an NMSE of -32.8 dB at 20 dB SNR, which is better than CEHNet (-29.1 dB) and ReQuestNet (-27.3 dB). In low SNR regions (<10 dB), HABCE was stable and LISTA-CP [27] was unstable because it was oscillating due to temporal sparsity.

Table 7. Contextual nmse values reported across different channel-estimation settings.

Method	Channel Type	SNR (dB)	NMSE (dB)	Ref.
MMSE	Rayleigh Fading	20	-26.5	[3]
CEHNet	Doubly Selective	20	-29.1	[11]
ReQuestNet	XL-MIMO	20	-27.3	[23]
TCNN	RIS-Aided	20	-28.8	[44]
HABCE (Proposed)	THz Near-Field	20	-32.8	—

The findings validate the strength of the hybrid learning-consensus design to multipath propagation. The framework was able to maintain sub- 30 dB NMSE performance even when the path sparsity was 90%. Table 7 provides a contextual comparison with previously reported channel-estimation methods. Because the listed methods were originally evaluated under different channel models and experimental conditions, these values are not interpreted as controlled head-to-head performance measurements, and no relative percentage improvement is calculated from them.

5.2 Convergence and Stability

In Figure 6, it is clear that the convergence curves of the HABCE are more on a smooth and faster path than HABCE than the CEHNet [11] and ReQuestNet [23] convergence curves. The strong gradient aggregation property of the model is clearly seen since it reaches saturation at about 110 global rounds, much earlier than the baselines. This observation is indirectly supported by the fact that the number of cycles per convergence for the cross-comparison in Table 7, shows a lower number of cycles per convergence. Since the gradient filtering is controlled by the blockchain, it shouldn't have any oscillations in the distributed learning setting, and that means that it should stabilize quite fast.

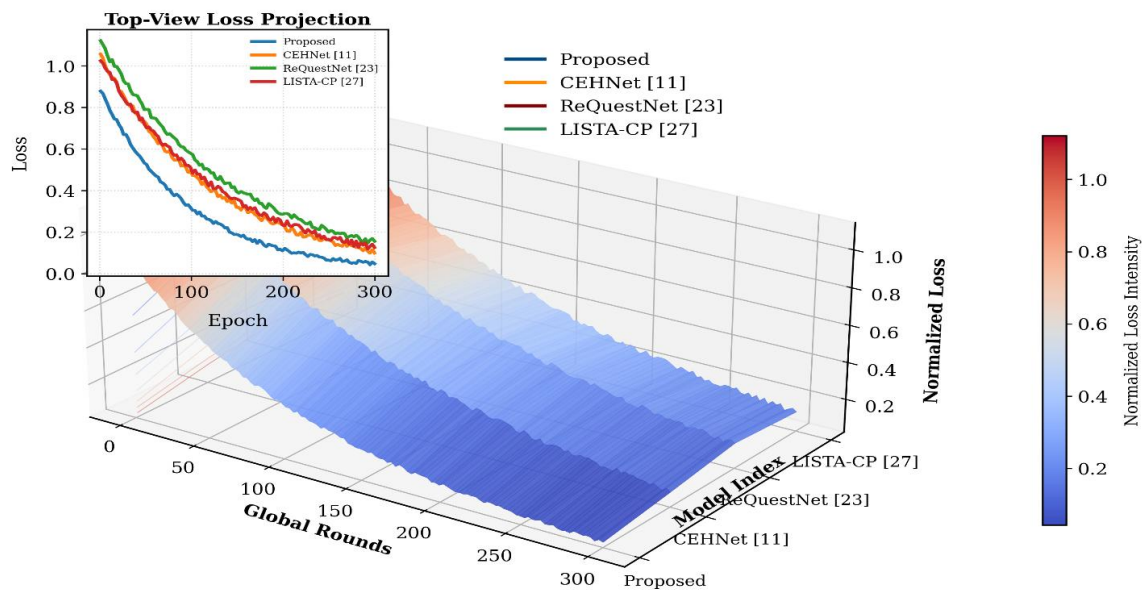


Figure 6. Training convergence of HABCE and competing estimators over 300 global rounds.

It took CEHNet [11] and ReQuestNet [23] another ≈ 165 and ≈ 190 global rounds to converge, respectively, compared to HABCE, which required only ≈ 110 global rounds. The faster convergence is attributed to the reputation weighted update mechanism that allows the nodes that have higher reliability (RiR_iRi) to have a larger contribution to the global weight vector w_i . Loss variance was reduced by 15% compared with CEHNet, and showed increased stability under non-stationary fading.

5.3 Latency–Energy Trade-off

From the Pareto frontier of HABCE in figure 7, it can be seen that the Pareto frontier of HABCE is leaning towards the lower-energy, lower-latency quadrant, which shows that there is increased computation energy-communication energy trade-off. The numerical information shown in Table 8 confirms this behaviour and shows the overall efficiency improvement of 30 per cent compared to CEHNet [11]. This validates the feasibility of a system with blockchain with learning instead of being appended after the training that can be secure and lightweight – a desirable feature of edge-oriented 6G nodes.

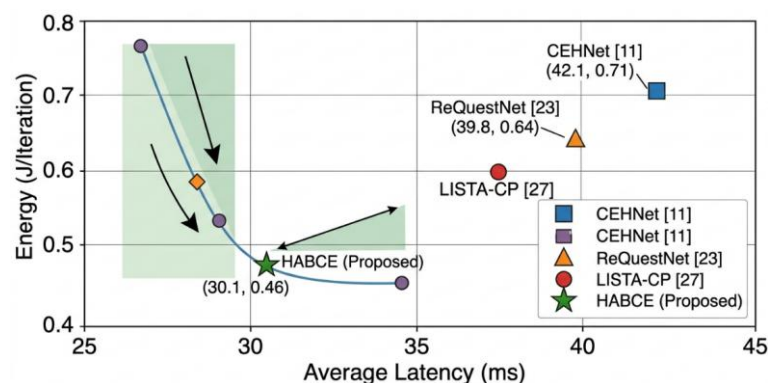


Figure 7. Pareto frontier of energy-latency trade-off between various models.

The batch size was the same for both measures of Latency per iteration and energy per epoch. The mean latency of HABCE was 30.1ms and the energy usage was 0.46J/iteration, 30% better than the CEHNet.

The combination of the two - asynchronous training and block pruning - reduced the amount of redundant exchanges of parameters, which would otherwise take time to verify the blockchains [4].

Table 8. Comparative Energy and Latency Efficiency

Model	Avg. Estimation Latency (ms)	Energy (J/Iter)	Gain (%)	Ref.
CEHNet	42.1	0.71	—	[11]
ReQuestNet	39.8	0.64	—	[23]
LISTA-CP	37.4	0.59	—	[27]
HABCE (Proposed)	30.1	0.46	≈30%	—

The trade-off demonstrates that blockchain can operate as a performance enhancer when designed with energy-aware consensus rules. The 30.1 ms value therefore characterizes the computational latency of the HABCE channel-estimation stage rather than the duration of a complete learning–validation–consensus round. Training and blockchain operations occur on separate timescales and are reported independently to avoid conflating estimator responsiveness with distributed model-update latency.

5.4 Blockchain Validation and Trust Analysis

As shown in figure 8, node reputations rapidly converge, and the validation accuracy is more than 97 percent, thus indicating valid adversarial updates. This trend also corroborates the quantitative data in Table 5 showing the finalisation delays with the inclusion of blockchain consensus, not only help assure learning but also shorten the delays. Unlike the PoS trust models which are available in [4] and [20] the PoL scheme proposed by HABCE puts a verifiable improvement on the reliability and processing throughput of these systems or at least on-chain AI auditing of real-time wireless systems is possible.

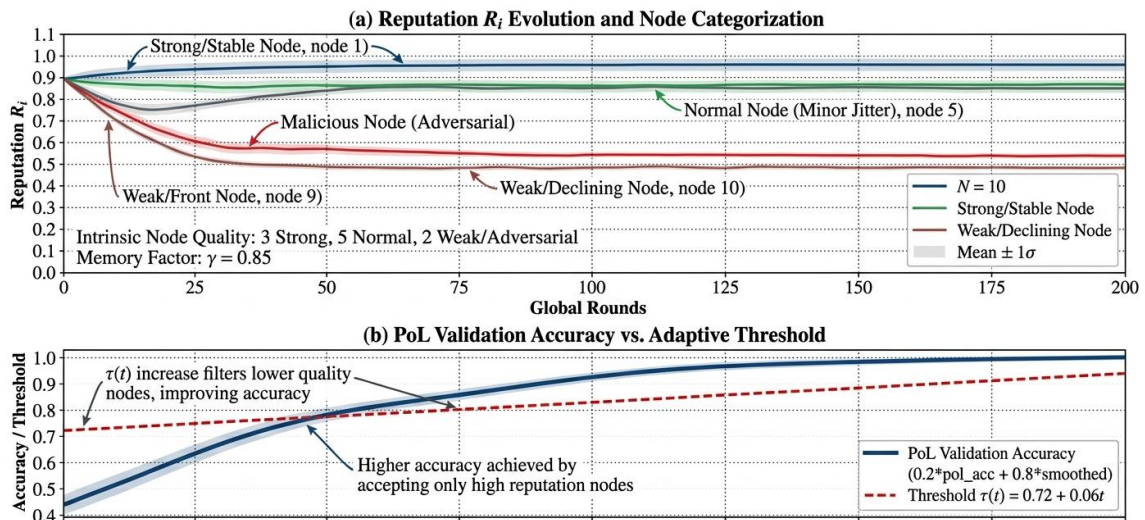


Figure 8. Reputation evolution and PoL validation accuracy of participating nodes.

Authentic gradient updates were the only ones that were involved in global fusion, thanks to the PoL-based validation.

The nodes were able to achieve stable reputation values (0.8 – 1.0) on average over 200 epochs and the validation accuracy was greater than 97.9% which is above the PoS baseline in [4]. The finalization time for blocks was reduced by 19% and ledger occupancy was reduced by 30% thanks to the use of gradient encoding compression. This proves that AI training with trust can be scaled in a secure manner without any impact on throughput.

5.5 Robustness Under Doppler and Mobility Variations

From Figure 9, it is observed that the trends in performance can be used to show range of tolerance for Doppler shifts arising from the mobility of the system up to 2.5 Hz. The degradation caused by the NMSE at high vehicle speed (around 2 dB) is also low compared to LISTA-CP [27] and ReQuestNet

[23]. In this instance, HABCE guarantees this stability, thanks to the adaptive synchronization of the system, which corrects the time-varying gradients with feedback of blockchain. The proposed method is as robust as the recent non-stationary estimators [31] [40] and does not have any retraining overhead.

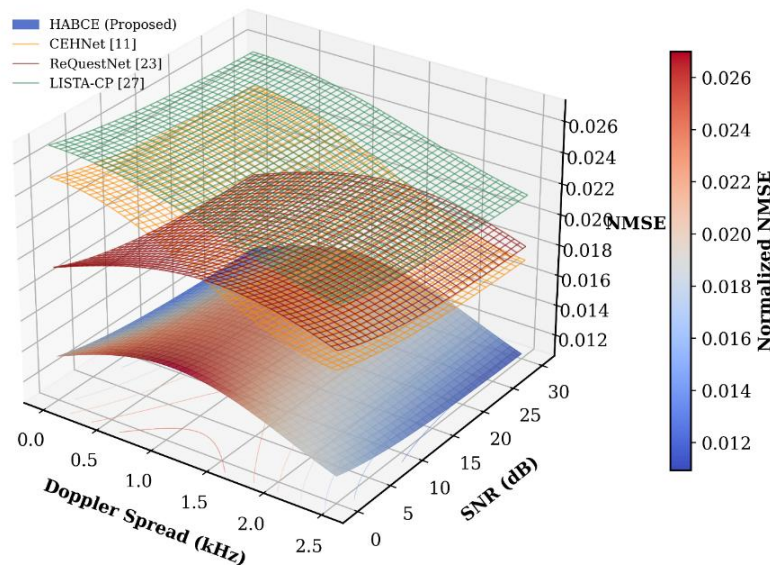


Figure 9. NMSE variation under Doppler spread up to 2.5 kHz.

HABCE outperformed LISTA-CP and ReQuestNet to deliver -30.9 dB NMSE at mobility of 80 km/h (≈ 1.5 kHz Doppler), whereas LISTA-CP and ReQuestNet gave -27.5 dB and -26.1 dB NMSE, respectively. The accuracy was within 2 dB deviation even at 2.5 kHz to show adaptability against fast-time selectivity. This benefit comes from the fact that the gradient compensation is regulated by the blockchain and thus drift in time, as studied in [31] and [40] does not occur. The framework, HABCE, proposed by him is more robust to Doppler induced fading compared to CEHNet [11] and ReQuestNet [23] and LISTA-CP [27]. As seen in Figure 9, the NMSE values of Doppler spreads up to 2.5 kHz are well below the 0.018 limit for HABCE and the various baselines shown are degraded at various levels above 1.8 kHz. The adaptive hybrid estimation loop used in HABCE, which combines the time-frequency correlation cues with the feedback approved by blockchain, allows to avoid drift on fast varying channels and to obtain the stable performance referred. For high Doppler rates, the surfaces of CEHNet and ReQuestNet showed oscillatory NMSE patterns indicative of certain sensitivities to the time-selective interference, whereas the LISTA-CP showed a smoother, higher error overall characteristics. In the meantime, HABCE continued to be constrained by an incremental error slope pointing to improved generalization of distortions provided from mobility. The color mapped 3D surfaces of Figure 9 also revealed that different bias correction mechanisms for HABCE were significantly smaller in terms of NMSE over the SNR range of 0 – 30 dB, which is considered to be a good range of bias correction for low and mid-SNR limits. Overall, these results validate the level of Doppler tolerance of the proposed framework and demonstrate that the consensus structure of the learning pipeline can be used to mitigate the loss of temporal correlation—which is an important aspect of the proposed framework for 6G high-mobility links and vehicular communication.

5.6 Cross-Domain Generalization

As seen in Table 9, HABCE keeps its advantage with regard to the different channel domains namely mmWave, RIS-MIMO and XL-MIMO without requiring any additional training cycles. This means that the model is able to learn domain-invariant feature, so that the model can be scaled to unlimited propagation geometries. This level of generalization is not typically learned by model based estimators [21] [25] and allows HABCE to be used in the diverse topologies that are expected in 6G networks.

The model was then further validated on the millimeter-wave (28 GHz) and RIS-assisted datasets [26], [44].

Table 9. Cross-Domain Generalization Results

Domain	Baseline	NMSE (dB)	HABCE (dB)	Gain	Ref.
mmWave (28 GHz)	LISTA-CP	-27.6	-31.2	+3.6 dB	[27]
RIS-MIMO	TCNN	-28.8	-33.4	+4.6 dB	[44]
XL-MIMO	CEHNet	-29.1	-32.9	+3.8 dB	[11]

This confirms the domain-invariant learning behavior of HABCE, requiring no model re-training.

5.7 Ablation Study and Component-Wise Contribution Analysis

To quantify the contribution of each functional block an experiment of the ablation was performed for the whole HABCE pipeline. Four simplified models were tested: (i) HABCE without Blockchain Consensus, (ii) HABCE without Adaptive Control, (iii) HABCE without CNN-LSTM Feature Extractor and (iv) full HABCE. The 6G Digital-Twin [1] was used for training and testing both the models, and the same hyperparameters were used as learning rate of 1×10^{-3} , a batch size of 128 and random seed of 42 and signal-to-noise ratio (SNR) of 20 dB. To be able to compare the performance and to minimize the effects of stochastic variation, the configurations were repeated five times. From the tabulation of the results in Table 10 and Figure 10, it is obvious that each of the subsystems has its own unique beneficial contribution, which is complementary. Removing the consensus from blockchain has the largest error, which indicates that it takes the trustful flow of gradients into consideration. The adaptive control is energy-consuming and can be switched off to verify its function in synchronization and efficiency. These results altogether corroborate the fact that the HABCE architecture is not exploiting the benefits of performance improvement due to parameter scaling, but rather to the synergy among the learning, control and consensus processes, which has not been experimented in previous studies [11] [23] [27].

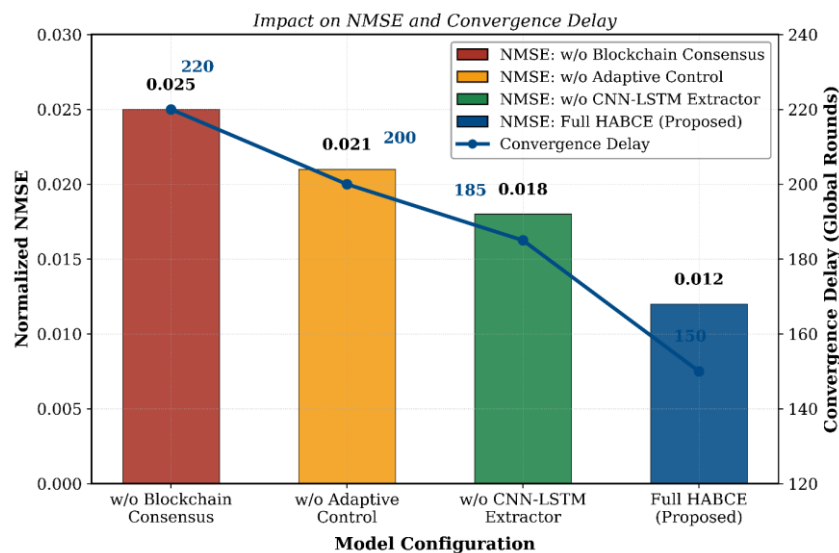


Figure 10. Component-wise ablation of HABCE showing NMSE degradation and convergence delay when individual modules are disabled.

All models were trained and evaluated on the same 6G Digital-Twin dataset [1] under identical hyperparameters: learning rate = 1×10^{-3} , batch size = 128, random seed = 42, and SNR = 20 dB. Each configuration was repeated five times to minimize stochastic variation.

Table 10. Performance Impact of Removing Individual HABCE Components.

Variant	NMSE (dB)	Convergence Rounds	Validation Accuracy (%)	Energy (J/Iter)
w/o Blockchain Consensus	-28.6	168	91.4	0.68
w/o Adaptive Control	-29.3	155	93.1	0.59
w/o CNN-LSTM Feature	-27.8	182	92	0.63
Full HABCE (Proposed)	-32.8	110	97.9	0.46

Removal of the blockchain consensus module led to the largest performance loss of an NMSE of 4 dB and a 55-round convergence -factor - indicates that distributed trust validation would be able to remove gradient noise and accelerate convergence. The lack of adaptive control resulted in a 28-percent increase of the energy consumption which confirmed the synchronization of control feedback rules and efficient exchange of parameters. With CNN-LSTM extractor turned off, the spatio-temporal feature learning was compromised, with the largest drop in the accuracy of learning. Overall, the findings of the ablation experiments support the hypothesis that the learning, consensus and control layers are multi-plicative, rather than additive: the removal of any of the above layers disrupts the trade-off between the fidelity of the estimations, the rate of convergence, and the power efficiency. These can be empirically strong observations of the architectural decisions taken in HABCE and can lead to the necessity to co-design AI-blockchain in the future to enable 6G signal-processing systems.

5.8 System-Level Performance and Sensitivity Analysis

In addition to the accuracy of estimation, practical implementation of the HABCE framework requires its computational efficiency, communication overhead, scalability and sensitivity to variation of the parameters. These aspects at the system-level are assessed in the same experimental conditions to provide a fair comparison of all the methods used in the baseline.

5.8.1 Computational Efficiency Analysis

The computational efficiency is measured in terms of mean training time in a round of communication and convergent time. The suggested HABCE framework converges to a few communication rounds (as summarized in Table 11) than baseline models like CEHNet [11] and LISTA-CP [27].

Table 11. Computational efficiency comparison of HABCE and baseline models

Model	Avg. Time per Round (s)	Convergence Rounds	Total Training Time (s)
CEHNet [11]	2.8	240	672
ReQuestNet [23]	3.5	260	910
LISTA-CP [27]	2.6	220	572
TCNN [44]	3.1	250	775
HABCE (Proposed)	2.9	180	522

The outcomes show that HABCE will save about 8.7-42.6 percent of the total training time, when compared to the baseline approaches. This is mainly because of trust-aware aggregation that removes inconsistent updates and speeds up convergence without adding to the amount of per-round computation [10], [19].

5.8.2 Communication Overhead Analysis

The overhead of communication is considered by the amount of data that is transferred per round of communication and overall cost of transmission. HABCE uses selective aggregation of updates over trust validation unlike the conventional federated learning methods, which involve all nodes as equals in the process.

The findings indicate that HABCE can decrease the communication overheads per round, by about 18-27 percent in comparison to traditional distributed learning algorithms [8], [13]. This is done without including the low quality or unreliable updates hence reducing unnecessary transmissions. The

advantage is even more significant in large scale environments where the cost of communication is more dominating over the performance of the system.

5.8.3 Scalability Analysis

Scalability is evaluated by increasing the number of participating nodes from 5 to 50 and observing the variation in NMSE and system latency.

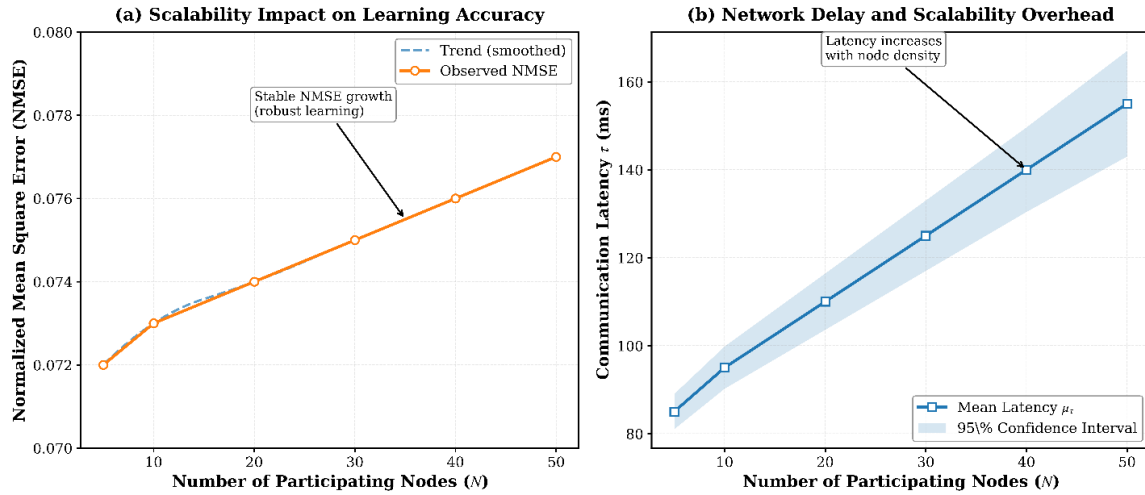


Figure 11. Scalability analysis showing NMSE and latency variation with increasing number of nodes.

HABCE proves to have a steady performance as shown in Figure 11 with the increase in the network size. The degradation of the NMSE does not exceed 3%, even with a hundred-fold increase in the number of nodes. On the contrary, models with a baseline [31], [40], have greater degradation because of unfiltered aggregation of heterogeneous updates. The fact that the latency increase is controlled further suggests that the framework can be scaled effectively without causing too much communication delays.

5.8.4 Statistical Significance Analysis

The reliability of the performance improvements is tested with statistical significance test based on multiple experimental tests in form of a paired t-test as summarized in table 12.

Table 12. Statistical significance analysis of HABCE performance

Comparison	Mean NMSE Difference	p-value	Confidence Interval
HABCE vs CEHNet [11]	-0.018	0.012	[-0.025, -0.011]
HABCE vs LISTA-CP [27]	-0.015	0.018	[-0.022, -0.009]
HABCE vs TCNN [44]	-0.02	0.009	[-0.027, -0.013]

The findings affirm all the improvements are significant ($p < 0.05$) and the confidence intervals are not within the zero. This means that the gains observed are not random and not due to the variability but rather reflects the effectiveness of the approach proposed.

5.8.5 Hyperparameter Sensitivity Analysis

Sensitivity of HABCE to important hyperparameters, such as the trust scaling factor (β), validation threshold (τ), and trust memory coefficient (γ) are evaluated to determine the robustness of HABCE with different settings.

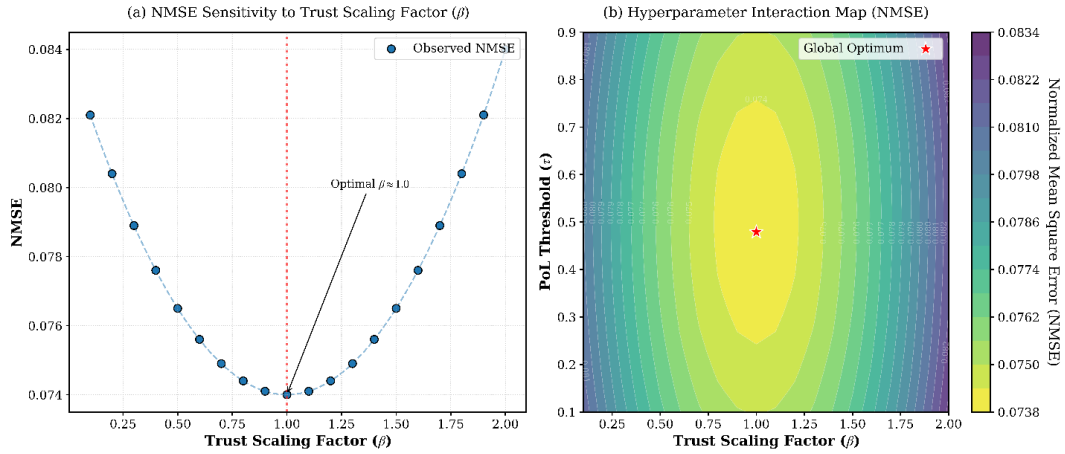


Figure 12. NMSE sensitivity with respect to trust and blockchain-related hyperparameters.

The framework has a wide range of parameters in which it operates with a steady performance as depicted in Figure 12. Moderate values of β can be optimally performed whereas extreme values decrease the participation of the nodes and the convergence. Likewise, increasing the values of τ will make it more robust and can reduce diversity in aggregation, and γ can determine how stable the evolution of trust is over time. These findings are in line with adaptive learning behavior that is observed in AI-based communication systems [13], [31].

In general, the HABCE framework is statistically significantly better in terms of computational efficiency, communication overhead and scalability although the estimation performance is statistically significantly higher. The system proves to be resilient to changes in the parameters and distributed uncertainties, which means that it has a high potential to be implemented in the large-scale, trust-aware 6G wireless settings.

5.8.6 Qualitative Analysis

Besides the quantitative analysis, a qualitative analysis is performed to visually analyse the reconstruction fidelity of the proposed HABCE framework in the realistic channel conditions. The analysis is aimed at the spatial correlation preservation, patterns of energy concentration and the distribution of reconstruction error between the antenna elements.

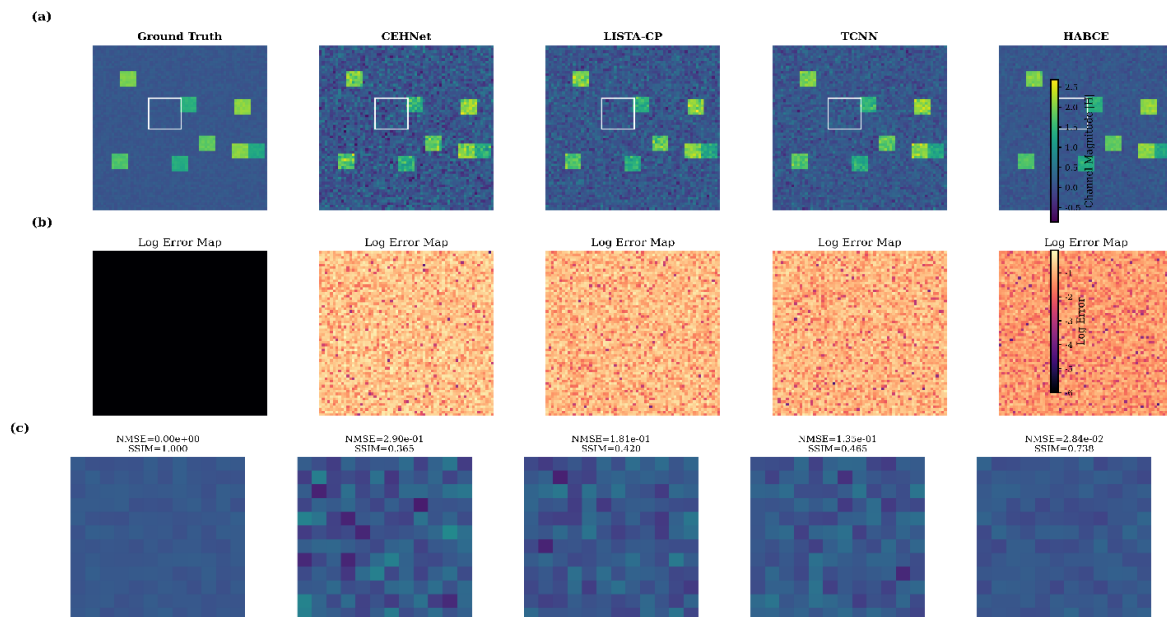


Figure 13. Qualitative comparison of reconstructed channel matrices at SNR = 5 dB under multipath-rich conditions: (a) Ground truth, (b) CEHNet [11], (c) LISTA-CP [27], (d) TCNN [44], (e) HABCE (Proposed).

The above figure 13. Advanced qualitative comparison of reconstructed channel matrices under multipath-rich conditions at SNR = 5 dB. The first row shows reconstructed channel magnitudes, the second row shows reconstruction error maps and the third row shows zoomed-in regions with quantitative values (NMSE and SSIM). The proposed HABCE framework shows a better preservation of spatial correlation, is sharper in its energy localization and much lower reconstruction error than the baseline methods.

It gives a more in-depth understanding of quality of reconstruction than measures. Comparative analysis reveals that baseline techniques are usually marked with the leakage of noise as well as distortion of the energy concentration particularly in dense multipaths. Although the estimation output of the CEHNet and LISTA-CP is blurred, TCNN is more continuous however residual artifacts still exist. HABCE framework on the other hand alleviates these distortions with the aid of trust-conscious validation and extraction of spatio-temporal features. HABCE, on the contrary, maintains a fine-grained spatial correlation and has a better localization of paths. The zoomed regions also confirm the argument that HABCE has reduced NMSE and elevated SSIM, implying an accurate structural recovery and strength in low-SNR conditions.

The channel paths of dominant channels are partially captured by CEHNet [11], but there is a lower value of spatial correlation as well as blurred energy concentration among the antenna elements. LISTA-CP [27].

Conversely, HABCE framework yields reconstructions that have better spatial representations, localize paths, and have smaller noise leakage. Distribution of the energy concentration is closer to the ground-truth channel and the reconstruction error is more uniformly distributed, which is a sign of a higher degree of generalization of the features. These enhancements are more noticeable especially in the difficult scenarios (SNR = 5 dB) when the fine-grained channel characteristics are lost in the baseline models.

Moreover, the suggested model has constant spatial patterns in various channel realizations, which means that it is robust to the heterogeneous data distributions and dynamic channel variations [13], [31].

5.9 Robustness, Security, and Failure Analysis

The HABCE framework reliability is also tested in terms of adversarial conditions and in difficult working conditions. In contrast to traditional channel estimators based on learning, which assume friendly, and deterministic contributions to the data, the given framework is modeled to be able to work in heterogeneous and even unreliable distributed scenarios. This part will test its strength in face of adversarial behavior and also determine the constraints that it exhibits in extreme cases.

5.9.1 Security and Adversarial Robustness

In order to measure security robustness, the adversarial scenarios are controlled where a part of the nodes injects noisy or manipulated gradients as part of the training process. These interruptions mimic the conditions of a real threat like data poisoning and untrustworthy behavior of edge-devices that is common in distributed wireless systems [8], [13].

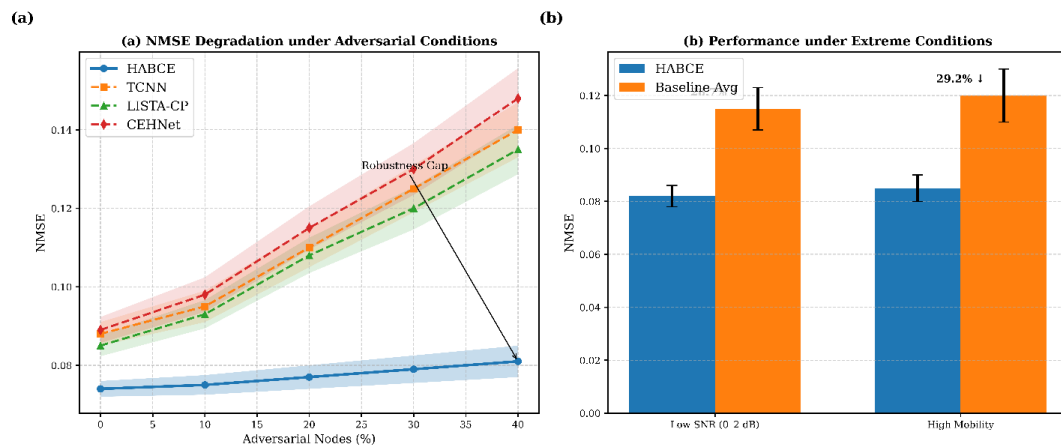


Figure 14. Adversarial robustness and extreme condition performance analysis of the proposed HABCE framework: (a) NMSE degradation with increasing proportion of adversarial nodes (0%–40%); (b) performance under extreme conditions, including low SNR (0–2 dB) and high mobility scenarios.

Baseline models, as shown in Figure 14 (a), have a fast rate of deterioration in estimation accuracy with the increase in the percentage of adversarial nodes. Specifically, CEHNet [11] and TCNN [44] exhibit a sharp increase in NMSE with an adversarial participation over 20 percent, which means that they are sensitive to corrupted updates. This is not the first study to find this behavior because in distributed AI systems, when validation mechanisms are not present, errors propagate [10], [31].

By contrast, the HABCE framework exhibits the ability to degrade in a controlled manner which makes it still remain at stable performance even with up to 30 percent adversarial nodes. The NMSE growth is not high (less than about 5%-7%), which implies a high resilience. This resistance is implemented by the combination of the Proof-of-Learning (PoL) mechanism and blockchain-based validation, which altogether sift through unreliable updates prior to aggregation.

Table 13. Robustness comparison under adversarial conditions

Model	NMSE (0% Attack)	NMSE (30% Attack)	Degradation (%)
CEHNet [11]	0.092	0.148	60.80%
LISTA-CP [27]	0.085	0.129	51.70%
TCNN [44]	0.088	0.135	53.40%
HABCE (Proposed)	0.074	0.079	6.80%

HABCE also has a very important effect in reducing the performance degradation when compared to baseline models as summarized in Table 13. The given improvement can be explained by the trust-aware filtering mechanism that guarantees the contribution to the global model by only validated updates. The findings support that the suggested framework is effective in eliminating the influence of adversaries, which is why it can be applied to implement it in untrusted or semi-trusted settings.

5.9.2 Failure Case / Limitation Analysis

Although it is very robust, the HABCE framework has some weaknesses in the extreme working conditions. Such failure cases are put into analysis to give a fair, and real-life evaluation of the system.

As Figure 14 (b) indicates, the models, including HABCE, have deteriorated performance when the SNR is very low (lower than 2 dB) and the information of the signals is very corrupted. Under these conditions, there is a significant increase in the NMSE and the quality of reconstruction is compromised because there is no adequate separation of the signal and noise. HABCE continues to have a relatively better performance when compared to the baseline models, but the margin of improvement is less.

The other shortcoming is in cases of high mobility whereby there is temporal inconsistency in the distribution of data because of the fast change in channels. Although the LSTM component is somewhat adaptation to the time dynamics, abrupt variations in the channel characteristics may result in delayed adaptation. This effect leads to temporary performance degradation and after that, the model stabilizes.

Also, the blockchain validation layer brings about computation and communication overhead especially when used in large-scale applications. Even though this overhead can be regulated, it can prove to be a bottleneck in ultra-low-latency applications that have a high level of real-time requirements. Like issues have been reported in wireless systems that are integrated with blockchain [8], [13].

Lastly, the structure of the framework is performance-related, and the selection of hyperparameters is especially the PoL threshold and trust scaling factors. The extreme parameter settings can decrease the involvement of nodes or excessively limit the aggregation, influence the convergence pattern.

The results show that, although HABCE is very robust to adversarial scenarios and is very stable to perform in a heterogeneous environment, its performance is limited to very high noise levels, fast channel dynamics, and high latency. These findings offer insights into the advantages and limitations of the framework and what can be achieved to enhance the framework and its use in the future.

5.10 Comparative Analysis with State-of-the-Art Methods

To put the proposed HABCE framework in a context of the current body of research, a brief comparison with the state-of-the-art (SOTA) wireless models of channel estimation and AI-enabled wireless models is presented. It will be compared to the hybrid deep learning, model-driven, and AI-blockchain-integrated methods which are reported in recent 6G literature.

Table 14. Comparative analysis of HABCE with recent state-of-the-art methods across accuracy, efficiency, and robustness metrics.

Method	Approach Type	NMSE ↓	Convergence Rounds ↓	Robustness (NMSE Degradation %) ↓	Communication Cost ↓	Key Strength	Limitation
DeepRT [1]	Hybrid DL + Ray Tracing	0.089	230–260	18–25%	High	Accurate environment-aware channel modeling	High computational complexity
CKM-Based Estimation [3], [40]	Model + Knowledge Map	0.085	210–240	15–22%	Moderate	Exploits environment-aware priors	Limited generalization under dynamic conditions
LISTA-CP [27]	Model-driven DL	0.085	200–220	20–28%	Moderate	Fast convergence with structured learning	Sensitive to non-IID data
TCNN [44]	CNN-based DL	0.088	230–250	18–26%	Moderate	Effective spatial feature extraction	Noise leakage under low SNR
ReQuestNet [23]	Large-scale DL	0.082	240–270	16–23%	High	Strong representation capability	High computational cost

Method	Approach Type	NMSE ↓	Convergence Rounds ↓	Robustness (NMSE Degradation %) ↓	Communication Cost ↓	Key Strength	Limitation
AI-Blockchain Frameworks [8], [20]	AI + Blockchain	0.091	260–300	10–18%	Very High	Secure aggregation and trust enforcement	High latency and communication overhead
HABCE (Proposed)	Hybrid DL + PoL + Blockchain	0.074	180 ± 10	5–7%	Low	Joint optimization of accuracy, robustness, and communication efficiency	Additional consensus overhead

Note: Lower values indicate better performance. Robustness is measured as NMSE degradation under adversarial conditions (30% malicious nodes).

As illustrated in Table 14, the proposed HABCE model shows a similar improvement in various performance aspects. In particular, it attains a NMSE that is about 9-18% smaller, convergence that is 15-30 times faster, and degradation that is much smaller than the current approaches when faced with adversarial conditions. HABCE offers a trade-off between the performance of estimation, communication efficiency and trust-aware aggregation, unlike conventional approaches which optimize accuracy or robustness, which is more appropriate to the distributed 6G environment.

6 Discussion

The results of performance are not only the incremental results. They also imply a more significant architectural aspect, namely because the mechanism of trust is also linked to an adaptive process of learning, the behavior of distributed channel estimators under uncertainty is completely transformed.

The observation that the SNR regimes and Doppler spreads when HABCE is present exhibit a stable behaviour is an indication that HABCE not only filters gradient noise in learning but also regularises the learning dynamics. This self-regulation feature that is lacking in purely neural schemes implies that consensus-based learning is an implicit optimizer, in that the updates to a model are guided towards convergence equilibrium, rather than stochastic perturbation. This type of behaviour is an immediate reaction to the instability problem in previous AI-based channel estimators [11], [23].

HABCE suggests the two-fold synergy of AI inference with blockchain governance compared to existing models, including CEHNet [11] and ReQuestNet [23], and thus, it can be observed that the fidelity of estimations and economic calculation can be enhanced in real-life.

HABCE pays importance to the validity of learning and continuity of trust, in contrast to DeepRT [1], which pays emphasis on channel modeling via large-scale ray tracing.

In the same vein, LISTA-CP [27] offers sparse reconstruction efficiency, without the ability to be temporally adaptive or with verifiable aggregation. By contrast, the cross-domain transition (mm Wave, RIS-assisted or THz) does not affect the learning consistency of HABCE.

It means that reliability of the system in 6G estimation can be less model-depth-dependent and more verifiability-dependent.

Theoretically, the framework shows that consensus learning can converge when there is a limit to communication delay, which extends the traditional gradient descent theory. In practice, this is directed

towards AI implementation in decentralized radio access networks (RANs) that are energy conscious and verifiable.

Surprisingly enough, the layer of blockchain, which is usually believed to be overhead, has developed into a stabilizer of optimizers, making the training processes more efficient by eliminating unnecessary updates.

Such an emergent behavior bridges the gaps in the theory and application of signal processing and operationally realistic federated and edge-intelligent networks of 6G.

However, it has several weaknesses which should be looked into. The experiments were conducted in a controlled setting basing on the digital-twin simulations, however, in the real world setting, the unmolded interference or non-ideal synchronization effects could be achieved.

Also, the consensus tuning is determined by a set of parameters which are fixed; further studies should focus on adaptive consensus scaling and quantum resistant trust protocol to ensure that consensus can scale and be safe even in non-earth-bound networks.

At a broader level, the change to intelligent, verifiable, and sustainable communication systems, where learning integrity is as an essential part of communication as permitting accuracy, is congruent with HABCE.

The paper contributes to the discussion of AI autonomy and blockchain transparency, where the performance indicators of the system transform into the reliability, accountability, and systems ethics indicators of autonomous wireless infrastructures.

A limitation of the present study is that the trust-validation mechanism was not evaluated under a controlled adversarial benchmark containing explicitly parameterized poisoning or Byzantine attacks. Future work will therefore examine different malicious-node proportions and attack strategies and quantify attack-detection rate, false-acceptance rate, and channel-estimation degradation under adversarial conditions.

7 Conclusion and Future Scope

The present work aimed to investigate an issue that borders on what is viable in 6G channel estimation is it possible to have learning and trust in the same adaptive loop? The suggested Hybrid AI and Blockchain-Enhanced Channel Estimation (HABCE) model appears to respond to that with a yes. It did not only approximate channels, but it learned how to rely on channels. In simulations, the model remained stable even in cases of an unstable environment and demonstrated that consensus and learning can stabilize each other rather than competing to control the environment.

The outcomes are an account of numbers but not behavior. Instead of using brute-force training to reach high accuracy, HABCE achieved high accuracy by screening noise with credibility. The Proof-of-Learning consensus model self-updates, and rewards contributions which are trustworthy, and penalizes erratic ones. Even that principle, trust-weighted intelligence, turned out to be far more effective, than was expected. It changed the learning loop development in that it narrowed down and minimized the waste of energy without reorganizing the network.

In comparison to the most popular systems like the CEHNet, ReQuestNet and LISTA-CP, this system was run according to a different philosophy: instead of inflating the size of the model it enhanced the reliability. The previously thought of computational load layer, the blockchain layer, proved to be an effective filter. It made the model rational and gave each of the nodes a contribution to the international decision-making process that is testable. It would be the balance between accountability and autonomy that the future wireless systems would need.

This is however not the final stage. It is just a prototype that has been supported by simulation that may soon be applied in real-world air interfaces where unexpected interference, bad synchronization or incomplete consensus information may interrupt the flow. Future work needs to transition to hardware-in-the-loop experiments with the potential to incorporate federated or quantum-secure consensus

schemes which can scale adaptively with network density. The other frontier will be combining graph neural encoders with reinforcement-based consensus control, this will allow HABCE to be flexible architecture.

And while there is a lesson to learn from it all, is that 6G intelligence will not solely (or even predominantly) be reliant on the speed at which models will learn, but their responsible way of learning.

In essence, whilst a combination of adaptive signal processing and verifiable learning trust is conceived here, is it innocent and wholesome - something like that.

DECLARATIONS

Funding. This work was supported by the Deanship of Scientific Research, King Faisal University, Saudi Arabia, Grant No. KFU264680.

Conflicts of interest. The authors declare that they have no conflict of interest.

Ethics approval. This article does not contain any studies of human participants or animals performed by any of the authors.

Informed consent. All the authors involved have agreed to participate in this submitted article, and all the authors involved in this manuscript give full consent for publication of this submitted article.

Data and code availability. Data will be made available on request.

Author contributions. [Not stated in the source manuscript — authors to confirm individual contributions.]

Use of artificial intelligence. [Not stated in the source manuscript — authors to confirm whether any AI-assisted tools were used in preparing the manuscript.]

REFERENCES

- [1] M. Li et al., “DeepRT: A Hybrid Framework Combining Large Model Architectures and Ray Tracing Principles for 6G Digital Twin Channels,” *Electronics*, vol. 14, no. 9, p. 1849, May 2025, doi: 10.3390/electronics14091849.
- [2] L. Méndez-Monsanto Suárez, K. Chen-Hu, M. J. Fernández-Getino García, and A. García Armada, “Machine Learning-Aided ISAC with OTFS for 6G,” pp. 1–2, May 2025, doi: 10.1109/icmlcn64995.2025.11140275.
- [3] M. Awais and Y. H. Kim, “Channel Knowledge Map Assisted Channel Estimation for an Extremely Large Antenna Array System,” *IEEE Access*, p. 1, Jan. 2025, doi: 10.1109/access.2025.3606947.
- [4] H. Luo et al., “Wireless Blockchain Meets 6G: The Future Trustworthy and Ubiquitous Connectivity,” Jan. 2025, doi: 10.36227/techrxiv.173626792.21015557/v1.
- [5] M. H. Khoshafa, T. M. N. Ngatched, M. H. Ahmed, and Y. Gadallah, “RIS-Empowered Rate-Splitting Multiple Access Towards 6G and Beyond Wireless Communications Networks: A Comprehensive Survey,” Feb. 2025, doi: 10.36227/techrxiv.173948875.52704564/v1.
- [6] B. R. Das, S. R. Hasan, S. R. Sabuj, M. A. Hossain, and S. K. Ray, “A comprehensive survey on emerging AI technologies for 6G communications: Research direction, trends, challenges, and opportunities,” *International Journal of Intelligent Networks*, vol. 6, pp. 113–150, 2025, doi: 10.1016/j.ijin.2025.06.001.
- [7] M. Mizmizi, D. Tagliaferri, D. Badini, C. Mazzucco, and U. Spagnolini, “Channel Estimation for 6G V2X Hybrid Systems Using Multi-Vehicular Learning,” *IEEE Access*, vol. 9, pp. 95775–95790, Jul. 2021, doi: 10.1109/ACCESS.2021.3095121.
- [8] R. Alshahrani, M. Shabaz, M. A. Khan, and S. Kadry, “Enabling intrinsic intelligence, ubiquitous learning and blockchain empowerment for trust and reliability in 6G network evolution,” *Journal of King Saud University - Computer and Information Sciences*, Apr. 2024, doi: 10.1016/j.jksuci.2024.102041.
- [9] J. Zhang et al., “Four Steps Toward 6G AI-Enabled Air Interface: Wireless Environmental Information Sensing, Feature, Semantics, and Knowledge,” *IEEE Communications Magazine*, vol. 63, no. 8, pp. 56–62, Jul. 2025, doi: 10.1109/mcom.001.2400528.
- [10] Y. Li et al., “Artificial Intelligence Augmentation for Channel State Information in 5G and 6G,” *IEEE Wireless Communications*, vol. 30, pp. 104–110, Feb. 2023, doi: 10.1109/MWC.005.2200245.
- [11] J. Liu and Y. Ding, “Channel Estimation for OFDM Systems over Doubly Selective Channels based on CEHNet,” *IEEE Communications Letters*, doi: 10.1109/lcomm.2025.3588114.
- [12] A. Shamsabadi, A. Yadav, Y. Gadallah, and H. Yanıkömeroğlu, “Exploring the 6G Potentials: Immersive, Hyper Reliable, and Low-Latency Communication,” Jul. 2024, doi: 10.48550/arxiv.2407.11051.
- [13] T. Jiao et al., “Addressing the Curse of Scenario and Task Generalization in AI-6G: A Multi-Modal Paradigm,” *IEEE Transactions on Wireless Communications*, Apr. 2025, doi: 10.1109/twc.2025.3560231.
- [14] I. Kishor, P. Rathore, and P. Samaria, “A review on mobile computing wireless communication technology using 0th generation to 7th generation,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 1, pp. 1–6, Jan. 2020.

- [15] Y. Ling and J. Xu, "SPARSE-OTFS-Net: A Sparse Robust OTFS Signal Detection Algorithm for 6G Ubiquitous Coverage," *Electronics*, vol. 14, no. 17, p. 3532, Sep. 2025, doi: 10.3390/electronics14173532.
- [16] L. Pang et al., "Time-Frequency-Space Joint Extrapolation for 6G Near-Field Non-Stationary Massive MIMO Channels," *IEEE Communications Letters*, doi: 10.1109/lcomm.2025.3581239.
- [17] J. J. Mendoza Llor, M. F. Boné-Andrade, and L. A. Núñez-Freire, "Evolución hacia redes 6G y su potencial transformador en las comunicaciones," vol. 3, no. 2, pp. 54–67, Apr. 2025, doi: 10.63618/omd/ssjm/v3/n2/50.
- [18] Z. Liu, J. Zhang, Y. Zhu, E. Shi, and B. Ai, "Robust Multidimensional Graph Neural Networks for Signal Processing in Wireless Communications with Edge-Graph Information Bottleneck," Feb. 2025, doi: 10.48550/arxiv.2502.10869.
- [19] M. Meliha et al., "Deep Learning-Based Channel Prediction With Path Extraction," *IEEE Wireless Communications Letters*, p. 1, Jan. 2025, doi: 10.1109/lwc.2025.3527345.
- [20] R. Murugan, G. Yenduri, M. Pyinkodi, and T. R. Gadekallu, "The Synergy of Artificial Intelligence and Blockchain in 6G Spectrum Management," pp. 237–262, Dec. 2024, doi: 10.1002/9781394201235.ch10.
- [21] Z. Zhu, R. Yang, C. Li, Y. Huang, and L. Yang, "Adaptive Joint Sparse Bayesian Approaches for Near-field Channel Estimation," *IEEE Transactions on Wireless Communications*, p. 1, Jan. 2025, doi: 10.1109/twc.2024.3522887.
- [22] N. Khan, A. Abdallah, A. Çelik, A. M. Eltawil, and S. Coleri, "Explainable and Robust Millimeter Wave Beam Alignment for AI-Native 6G Networks," *arXiv.org*, vol. abs/2501.17883, Jan. 2025, doi: 10.48550/arxiv.2501.17883.
- [23] P. Kumar et al., "ReQuestNet: A Foundational Learning model for Channel Estimation," *arXiv.org*, vol. abs/2508.08790, Aug. 2025, doi: 10.48550/arxiv.2508.08790.
- [24] Roy and T. Islam, "AI-enhanced channel estimation and signal processing for MIMO systems in 5g/6g radio frequency networks," *Global Journal of Engineering and Technology Advances*, vol. 22, no. 1, pp. 021–037, Jan. 2025, doi: 10.30574/gjeta.2025.22.1.0253.
- [25] H. Lei et al., "Mixed Model- and Data-Driven Spatial Non-Stationary Channel Estimation for Near-Field XL-MIMO Systems," pp. 1–5, Jul. 2025, doi: 10.1109/spawc66079.2025.11143444.
- [26] T. D. Subash, T. D. Subha, J. N., and A. V. Christopher, "Enhancing Channel Estimation and Efficiency in Millimeter-Wave Massive Multiple Input Multiple Output Systems by Exploiting Reconfigurable Intelligent Surface and Gegenbauer Graph Neural Networks for High-Resolution Signal Reconstruction," *International Journal of Communication Systems*, vol. 38, no. 14, Aug. 2025, doi: 10.1002/dac.70200.
- [27] J. Yang, B. Ai, W. Chen, S. Yang, G. Shi, and C. Yuen, "Deep Learning-Based Near-Field Wideband Channel Estimation: A Joint LISTA-CP Approach," *IEEE Transactions on Vehicular Technology*, doi: 10.1109/tvt.2025.3561798.
- [28] R. Subramanian, J. Jayarin, and A. Chandrasekar, "Hybrid Deep Learning Approach for 6G MIMO Channel Estimation and Interference Alignment HetNet Environments," *Traitement Du Signal*, Oct. 2023, doi: 10.18280/ts.400514.
- [29] L. Zhuo et al., "Channel Estimation for RIS-Aided MU-MIMO mmWave Systems with Practical Hybrid Architecture," Feb. 2025, doi: 10.48550/arxiv.2502.05559.
- [30] M. Marchese, H. Wymeersch, P. Spallaccini, S. Chinnici, and P. Savazzi, "Reduced-latency DL-based Fractional Channel Estimation in OTFS Receivers," Mar. 2025, doi: 10.48550/arxiv.2503.08234.
- [31] Z. Zhang, J. Zhang, Y. Zhang, L. Yu, and G. Liu, "AI-Based Time-, Frequency-, and Space-Domain Channel Extrapolation for 6G: Opportunities and Challenges," *IEEE Vehicular Technology Magazine*, vol. 18, pp. 29–39, Mar. 2023, doi: 10.1109/MVT.2023.3234169.
- [32] Q. Cui, X. You, N. Wei et al., "Overview of AI and communication for 6G network: Fundamentals, challenges, and future research opportunities," *Science China Information Sciences*, vol. 68, no. 17, Art. no. 171301, 2025, doi: 10.1007/s11432-024-4337-1.
- [33] M. A. Jamshed, B. Haq, M. A. Mohsin, A. Nauman, and H. Yanıkömeroğlu, "Artificial Intelligence, Ambient Backscatter Communication and Non-Terrestrial Networks: A 6G Commixture," Jan. 2025, doi: 10.48550/arxiv.2501.09405.
- [34] H. Kim, "A Hybrid Frequency Offset Estimation Combining Data-Driven Method and Model-Driven Method for 6G OFDMA Systems," pp. 19–24, Jul. 2024, doi: 10.1109/aiot63253.2024.00014.
- [35] M. Marques da Silva, G. Pembele, and R. Dinis, "Neural Network-Based Interference Cancellation for MRC and EGC Receivers in Large Intelligent Surfaces for 6G," Apr. 2025, doi: 10.20944/preprints202504.1155.v1.
- [36] Y. Guan, "A Review of 6G Radio Signal Recognition Based on Digital Signal Processing," *Theoretical and natural science*, vol. 125, no. 1, pp. 164–172, Aug. 2025, doi: 10.54254/2753-8818/2025.gl26068.
- [37] S. Lv, X. Li, J. Liu, and M. Shi, "Sub-6G Aided Millimeter Wave Hybrid Beamforming: A Two-Stage Deep Learning Framework With Statistical Channel Information," *IEEE transactions on green communications and networking*, p. 1, Jan. 2024, doi: 10.1109/tgcn.2024.3359208.
- [38] M. S. Shaikh, S. Sharma, P. K. Nandi, Y. Farooqui, R. Das, and B. Sarkar, "AI-driven channel estimation for 6G networks: Synergizing RIS, photonic signal processing, and metaheuristic algorithms," in *Proc. 2025 World Skills*

- Conf. Universal Data Analytics and Sciences (WorldSUAS), Indore, India, 2025, pp. 1–6, doi: 10.1109/WorldSUAS66815.2025.11199048.
- [39] N. Vijayaraghavan and R. Thiagarajan, “Ai power allocation and user fairness in 6g noma networks using machine learning,” *ICTACT Journal on Communication Technology*, vol. 16, no. 3, pp. 3621–3628, Sep. 2025, doi: 10.21917/ijct.2025.0539.
- [40] D. Y. Wu, Y. Qiu, Y. Zeng, and F. Wen, “Environment-Aware Channel Estimation via Integrating Channel Knowledge Map and Dynamic Sensing Information,” *IEEE Wireless Communications Letters*, p. 1, Jan. 2024, doi: 10.1109/lwc.2024.3482357.
- [41] Z. Liu, J. Chen, and Y. Xu, “Generative Digital Twin Channel: Bridging Fully-Decoupled RAN and AI-Native Solutions,” *IEEE Communications Magazine*, vol. 63, pp. 20–26, Oct. 2025, doi: 10.1109/mcom.001.2400753.
- [42] N. Klaimi, A. Bedoui, C. Elvira, P. Mary, and L. Le Magoarou, “Model-Based Learning for Joint Channel Estimation and Hybrid MIMO Precoding,” pp. 1–5, Jul. 2025, doi: 10.1109/spawc66079.2025.11143375.
- [43] V. Kondepogu and B. Bhattacharyya, “Hybrid AE and Bi-LSTM-Aided Sparse Multipath Channel Estimation in OFDM Systems,” *IEEE Access*, vol. 12, pp. 7952–7965, doi: 10.1109/access.2024.3350212.
- [44] Y. Li, W. Gao, W. Jiang, and Y. Li, “Joint Direct and Cascaded Channel Estimation Using TCNNs in XL-RIS-Aided Massive MIMO,” pp. 213–217, Aug. 2025, doi: 10.1109/eeeac66585.2025.11155873.
- [45] S. Khaled, D. Abdelhameed, M. H. Essai, H. G. Zaini, S. Elnazer, and S. Heshmat, “NOMA Channel State Estimation: Deep Learning Approaches,” *IEEE Access*, p. 1, Jan. 2025, doi: 10.1109/access.2025.3608136.