Privacy Preserving Federated Fine Tuning of Large Language Models for Autonomous Aerial Vehicles Command Generation

Privacy Preserving Federated Fine Tuning of Large Language Models for Autonomous Aerial Vehicles Command Generation*

Shafiq Ahmed[a], Mohammad Hossein Anisi[b], Ayesha Iqbal[c], Zakawat Liaqat[d], Mohammed Amoon[e] and Saru Kumari[f]

[a]School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom
[b]School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom
[c]School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom
[d]Webster University Leiden Campus, Eschertoren 10F, Leiden, 2316 ET, Netherlands
[e]Department of Computer Science and Engineering, Applied Studies College, King Saud University, P.O. Box 28095, Riyadh, 11437, Saudi Arabia
[f]Department of Mathematics, Chaudhary Charan Singh University, , Meerut, 250004, India

ARTICLE INFO

Keywords:
Federated Learning
Large Language Models
Differential Privacy
Secure Aggregation
Byzantine Resilience
Low Rank Adaptation

ABSTRACT

Fine tuning large language models for Autonomous Aerial Vehicles command generation exposes flight logs, patrol routes, and Global Positioning System (GPS) coordinates during collaborative training. This paper presents a privacy preserving federated fine tuning framework that maps operator instructions to structured JavaScript Object Notation (JSON) drone commands while defending against privacy leakage, server visibility, and poisoned clients. The framework builds on Low Rank Adaptation (LoRA) based Federated Averaging (FedAvg). Differential privacy (DP) is enforced through per step Differentially Private Stochastic Gradient Descent (DP-SGD) with Rényi Differential Privacy (RDP) accounting, which bounds per client leakage. Elliptic Curve Cryptography (ECC) based pairwise secure aggregation over the National Institute of Standards and Technology (NIST) P-256 curve with hash based key derivation hides individual updates. Krum and coordinate wise trimmed mean filter Byzantine updates. We evaluate 14,446 command pairs across six mission categories under non independent and identically distributed (non IID) Dirichlet partitions. With five simulated clients, TinyLlama-1.1B, $\varepsilon = 8$, and Krum, the full framework reaches a Bilingual Evaluation Understudy (BLEU) score of 0.8608 and 100% JSON validity under one sign flip attacker, compared with 0.8618 BLEU for the centralized baseline. LoRA subsampling makes per step DP noise negligible ($\sim 10^{-6}$ per parameter), so data heterogeneity dominates utility. Without Byzantine resilient aggregation, FedAvg falls to 76.6% JSON validity.

1. Introduction

Autonomous aerial vehicles have shifted from specialized military tools to widely deployed platforms that serve agriculture, infrastructure inspection, disaster response, and logistics operations around the world. The global market for these vehicles stood at roughly USD 26 billion in 2025, and current projections suggest it will nearly double before the end of the decade [1]. As fleets expand and missions grow in complexity, operators have started relying on natural language interfaces to issue commands instead of manually programming waypoints. Large language models are what make this practical. Recent studies have demonstrated that an LLM can take a spoken instruction like "fly to the bridge and take thermal photos" and convert it into a structured JSON command that a flight controller executes directly [2, 3, 4]. The appeal is hard to deny. The risks, on the other hand, tend to go unnoticed.

Fine tuning these models demands training data drawn from actual operations, and that data is inherently sensitive. Flight logs, GPS coordinates, patrol routes, and surveillance targets all contain operational or personal information that autonomous aerial vehicle operators would be reluctant to share openly. Federated learning presents a natural alternative by keeping raw data on each drone's edge device and transmitting only model updates to a central server [5]. Sharing model updates, however, is not the same as sharing nothing. Gradient inversion attacks [6] have been shown to reconstruct fragments of training data from the updates a client sends, and more recent findings indicate that even text sequences can be recovered from language model gradients with surprising accuracy [7]. A curious aggregation server, or a passive eavesdropper monitoring air to ground transmissions, could exploit this vulnerability to extract mission critical information.

Defending against these threats with a single mechanism is not enough. Differential privacy [8] places a bound on what any observer can infer from a model update, but it still leaves each client's noised update visible to the server individually. Secure aggregation [9] conceals individual contributions behind pairwise cryptographic masks, yet it offers no protection when a drone in the fleet has been compromised and begins injecting poisoned updates. Byzantine resilient aggregation rules such as Krum [10] and coordinate wise

*The authors extend their appreciation to Ongoing Research Funding Program (ORF-2026-968), King Saud University Riyadh, Saudi Arabia.

s.ahmed@essex.ac.uk (S. Ahmed); m.anisi@essex.ac.uk (M.H. Anisi); ayeshaiqbal0197@gmail.com (A. Iqbal); zakawatliaqat@gmail.com (Z. Liaqat); mamoon@ksu.edu.sa (M. Amoon); saryusiirohi@gmail.com (S. Kumari)

ORCID(s): 0000-0001-9599-1887 (S. Ahmed); 0000-0001-8414-2708 (M.H. Anisi); 0009-0005-8061-7821 (Z. Liaqat); 0000-0003-4929-5383 (S. Kumari)

trimmed mean [11] can identify and discard malicious contributions, though they were originally designed for environments without privacy noise and may not interact well with DP perturbations. Each of these defenses targets a different adversary, and getting them to work together reliably is more difficult than it might seem at first glance.

Several recent efforts have tackled subsets of this problem. Sun et al. [12] combined federated LoRA with differential privacy but did not address Byzantine resilience or secure aggregation. Gao et al. [13] handled Byzantine resilience and DP together with verifiable aggregation, but targeted convolutional networks on image benchmarks rather than language models. Xia et al. [14] addressed DP, Byzantine tolerance, and communication efficiency jointly, again on vision tasks. In the Autonomous Aerial Vehicles domain, Ayana et al. [15] introduced privacy preserving LLM inference for drone swarms via secure multiparty computation, but left the fine tuning phase, where raw operational data is most directly exposed, entirely unprotected. To the best of our knowledge, no existing work unifies federated LLM fine tuning with differential privacy, cryptographic secure aggregation, and Byzantine resilient aggregation for Autonomous Aerial Vehicles command generation under non-IID data conditions.

This manuscript presents a privacy preserving federated fine tuning framework for LLM driven Autonomous Aerial Vehicles command generation. We fine tune TinyLlama-1.1B [16] using LoRA [17] adapters across a simulated fleet of five Autonomous Aerial Vehicles clients, where data heterogeneity is modeled via Dirichlet distributed class imbalance. The framework layers four defense mechanisms that operate simultaneously. Our main contributions are as follows.

1. We propose the first framework, to our knowledge, that jointly integrates LoRA based federated fine tuning, per step differentially private stochastic gradient descent with Rényi DP accounting, ECC based (NIST P-256) pairwise secure aggregation with forward secrecy, and Byzantine resilient aggregation (Krum and trimmed mean) for LLM driven Autonomous Aerial Vehicles command generation.
2. We construct a domain specific dataset of 14,446 natural language to structured JSON command pairs spanning six Autonomous Aerial Vehicles mission categories and evaluate under realistic non-IID conditions using Dirichlet concentration parameters $\alpha \in \{0.1, 0.5, 1.0, 10.0\}$.
3. We demonstrate, through a 500 sample evaluation, that the full framework with DP ($\varepsilon = 8$) and Krum aggregation achieves a BLEU score of 0.8608 and 100% JSON validity under a single Byzantine attacker performing sign flip poisoning, compared to 0.8618 BLEU for the centralized non private baseline. We also show that LoRA's low rank parameter structure combined with subsampling amplification renders per step DP noise negligible in this regime, with per parameter noise on the order of 10^{-6}, a finding that is itself of practical interest.
4. We empirically characterize the interaction between data heterogeneity and differential privacy under federated LoRA fine tuning, finding that Dirichlet concentration α is the dominant factor affecting model quality rather than the privacy budget ε.

The remainder of this paper is organized as follows. Section 2 surveys the related literature across federated LLM fine tuning, differential privacy, secure aggregation, Byzantine resilience, and LLM driven Autonomous Aerial Vehicles systems. Section 3 formalizes the system architecture and threat model. Section 4 describes the proposed framework in detail. Section 5 presents the experimental setup and results. Section 6 discusses the findings, limitations, and practical implications. Section 7 concludes the paper with directions for future work.

2. Related Work

We are addressing five research problems in this section that lead to one problem. No single research problem individually addresses the issue we are addressing in this manuscript.

2.1. Federated Fine Tuning of Large Language Models

The original FedAvg protocol [5] was built for relatively small models. Applying it to a billion parameter language model means transmitting several gigabytes per client per round, which is flatly impractical over an autonomous aerial vehicle radio link. LoRA [17] changed the picture. By freezing the pre trained weights and training only small injected matrices, it cuts the update size by roughly two orders of magnitude. Naturally, people started combining the two. FedIT [26] showed that a straightforward LoRA plus FedAvg pairing already performs well on instruction tuning benchmarks. FlexLoRA [18] at NeurIPS 2024 went further, letting each client pick its own LoRA rank depending on hardware and reconciling the mismatched ranks through SVD during aggregation. FFA-LoRA [12] took a simpler route, freezing one of the two LoRA matrices entirely so that non IID inconsistencies during aggregation are halved. A survey by Tian et al. [27] catalogued these and other FedLoRA variants into three broad families.

All of this is encouraging work, but it shares a blind spot. Every FedLoRA paper we could find evaluates on text classification or open ended generation. None has tested whether federated LoRA holds up when the model must produce valid, machine executable JSON. A hallucinated field or a misplaced bracket in an autonomous aerial vehicle command is not just an accuracy drop. It is a safety hazard.

2.2. Differential Privacy in Federated Learning

DP-SGD [8] gave us a clean mechanism for bounding information leakage during training. Clip gradients, add noise, proceed. Mironov [28] later made this practical over

Table 1
Comparison of the proposed framework with existing related work. ✓ indicates the feature is addressed and × indicates it is not.

Work	Year	Federated	LLM/LoRA	Diff. Privacy	Secure Agg.	Byzantine	AAV Domain
[4]	2024	×	✓	×	×	×	✓
[5]	2017	✓	×	×	×	×	×
[9]	2017	✓	×	×	✓	×	×
[10]	2017	✓	×	×	×	✓	×
[12]	2024	✓	✓	✓	×	×	×
[13]	2024	✓	×	✓	✓	✓	×
[14]	2025	✓	×	✓	×	✓	×
[15]	2025	×	✓	×	✓(MPC)	×	✓
[18]	2024	✓	✓	×	×	×	×
[19]	2023	✓	×	✓	×	✓	×
[20]	2024	✓	×	×	✓	×	×
[21]	2023	✓	×	×	✓	✓	×
[22]	2025	✓	×	×	✓	✓	×
[23]	2026	✓	×	✓	×	×	✓
[24]	2023	✓	×	✓	✓	×	×
[25]	2024	✓	×	✓	✓	×	×
Proposed Framework	**2026**	✓	✓	✓	✓	✓	✓

many iterations by introducing Rényi DP (RDP), which yields far tighter composition bounds than the original moments accountant. In federated settings, McMahan et al. [29] showed that client level DP avoids trusting the server but costs more in utility, and Kairouz et al. [30] identified the interplay between data heterogeneity and DP as a critical open question. That happened in 2021 and it remains just as significant today.

More recently researchers have started asking how DP behaves inside LoRA specifically. FFA-LoRA [12] demonstrated that perturbing both low rank matrices A and B amplifies noise multiplicatively, and that freezing one eliminates this effect. PrivateLoRA [31] restructured the decomposition to reduce sensitivity, while Zhao et al. [32] proposed momentum based gradient filtering to shrink norms before clipping. These are useful techniques. But the particular kind of heterogeneity found in Autonomous Aerial Vehicles fleets, where one drone mostly handles crop surveillance and another mostly responds to emergencies, creates gradient distributions so divergent that aggressive clipping may destroy the signal entirely. We are not aware of prior work that examines this interaction empirically for LLM based command generation, and our results in Section 5 suggest the effects are significant.

2.3. Secure Aggregation Protocols

Differential privacy limits what an adversary can infer, but the aggregation server still sees each client's noised update clearly. SecAgg [9] fixed this elegantly. Every client pair generates a shared mask via Diffie Hellman key exchange, the masks cancel upon summation, and the server learns only the aggregate. It tolerates up to a third of clients dropping out mid round, which matters when Autonomous Aerial Vehicles links are unreliable. The downside is $O(n^2)$ communication. Bell et al. [33] brought this down to poly logarithmic levels, and Flamingo [34] amortized setup costs across rounds. Gao et al. [13] combined verifiable aggregation with Byzantine resilience under DP, handling the case where both server and clients may be compromised. Our protocol uses the pairwise masking idea from Bonawitz et al. but over ECC (NIST P-256) for shorter keys (256 bits versus 2048 bits at equivalent security), and derives round specific masks via HKDF to achieve forward secrecy.

2.4. Byzantine-Resilient Aggregation

In an autonomous aerial vehicle swarm, a compromised drone could submit poisoned model updates. Blanchard et al. [10] addressed this by selecting the update closest to the population majority, tolerating up to $f < (n-2)/2$ Byzantine participants, but it discards all other honest contributions per round. Coordinate wise trimmed mean [11] is less wasteful, trimming extremes dimension by dimension and approaching optimal rates under sub Gaussian noise. The genuinely hard part though is making Byzantine detection work alongside DP. The noise added for privacy can look a lot like poisoned perturbations. Xiang et al. [19] found that normalizing gradients before noise injection substantially helps separate honest from malicious updates. Perhaps the closest prior work to ours is Xia et al. [14], which jointly addresses DP, Byzantine resilience, and communication efficiency. But it targets convolutional networks on image benchmarks and does not layer cryptographic secure aggregation on top.

Recent vehicular FL work has moved closer to this combined privacy and integrity threat model, although usually with smaller perception models rather than language based command generation. Batch-Aggregate [20] reduces private aggregation overhead in VANETs and handles unstable vehicle connections, but it does not study Byzantine poisoning. RSAM [21] uses a private approximate median to combine secure aggregation with Byzantine robustness for Internet of Vehicles clients. DRL-PBFL [22] goes further by using deep reinforcement learning to weight vehicle updates while

protecting them from an honest but curious mobile edge computing node. These studies make the vehicular lesson fairly clear. Privacy and update integrity have to be considered together. Our work carries that lesson into Autonomous Aerial Vehicles command generation, where the model is a LoRA tuned LLM and the output must remain valid executable JSON under non IID data, secure aggregation, DP noise, and poisoned client updates.

Several recent secure FL studies are also useful for positioning the proposed framework. AeroVeil-FL [23] is close to the aerial setting, since it combines FL, adaptive DP, and mix networks for UAV assisted disaster detection. LSBlocFL [24] and the hierarchical protection and multiple aggregation scheme of Wang et al. [25] are relevant from a security design perspective because they pair FL with privacy mechanisms such as DP, homomorphic encryption, blockchain based coordination, or contribution aware aggregation. These papers support the layered security direction followed here, but their numerical results are not copied into our experimental tables because they evaluate image or general IoT learning tasks rather than LoRA tuned LLM command generation. We therefore include them in Table 1 and use them for qualitative positioning, while the quantitative comparisons in Section 5 remain restricted to methods executed by the authors on the same Autonomous Aerial Vehicles command dataset.

2.5. LLM Driven Autonomous Aerial Vehicles Systems

Natural language drone control has moved quickly from concept to prototype. Javaid et al. [4] mapped the emerging space in their survey. Tazir et al. [2] wired GPT-3.5 into PX4/Gazebo for free form flight commands, and Chen et al. [3] built TypeFly for real time typed control. Ferrage et al. [35] contributed 50,000 validated mission scenarios in structured JSON. Yet across this entire body of work, privacy during model training is almost never discussed. The one exception is Ayana et al. [15] which protects LLM inference via MPC but leaves federated fine tuning, the phase where raw operational data is most exposed, entirely unaddressed.

2.6. Research Gap

Table 1 shows the situation plainly. Some works achieve federated training but skip privacy. Others combine DP with Byzantine defenses but have never been applied to language models. Ayana et al. operates in the Autonomous Aerial Vehicles domain but only at inference time. No existing framework to the best of our knowledge unifies federated LLM fine tuning with DP, secure aggregation, and Byzantine resilience for Autonomous Aerial Vehicles command generation under non IID conditions. Because the closest prior studies use different objectives and datasets, copying their numerical scores would not give a fair comparison. We therefore use the literature to identify the closest reusable algorithmic components, then evaluate those components progressively on the same Autonomous Aerial Vehicles command generation benchmark in Section 5.

3. System Model and Threat Model

This section formalizes the system architecture and the adversarial assumptions against which the proposed framework is designed to operate.

3.1. System Architecture

Consider a fleet of n autonomous aerial vehicle nodes $\mathcal{U} = \{U_1, \ldots, U_n\}$ operating across diverse mission domains. Each vehicle carries a lightweight compute module running a frozen base language model augmented with a trainable adapter. A central edge server S coordinates federated training but never accesses raw mission data. The target task is translating natural language operator commands into structured JSON drone actions, a text to structured output problem that maps naturally onto causal language model fine tuning.

3.1.1. LoRA Based Parameter Efficient Adaptation

We employ Low Rank Adaptation (LoRA) [17] to constrain the trainable parameter space. For a pre trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$ in the transformer attention layers, LoRA decomposes the update as

$$W = W_0 + \Delta W = W_0 + AB \tag{1}$$

where $A \in \mathbb{R}^{d\times r}$, $B \in \mathbb{R}^{r\times k}$, and $r \ll \min(d, k)$. Only A and B are updated during training, reducing trainable parameters from dk to $r(d + k)$. For $d = k = 2048$ and $r = 16$, this is a reduction from ∼4.2M to ∼65.5K parameters per layer. In federated terms, the per round communication payload drops from approximately 4.4 GB (full 1.1B parameter model at 32 bit) to roughly 18 MB, making LLM federation over bandwidth constrained air to ground links feasible.

3.1.2. Federated Averaging Protocol

Training proceeds over T communication rounds. At round t, the server broadcasts the global adapter state $\theta^{(t)}_{global}$. Each client U_i performs E local epochs on its private dataset $\mathcal{D}_i$, producing $\theta^{(t)}_i$. The server then computes

$$\theta^{(t+1)}_{global} = \sum_{i=1}^{n} \frac{|\mathcal{D}_i|}{\sum_{j=1}^{n} |\mathcal{D}_j|} \cdot \theta^{(t)}_i \tag{2}$$

following McMahan et al. [5]. No raw data ever leaves any autonomous aerial vehicle node.

3.1.3. Non IID Data Distribution

Different autonomous aerial vehicles specialize in different missions, creating pronounced data heterogeneity. We model this via a Dirichlet distribution [36]. For C command categories, each client U_i draws a proportion vector

$$\mathbf{p}_i = (p_{i,1}, \ldots, p_{i,C}) \sim \mathrm{Dir}(\alpha, \ldots, \alpha) \tag{3}$$

where $\alpha > 0$ controls heterogeneity. Small α (e.g., 0.1) produces clients dominated by one or two command types. Large α (e.g., 10.0) approximates IID. Client U_i receives $|\mathcal{D}^{(c)}_i| = \lfloor p_{i,c} \cdot N_c \rfloor$ samples of type c, where N_c is the corpus wide count for that type.

3.2. Threat Model

Real world autonomous aerial vehicle operations face simultaneous risks from curious infrastructure operators, passive eavesdroppers, and compromised fleet members. We formalize three adversary classes and the corresponding defense mechanisms.

3.2.1. Adversary 1: Honest-but-Curious Server

The server $\mathcal{S}$ executes FedAvg correctly but may attempt to infer private information from individual updates $\theta_i^{(t)}$. This threat is concrete. Gradient inversion attacks [6] can reconstruct training data from model updates, potentially revealing mission coordinates or patrol routes. We deploy two layered defenses.

Client Level Differential Privacy. During local training, each client applies DP-SGD [8]. Per sample gradients are clipped to ℓ_2 norm C and Gaussian noise is injected at every optimizer step

$$\tilde{g}_i^{(t)} = \frac{1}{|\mathcal{B}|}\left(\sum_{x\in\mathcal{B}} \text{clip}\big(\nabla_\theta \ell(\theta, x),\, C\big) + \mathcal{N}\big(0,\, \sigma^2 C^2 \mathbf{I}\big)\right) \quad (4)$$

where $\text{clip}(g, C) = g \cdot \min(1,\, C/\|g\|_2)$ and σ is the noise multiplier. Cumulative privacy cost over T rounds is tracked via the Rényi DP (RDP) accountant [28], which converts to standard (ε, δ)-DP through

$$\varepsilon = \min_{\alpha>1}\left[\varepsilon(\alpha) + \frac{\log(1/\delta)}{\alpha - 1}\right] \quad (5)$$

Subsampling amplification from mini batch training (sampling rate $q = |\mathcal{B}|/|\mathcal{D}_i|$) substantially tightens this bound, as each step's RDP cost scales with $q^2\alpha/(2\sigma^2)$ rather than $\alpha/(2\sigma^2)$.

ECC Based Secure Aggregation. Even under DP, the noised updates remain individually visible to the server. Secure aggregation [9] ensures the server observes only the aggregate sum. Each client pair (U_i, U_j) with $i < j$ performs ECDH key exchange on curve $E(\mathbb{F}_p)$ with generator G and private keys $a_i, a_j \in \mathbb{Z}_q^*$, yielding shared secret

$$s_{i,j} = a_i(a_j G) = a_j(a_i G) = a_i a_j G \quad (6)$$

This seeds a PRG producing mask vectors. Client U_i transmits

$$\hat{\theta}_i^{(t)} = \tilde{\theta}_i^{(t)} + \sum_{j>i} \text{PRG}(s_{i,j}) - \sum_{j<i} \text{PRG}(s_{j,i}) \quad (7)$$

The pairwise masks cancel upon summation, so $\sum_{i=1}^n \hat{\theta}_i^{(t)} = \sum_{i=1}^n \tilde{\theta}_i^{(t)}$. Security reduces to the ECDDH assumption. No PPT adversary can distinguish (aG, bG, abG) from (aG, bG, cG) for random a, b, c with non negligible advantage. We derive round specific masks via HKDF to provide forward secrecy without additional round trips.

Table 2
Summary of adversary classes and corresponding defense mechanisms.

Adversary	Capability	Defense
Honest-but-curious server	Inspects individual updates during aggregation	DP-SGD (Eq. 4) + Secure aggregation (Eq. 7)
External eavesdropper	Intercepts air-to-ground transmissions	ECDH masking (Eq. 6) with forward secrecy
Byzantine clients	Sends arbitrary or poisoned updates	Krum (Eq. 8) or Trimmed Mean (Eq. 9)

3.2.2. Adversary 2: External Eavesdropper

A passive adversary $\mathcal{A}_{ext}$ intercepts air to ground transmissions under the Dolev Yao model [37]. The masked updates $\hat{\theta}_i^{(t)}$ are computationally indistinguishable from random to any party lacking the ECDH shared secrets. Ephemeral per round keys ensure that long term key compromise does not retroactively expose past communications.

3.2.3. Adversary 3: Byzantine Autonomous Aerial Vehicles Clients

Up to f out of n clients may be fully compromised, submitting arbitrary updates including random noise, sign flipped gradients ($-\lambda \cdot \theta_b^{(t)}$ for $\lambda > 1$), or updates trained on deliberately mislabeled data. We replace FedAvg with two Byzantine resilient alternatives.

Krum [10] selects the update closest to the population majority by computing, for each client i, the sum of squared distances to its $n - f - 2$ nearest neighbors

$$\text{score}(i) = \sum_{j\in\mathcal{N}_i} \|\theta_i^{(t)} - \theta_j^{(t)}\|^2 \quad (8)$$

and choosing the lowest scoring update. This tolerates $f < (n-2)/2$ Byzantine participants. Coordinate wise trimmed mean [11] operates per dimension, removing the f largest and smallest values before averaging

$$\theta_{global}^{(t+1)}[j] = \frac{1}{n-2f}\sum_{i\in\mathcal{R}_j} \theta_i^{(t)}[j] \quad (9)$$

where $\mathcal{R}_j$ is the set remaining after trimming for coordinate j. Trimmed mean is generally more resilient to coordinated attacks because it filters at parameter granularity rather than discarding entire updates.

Table 2 summarizes the adversary classes and their corresponding defenses. The composition of all three layers (DP, secure aggregation, Byzantine resilience) provides (ε, δ) differential privacy per client across T rounds, computational indistinguishability of individual masked updates under ECDDH, and bounded convergence degradation in the presence of up to f Byzantine nodes.

4. Proposed Framework

This section describes the concrete instantiation of the framework outlined in Section 3. We begin with an overview of the end to end pipeline, then detail the dataset construction model configuration federated training procedure with layered defenses and the evaluation protocol.

4.1. Framework Overview

The framework operates in four composable layers, each addressing a distinct threat class identified in Section 3.2. At the bottom sits LoRA based federated fine tuning via FedAvg (Eq. 2), which keeps raw data local. On top of it per step DP-SGD noise injection (Eq. 4) bounds the information any single update reveals about a client's training data. The third layer wraps each client's noised update in ECC based pairwise masks (Eq. 7) before transmission, ensuring the aggregation server sees only the aggregate sum. The fourth layer replaces standard averaging with a Byzantine resilient rule either Krum (Eq. 8) or coordinate wise trimmed mean (Eq. 9) to tolerate poisoned updates from compromised Autonomous Aerial Vehicles nodes.

One point worth emphasizing is that the secure aggregation layer is a communication level protocol. It produces mathematically identical aggregates to an unmasked round because the pairwise masks cancel during summation. It does not alter the model training dynamics. The remaining three layers by contrast all affect the optimization trajectory in some way and their interactions are what we study empirically in Section 5.

4.2. Dataset Construction

We constructed a domain specific dataset of 14,446 natural language to structured JSON command pairs for Autonomous Aerial Vehicles operations. The dataset is author-created and publicly available on Kaggle as the UAV Command Generation: Natural Language to Drone dataset [38]. Each sample consists of a free form English instruction (the input) and a machine executable JSON object (the output) that a flight controller could parse directly. The JSON schema includes seven top level fields, with `command_type` drawn from six Autonomous Aerial Vehicles mission categories and `action` specifying one of 34 operations across those categories. Table 3 shows the category level breakdown.

The benchmark problem is a text to structured command generation task. Given a natural language operator instruction, the model must generate a syntactically valid JSON command with the correct mission category, action, and parameter fields. The six benchmark categories are navigation, surveillance, inspection, emergency, formation, and status reporting. This setting is stricter than ordinary text generation because a fluent answer is not sufficient. The output must parse as JSON and preserve the command semantics needed by an autonomous aerial vehicle controller.

Samples were generated using OpenAI's GPT-4o-mini API in batches of 25, with a temperature of 0.9 to encourage lexical diversity. The system prompt enforced strict parameter ranges (altitude between 5 and 400 meters, speed between 1.0 and 25.0 m/s, and so on) and required variation across four formality levels ranging from terse military style commands to casual operator speech. Approximately 70% of the samples are single action commands, 25% chain multiple actions into compound sequences, and 5% are intentionally ambiguous inputs that require the model to flag missing information. Every generated sample underwent automated validation for JSON syntax parameter range compliance and taxonomy adherence. Near duplicate inputs (cosine similarity above 0.95) were removed.

Table 3
Distribution of Command Types in Dataset.

Command Type	Samples	Percentage
Navigation	4,334	30.0%
Surveillance	3,611	25.0%
Inspection	2,167	15.0%
Emergency	1,445	10.0%
Formation	1,445	10.0%
Status	1,444	10.0%
Total	**14,446**	**100%**

The final dataset was split into 12,033 training samples, 1,203 validation samples, and 1,210 test samples. The test set was held out entirely from all federated training and used only for final evaluation. This corresponds to approximately 83.3% training, 8.3% validation, and 8.4% test data. The split was stratified by `command_type` so that all six mission categories remained represented in the global training, validation, and test sets. Only the 12,033 training samples were redistributed among federated clients. The validation and test sets stayed as held out global evaluation sets and were never assigned to clients during local training. For each non IID setting, client partitions were generated from the training split by command type. For every class c with N_c training samples, a five dimensional allocation vector was sampled from $\mathrm{Dir}(\alpha, \ldots, \alpha)$, and samples from that class were assigned to the five clients according to the sampled proportions. We evaluated $\alpha \in \{0.1, 0.5, 1.0, 10.0\}$, where smaller values create mission specialized clients and $\alpha = 10.0$ gives a near IID partition.

4.3. Base Model and Adapter Configuration

We selected TinyLlama-1.1B-Chat-v1.0 [16] as the base model. With 1.1 billion parameters, it is small enough to plausibly run on Autonomous Aerial Vehicles edge hardware while still large enough to handle structured generation tasks. The chat variant was pre trained on 3 trillion tokens and instruction tuned with a system/user/assistant template that maps naturally onto our task format.

The base model weights remain frozen throughout training. Only the LoRA adapter matrices are updated, with rank $r = 16$, scaling factor $\alpha = 32$, and dropout 0.05. We target four attention projection matrices per transformer layer (query, key, value, and output projections), yielding approximately 4.5 million trainable parameters out of the

model's 1.1 billion total. The adapter update, roughly 18 MB in float32, is what gets exchanged each federated round.

For DP enabled training, we apply 4 bit NormalFloat (NF4) quantization [39] to the frozen base weights with bfloat16 compute precision and double quantization enabled. This reduces GPU memory consumption enough to accommodate both the quantized base model and the gradient clipping buffers on a single accelerator. We found empirically that running the DP configuration without quantization caused gradient overflow (NaN values) on both A100 and H100 hardware due to the interaction between bfloat16 arithmetic, gradient clipping at $\|\cdot\|_2 \leq 1.0$, and the noise injection callback. We did not observe the same overflow in the non private federated runs, which completed in native bfloat16 without NF4 because they did not combine per step clipping, DP noise injection, and the privacy callback. The centralized T4 baseline used NF4 mainly for memory efficiency, not because non private training itself was numerically unstable.

4.4. Federated Training with Layered Defenses

Training is organized into $T = 20$ communication rounds across $n = 5$ simulated Autonomous Aerial Vehicles clients. Each client trains for $E = 1$ local epoch per round using the AdamW optimizer. The learning rate, batch size, clipping, precision, and hardware settings for each experiment group are reported in Table 4. Training data is partitioned across clients using the Dirichlet distribution (Eq. 3) with concentration parameter α controlling the degree of non-IID skew.

Algorithm 1 presents the complete training procedure. A few implementation choices deserve explanation.

4.4.1. Per Step Noise Injection and Privacy Accounting

The distinction between per step and per round noise injection turns out to be critical for this framework, and getting it wrong was the source of considerable difficulty in our implementation. The naive approach would be to let each client train normally, compute the full update delta $\Delta\theta_i^{(t)} = \theta_i^{(t)} - \theta_{global}^{(t-1)}$, and add calibrated noise to $\Delta\theta_i^{(t)}$ once per round. But this approach treats the entire local training trajectory as one composition unit. Without mini batch sub sampling amplification, the noise multiplier required to achieve any reasonable ε becomes enormous. In our experiments, per round noise at $\varepsilon = 8$ produced a noise to signal ratio exceeding 4,700 to 1, causing complete model collapse by the second round.

Per step injection avoids this problem. We implement it through a training callback that fires after every optimizer step, adding $\mathcal{N}(0, \eta^2\mathbf{I})$ directly to the LoRA parameters. Here $\eta = \mathrm{lr} \cdot \sigma \cdot C/B$ converts the DP-SGD noise multiplier σ into parameter space, where $C = 1.0$ is the gradient clipping norm, lr is the learning rate, and B is the effective batch size. Because each step processes a mini batch sampled with rate $q = B/|\mathcal{D}_i| \approx 0.027$, the RDP accountant credits the subsampled Gaussian mechanism at each step with cost approximately $q^2\alpha/(2\sigma^2)$ rather than $\alpha/(2\sigma^2)$. Over all steps across all rounds, this composes to the target ε with dramatically smaller noise.

We determine σ through binary search. Given the target ε, failure probability $\delta = 10^{-5}$, total training steps per client $S = T \cdot \lceil |\mathcal{D}_i|/B \rceil$, and sampling rate q, the search finds the smallest σ such that

$$\min_{\alpha > 1} \left[\sum_{s=1}^{S} \frac{q^2 \alpha}{2\sigma^2} + \frac{\log(1/\delta)}{\alpha - 1} \right] \leq \varepsilon \tag{10}$$

Eq. 10 is the stopping condition for this binary search over the noise multiplier. The minimization runs over a grid of Rényi orders $\alpha \in \{1.25, 1.5, 2, 3, \ldots, 64\}$. For $\varepsilon = 8$ with our training configuration, this yields $\sigma \approx 0.502$ and a per parameter noise standard deviation of approximately 1.57×10^{-6}.

Algorithm 1 Privacy Preserving Federated Fine Tuning with Layered Defenses

Require: Clients $\{U_1, \ldots, U_n\}$ with local datasets $\{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$, rounds T, local epochs E, privacy budget (ε, δ), number of Byzantine clients f, aggregation rule $\mathcal{A} \in \{\text{FedAvg}, \text{Krum}, \text{TrimmedMean}\}$

Ensure: Trained global LoRA adapter $\theta_{global}^{(T)}$

Server: Initialize global LoRA adapter $\theta_{global}^{(0)}$
Server: Coordinate ECDH key exchange among all $\binom{n}{2}$ client pairs
Compute noise multiplier σ via binary search over RDP accountant (Eq. 5) to satisfy (ε, δ) over $T \cdot S$ total steps
Compute per-step parameter noise $\eta = \mathrm{lr} \cdot \sigma \cdot C/B$ {C: clip norm, B: batch size}
for round $t = 1$ to T **do**
  Server: Broadcast $\theta_{global}^{(t-1)}$ to all clients
  for each client U_i in parallel **do**
    $\theta_i^{(t)} \leftarrow \theta_{global}^{(t-1)}$
    for each optimizer step in E local epochs **do**
      Clip per-sample gradients: $\bar{g} \leftarrow \mathrm{clip}(\nabla_\theta \ell, C)$
      Update parameters via AdamW
      Add DP noise: $\theta_i^{(t)} \leftarrow \theta_i^{(t)} + \mathcal{N}(0, \eta^2\mathbf{I})$ {per-step injection}
    end for
    Derive round-t mask seeds via HKDF from ECDH shared secrets
    $\hat{\theta}_i^{(t)} \leftarrow \theta_i^{(t)} + \sum_{j>i} \mathrm{PRG}(s_{i,j}, t) - \sum_{j<i} \mathrm{PRG}(s_{j,i}, t)$ {SecAgg masking}
    Transmit $\hat{\theta}_i^{(t)}$ to server
  end for
  Server: Collect all $\hat{\theta}_i^{(t)}$ {pairwise $+\mathrm{PRG}(s_{i,j}, t)$ and $-\mathrm{PRG}(s_{i,j}, t)$ masks cancel in $\sum_i \hat{\theta}_i^{(t)}$}
  Server: $\theta_{global}^{(t)} \leftarrow \mathcal{A}(\{\hat{\theta}_1^{(t)}, \ldots, \hat{\theta}_n^{(t)}\}, f)$
end for

4.4.2. Secure Aggregation Protocol

The ECC based secure aggregation protocol described in Section 3.2 is instantiated over the NIST P-256 (secp256r1) curve, providing 128 bit equivalent security with compressed public keys of only 33 bytes. Each client pair performs ECDH key exchange once at initialization. The raw shared secret is then passed through HKDF-SHA256 to derive a 256 bit symmetric key. Rather than reusing this key directly, we feed it through a second HKDF derivation with a round specific info string (`round-`t) for each communication round. This ensures that even if an adversary obtains the mask used in round t, it cannot reconstruct masks from earlier or later rounds, providing forward secrecy at no additional communication cost.

The derived 256 bit key seeds a pseudorandom number generator that produces a mask vector of the same dimensionality as the LoRA update (∼4.5 million float32 values, approximately 17.19 MB). The masking and unmasking procedure adds $O(n^2)$ computational overhead per round, which in our five client configuration amounts to roughly 2.9 seconds per round, or about one minute across all 20 rounds. Section 5 reports detailed scaling measurements.

4.4.3. Byzantine Resilient Aggregation

We evaluate two aggregation rules against three attack types. The Krum rule (Eq. 8) selects a single update per round based on proximity to the population majority. Coordinate wise trimmed mean (Eq. 9) discards the f most extreme values at each parameter coordinate before averaging. We set f equal to the number of known (or assumed) Byzantine clients.

The three attack models simulate progressively more sophisticated adversaries. Random noise injection replaces the Byzantine client's update with $\theta^{(t)}_{global} + \mathcal{N}(0, 25 \cdot \mathbf{I})$, a high variance perturbation with standard deviation $\sigma_{attack} = 5.0$. Sign flip inversion computes the honest update delta $\Delta\theta = \theta^{(t)}_{b} - \theta^{(t-1)}_{global}$, negates it and amplifies by a factor $\lambda = 2.0$, sending $\theta^{(t-1)}_{global} - \lambda \cdot \Delta\theta$. This is the strongest generic untargeted attack because it directly opposes the direction of honest learning. Targeted poisoning is more subtle. The compromised client trains on deliberately mislabeled data in which emergency commands are relabeled as navigation and vice versa. For example, a poisoned pair may take the input "return to base immediately because the battery is critical" and attach `command_type=Navigation`, rather than the intended `command_type=Emergency`. The resulting update looks structurally similar to honest ones but steers the model toward dangerous misclassifications.

An important interaction arises when combining DP noise with Byzantine detection. The DP perturbations added during honest training could, in principle, make honest updates look suspicious to a distance based rule like Krum. Whether this actually happens in practice depends on the relative magnitudes of DP noise (on the order of 10^{-6} per parameter in our setting) versus the deviations introduced by Byzantine attacks (on the order of 10^0 to 10^1 per parameter for sign flip). Our experiments in Section 5 confirm that at the noise levels produced by per step DP in the LoRA parameter regime, the Byzantine detectors remain effective.

4.5. Protocol Evaluation

We evaluate trained models using four complementary metrics that capture different aspects of Autonomous Aerial Vehicles command generation quality.

BLEU score [40] measures character level n gram overlap between the predicted JSON string and the ground-truth JSON string, with smoothing applied to handle short sequences. While BLEU was originally designed for machine translation, it serves here as a proxy for structural fidelity of the generated JSON.

JSON validity is the percentage of generated outputs that parse as syntactically correct JSON. A model that produces high BLEU but low JSON validity would be useless in practice because the flight controller cannot execute a malformed command. This metric is binary per sample and is arguably the most safety critical of the four.

Command type accuracy measures whether the predicted `command_type` field matches the ground truth, computed only over samples with valid JSON. Confusing an emergency command with a navigation command for instance, would be a potentially dangerous error.

Action accuracy measures whether the predicted `action` field (the specific operation within a command type) matches the ground truth again computed only over valid JSON outputs. This is finer grained than command type accuracy and captures whether the model selects the correct operational behavior.

For example, an output with `command_type = Emergency`, `action = return_to_base`, and `altitude = 80` is counted as JSON valid if the object parses correctly, as command type correct if `Emergency` matches the reference label, and as action correct if `return_to_base` is also the reference action.

Intermediate evaluations during training use 100 randomly selected test samples every 5 communication rounds to track convergence. The final evaluation reported in Section 5 uses 500 test samples for the full framework configurations, while the centralized baseline is evaluated on the complete test set of 1,210 samples.

5. Experimental Setup and Results

This section presents the experimental configuration and results across six experiment groups organized in order of increasing framework complexity. We begin with the centralized and federated baselines, then layer on differential privacy, secure aggregation, and Byzantine resilience individually, before evaluating the complete framework with all defenses active simultaneously.

For the comparative study, all reported numerical results are executed by the authors on the same author-created 14,446 sample Autonomous Aerial Vehicles command dataset [38] and the same train, validation, and test split. The baselines are selected from the closest algorithmic families in the literature: centralized LoRA fine

Table 4
Hyperparameter Configuration for Each Experiment Group. Method abbreviations follow FedAvg [5], DP-SGD [8], Rényi DP accounting [28], and SecAgg [9] where applicable.

Parameter	Cent.	FedAvg	DP-Fed	Full
Rds / Epochs	3 ep	20 rds	20 rds	20 rds
Local epochs	—	1	1	1
Batch size	4×4	64	16×4	64
Learning rate	2e-4	5e-4	2e-4	5e-4
Quantization	NF4	None	NF4	None
Grad clip	0	0	1.0	1.0
Precision	fp32	bf16	bf16	bf16
Hardware	T4	H100	A100	A100

tuning, FedAvg [5], DP-SGD with Rényi accounting [8, 28], pairwise secure aggregation following Bonawitz et al. [9], and Byzantine resilient aggregation through Krum [10] and trimmed mean [11]. This progressive design makes the novelty claim testable. Each defense layer is first measured under the same data and model conditions, and the final experiment then evaluates the combined privacy, secure aggregation, and Byzantine resilient pipeline.

To avoid ambiguity, no numerical result in the experimental tables or figures is copied from a previous publication. Prior work is cited to identify the algorithmic origin of each baseline, while every reported score, runtime, privacy cost, and secure aggregation overhead was produced by our implementation under the experimental protocol described below.

To keep the main manuscript focused, this section retains the result tables and figures needed for the comparative study, ablation interpretation, and deployment discussion. More detailed execution material is kept in the public experiment notebooks described in the Data and Code Availability subsection, rather than repeated as secondary tables in the main text.

5.1. Experimental Configuration

All experiments use TinyLlama-1.1B-Chat-v1.0 [16] with LoRA adapters ($r = 16$, $\alpha = 32$, dropout 0.05) targeting the query, key, value, and output projection matrices across all 22 transformer layers, yielding 4,505,600 trainable parameters (0.73% of the model). Federated experiments simulate $n = 5$ Autonomous Aerial Vehicles clients over $T = 20$ communication rounds (15 rounds for isolated Byzantine experiments) with $E = 1$ local epoch per round. Data is partitioned via the Dirichlet distribution (Eq. 3) with varying concentration parameter α. Table 4 summarizes the key hyper parameters for each experiment group.

The hardware progression in Table 4 mainly records the execution environment rather than a change in the learning objective. The centralized baseline used a Kaggle T4 with NF4 quantization, the non private federated runs used an H100 for faster multi round LoRA training, and the DP enabled experiments were run on an A100 with NF4 because that setting needed clipping, noise injection, and the privacy callback. As noted earlier, NF4 was the stabilizing choice after unquantized DP runs produced overflow on both A100 and H100 hardware.

Fair tuning used the validation split only, and the test set was not used for choosing hyperparameters. For a given Dirichlet α, all compared methods used the same training, validation, and test split and the same client partition. We first fixed the backbone, tokenizer, LoRA target modules, LoRA rank $r = 16$, LoRA scaling $\alpha = 32$, dropout 0.05, AdamW, $T = 20$ rounds, and $E = 1$ local epoch for all federated methods. The methods then changed only the parameter required by the defense being tested. FedAvg used learning rate 5×10^{-4} with batch size 64. Isolated DP-FedAvg used learning rate 2×10^{-4} with effective batch size 16×4, clipping norm $C = 1.0$, $\delta = 10^{-5}$, and $\varepsilon \in \{1, 4, 8, 16\}$, with the resulting σ values reported in the DP-FedAvg results table. Full-framework runs used the same LoRA configuration, $T = 20$, $E = 1$, learning rate 5×10^{-4}, clipping norm $C = 1.0$, and $\varepsilon = 8$. Krum and trimmed mean used the same client updates as FedAvg and were given the same assumed Byzantine count f as the attack setting. Secure aggregation had no model hyperparameter to tune because correct mask cancellation leaves the aggregate unchanged.

The centralized baseline ran on a Kaggle T4 GPU with 4-bit NF4 quantization. Non private federated experiments ran on a Google Colab H100 with native bfloat16 and Scaled Dot Product Attention (SDPA). DP enabled experiments required 4 bit NF4 quantization on an A100 due to gradient overflow under bfloat16 arithmetic when combined with gradient clipping at $\| \cdot \|_2 \leq 1.0$ and noise injection. Secure aggregation benchmarks ran on CPU since the cryptographic operations require no GPU acceleration.

We report four evaluation metrics as defined in Section 4.5. Intermediate evaluations during training use 100 randomly selected test samples every 5 communication rounds. The centralized baseline is evaluated on the full 1,210 sample test set. Final full framework evaluations use 500 samples for more reliable estimates.

5.2. Centralized Baseline

The centralized baseline establishes an upper bound on what the model can achieve with unrestricted access to all 12,033 training samples. No federation, no privacy mechanisms, no adversaries. After three epochs of standard LoRA fine tuning, the model reached a final training loss of 0.3007 in 76.6 minutes. Fig. 1 shows the loss curve, which drops sharply in the first 200 steps and plateaus around 0.26 by epoch 3. Validation loss tracks training loss closely throughout, with no sign of overfitting.

On the full test set, the centralized model achieves a BLEU score of 0.8618, with perfect JSON validity (100%), 99.4% command type accuracy, and 88.4% action accuracy. These numbers serve as the reference point for every subsequent experiment. Any degradation relative to this baseline is the cost of federation, privacy, or adversarial tolerance.

Table 5 provides a per category breakdown that reveals interesting variation. Emergency and status commands are easiest for the model, with BLEU scores of 0.9234 and 0.9261 respectively and action accuracy above 94%. These

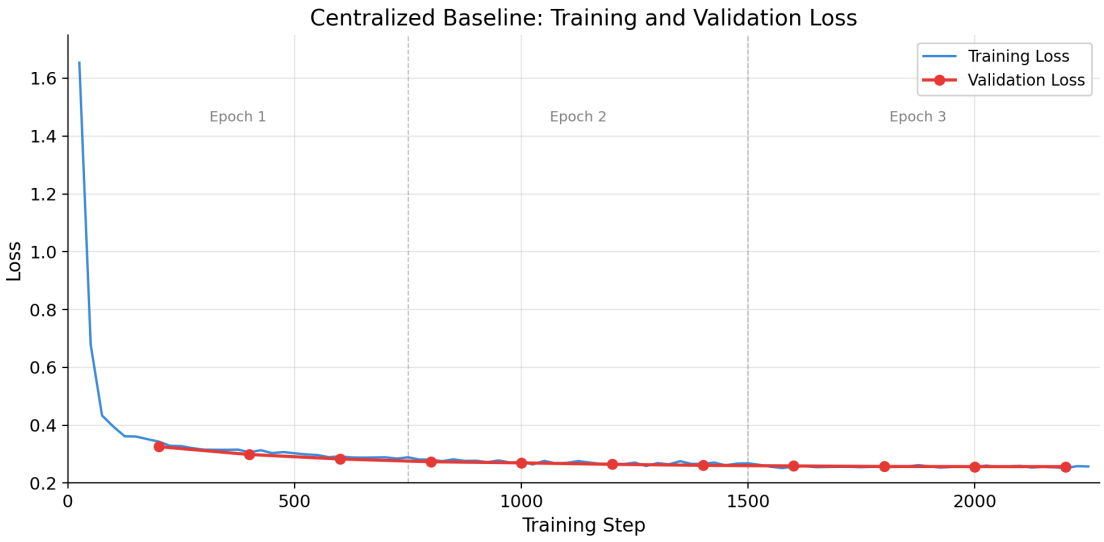

Figure 1: Centralized Baseline Training and Validation Loss Over 2,250 Steps (3 epochs).

Table 5
Per Category Performance

Category	n	BLEU ↑	JSON% ↑	CmdT% ↑	Act% ↑
Navigation	167	0.8373	100.0	99.4	80.8
Surveillance	104	0.8565	100.0	98.1	88.5
Inspection	88	0.8516	100.0	100.0	89.8
Emergency	43	0.9234	100.0	100.0	97.7
Formation	42	0.8511	100.0	100.0	97.6
Status	50	0.9261	100.0	100.0	94.0
Overall	**494**	**0.8618**	**100.0**	**99.4**	**88.4**

Table 6
FedAvg [5] Results at Round 20 for Varying Dirichlet α (100 Sample Eval). The longer runtime at $\alpha = 0.1$ comes from severe non IID partitioning, where one or two clients receive much larger local datasets. Boldface marks the best value in each metric column.

α	BLEU ↑	CmdType% ↑	Action% ↑	Time (min) ↓
0.1	**0.8642**	**99.0**	86.0	159.4
0.5	0.8595	98.0	84.0	**45.6**
1.0	0.8567	98.0	84.0	46.0
10.0	0.8641	98.0	**87.0**	46.1

categories tend to have simpler JSON structures with fewer parameter fields. Navigation is the hardest, with action accuracy at 80.8% and BLEU at 0.8373, likely because navigation commands involve more complex parameter combinations (latitude, longitude, altitude, speed, heading). This variation matters for the non IID federated experiments, where some clients are dominated by the harder categories.

5.3. Federated Averaging Under Data Heterogeneity

The first question is how much performance federation alone costs, and whether the degree of non IID skew has a measurable impact. We trained with FedAvg across four Dirichlet concentration values ($\alpha \in \{0.1, 0.5, 1.0, 10.0\}$) for 20 communication rounds. Fig. 2 shows the training loss convergence per round, while Fig. 3 compares BLEU scores at evaluation checkpoints across heterogeneity levels.

Table 6 summarizes the final round 20 metrics. The results are somewhat surprising. All four heterogeneity settings converge to similar final performance. At round 20, $\alpha = 0.1$ (severe non IID) achieves BLEU 0.8642, while $\alpha = 10.0$ (near IID) reaches 0.8641. The difference is within sampling noise. JSON validity remains at 100% across all settings. Command type accuracy ranges from 98.0% to 99.0%, and action accuracy from 84.0% to 87.0%. The gap between any federated configuration and the centralized baseline (BLEU 0.8618) is less than 0.006.

Where heterogeneity does matter is convergence speed. As Fig. 3 shows, at round 5 $\alpha = 0.1$ lags at BLEU 0.8202 while $\alpha = 10.0$ is already at 0.8641. The severely skewed setting needs roughly twice as many rounds to catch up. For applications where communication rounds are expensive, such as Autonomous Aerial Vehicles fleets with limited radio bandwidth, this convergence delay is operationally relevant even if the final accuracy is comparable. LoRA's low rank constraint appears to limit the hypothesis space enough that heterogeneous local datasets still arrive at a similar global optimum given sufficient rounds. This should not be read as a universal claim. With a larger base model, a higher LoRA rank, or a broader mission vocabulary, the adapter space may be wide enough for heterogeneity to pull clients toward different optima.

5.4. Differential Privacy and the Privacy Utility Tradeoff

We evaluate DP-FedAvg in two experiment sets. Experiment A fixes data heterogeneity at $\alpha = 0.5$ and varies the privacy budget $\varepsilon \in \{1, 4, 8, 16\}$, isolating the effect

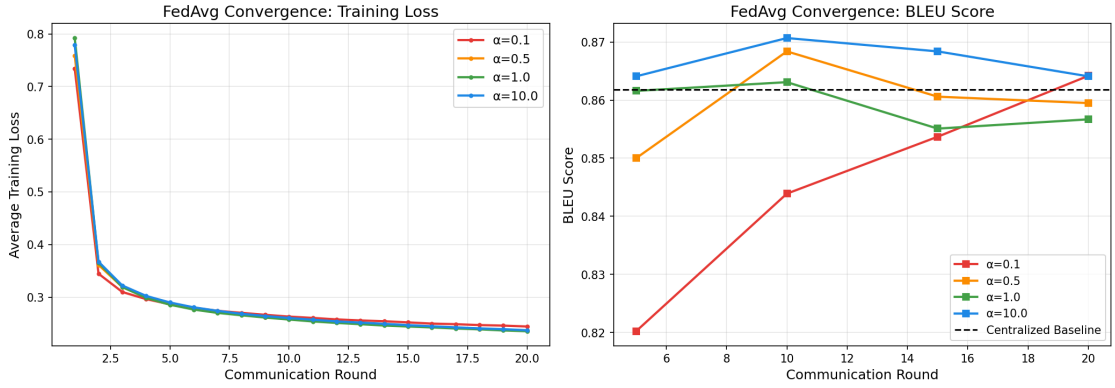

Figure 2: FedAvg Training Loss Convergence

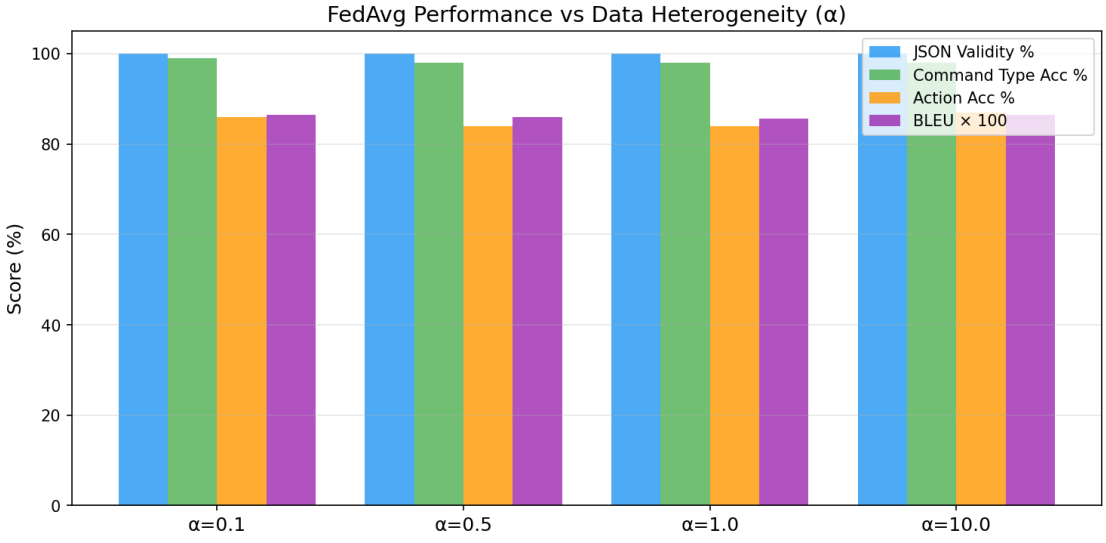

Figure 3: FedAvg BLEU Scores at Evaluation Checkpoints (Rounds 5, 10, 15, 20) for Different Dirichlet α Values.

Table 7
DP-FedAvg Results With DP-SGD [8] and Rényi DP Accounting [28] Under Varying Privacy Budget ($\alpha = 0.5$, 100 Sample Eval). σ is Noise Multiplier, η the Per Parameter Noise Standard Deviation. Boldface marks the best value in each utility metric column.

ε	σ	η	BLEU ↑	CmdType% ↑	Action% ↑
1	3.645	1.14×10^{-5}	0.8662	**98.0**	81.0
4	0.943	2.95×10^{-6}	0.8634	**98.0**	83.0
8	0.502	1.57×10^{-6}	0.8626	**98.0**	**84.0**
16	0.277	8.70×10^{-7}	**0.8672**	**98.0**	81.0

of DP noise. Experiment B fixes $\varepsilon = 8$ and varies $\alpha \in \{0.1, 0.5, 1.0, 10.0\}$, isolating the interaction between DP noise and data heterogeneity.

5.4.1. Experiment A: Varying Privacy Budget

Table 7 shows the results. Fig. 4 visualizes the privacy utility tradeoff and the convergence behavior.

Put simply, the privacy setting changes the formal leakage bound, but in this LoRA setup it barely changes the quality of the generated JSON commands.

The most notable finding from Table 7 is the near absence of a privacy utility tradeoff. Even at $\varepsilon = 1$, widely considered a strong privacy guarantee, the model achieves BLEU 0.8662, which is actually slightly above the $\varepsilon = 8$ result of 0.8626. The differences across all four epsilon values fall within the noise floor of the 100 sample evaluation. This is not a bug. The per parameter noise standard deviation ranges from 1.14×10^{-5} at $\varepsilon = 1$ down to 8.70×10^{-7} at $\varepsilon = 16$. LoRA's low rank structure (only 4.5 million parameters) combined with mini batch subsampling amplification ($q \approx 0.027$) reduces the actual perturbation per parameter per step to orders of magnitude below the learning signal. The right panel of Fig. 4 confirms this visually. All four convergence curves are essentially indistinguishable.

5.4.2. Experiment B: DP Under Varying Heterogeneity

Fig. 5 tells a different story. When we fix $\varepsilon = 8$ and vary the Dirichlet concentration, the spread in BLEU is

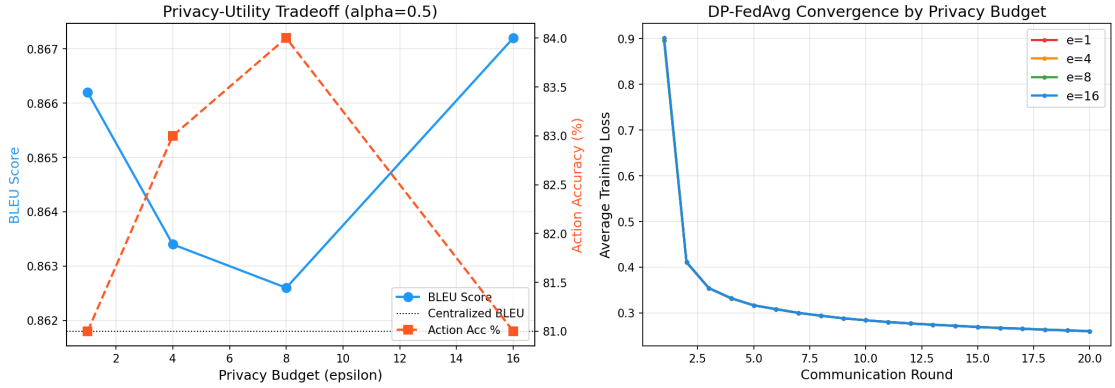

Figure 4: Left Panel Shows BLEU Score and Action Accuracy as a Function of Privacy Budget ε ($\alpha = 0.5$). Right panel Shows DP-FedAvg Training Loss Convergence For All Four ε Values.

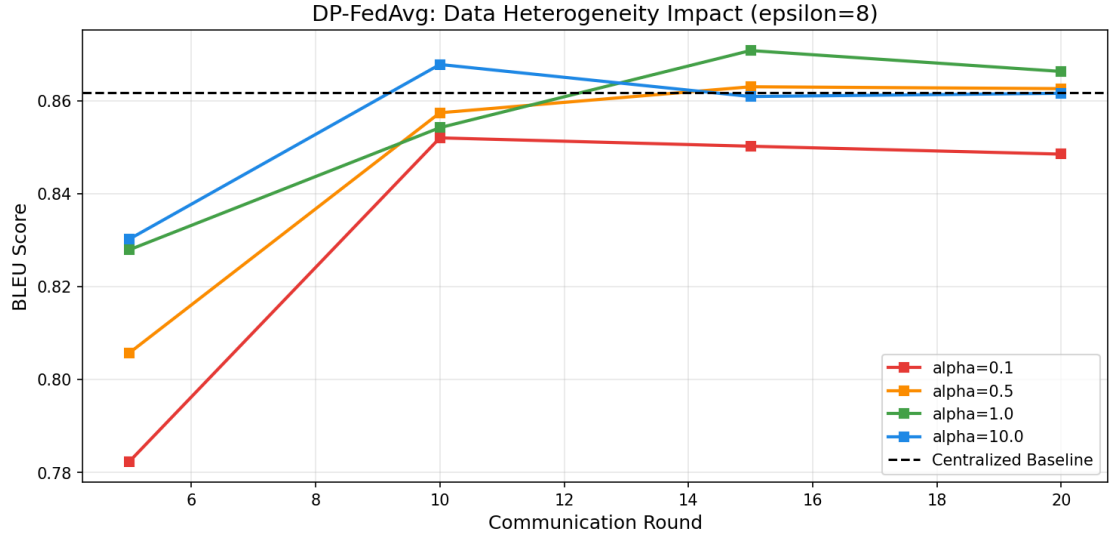

Figure 5: DP-FedAvg BLEU Scores at Evaluation Checkpoints for Varying Dirichlet α at Fixed $\varepsilon = 8$.

considerably larger than anything observed in Experiment A. At $\alpha = 0.1$, BLEU drops to 0.8485 and command type accuracy falls to 95.0%. At $\alpha = 1.0$, BLEU recovers to 0.8663 with 98.0% command type accuracy. The near IID setting ($\alpha = 10.0$) reaches 0.8616 with 99.0% command type accuracy.

The conclusion from these two experiment sets taken together is clear. In the LoRA parameter regime with per step noise injection and subsampling amplification, data heterogeneity (α) is the dominant factor affecting model quality, not the privacy budget (ε). For deployments that are not already constrained by a mandated privacy budget, this suggests that improving data balance across Autonomous Aerial Vehicles clients may give a larger utility gain than relaxing ε. Privacy bound deployments, of course, may still need to choose the privacy target first and then optimize the federation around it.

Table 8
Secure Aggregation Overhead by Fleet Size Using Pairwise SecAgg [9]. Update Size is 17.19 MB for All.

n	Pairs	Mask (ms/rd) ↓	ECDH (ms) ↓	20-Rd (s) ↓
3	3	895	0.81	17.9
5	10	2,916	2.07	58.3
7	21	6,164	4.25	123.3
10	45	13,375	8.62	267.5
15	105	30,955	18.96	619.1
20	190	56,175	33.17	1,123.5

5.5. Secure Aggregation Overhead

Secure aggregation does not change the model or its training dynamics. It is a communication level protocol that produces mathematically identical aggregates to an unmasked round. What matters is the computational and communication overhead it introduces. Table 8 gives the corresponding overhead values for fleet sizes ranging from 3 to 20 clients, with the LoRA update size fixed at 17.19 MB (4,505,600 float32 parameters). Fig. 6 visualizes the scaling behavior.

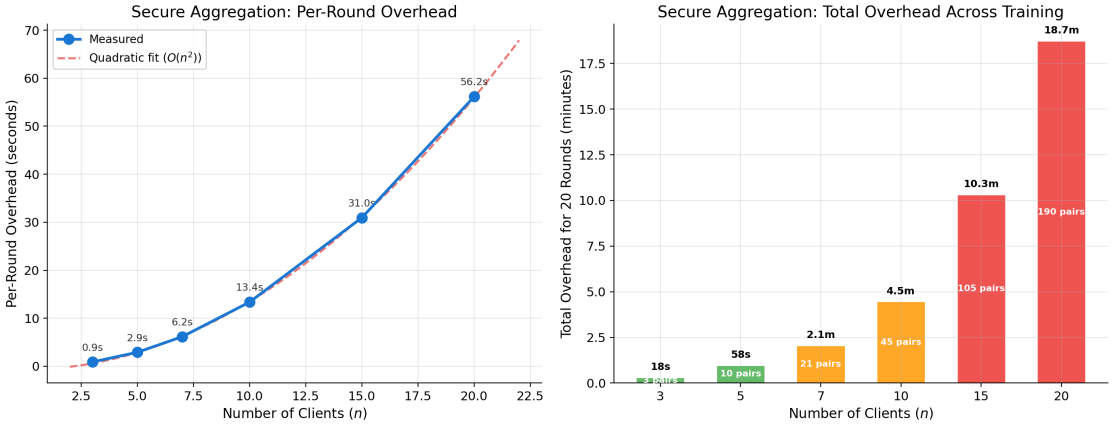

Figure 6: Per Round Secure Aggregation Overhead (Left) and Cumulative 20 Round Overhead (Right) as a Function of Fleet Size n, Confirming $O(n^2)$ Scaling via Quadratic Fit.

The $O(n^2)$ scaling is clearly visible in both Table 8 and Fig. 6. Moving from 5 to 10 clients roughly quadruples the per round masking time, from 2.9 s to 13.4 s. For our five client fleet, the total overhead across 20 rounds is 58.3 seconds, roughly 2.4% of the federated training time (∼40 minutes per experiment). The ECDH key exchange itself is negligible (2.07 ms for 10 pairwise exchanges at $n=5$), and the masked update is exactly the same size as the unmasked update (17.19 MB), so there is zero additional bandwidth cost.

At 20 clients, the overhead reaches 18.7 minutes across 20 rounds, which may become problematic for operational fleets on tight mission schedules. Poly logarithmic protocols such as those proposed by Bell et al. [33] would be needed for deployments beyond roughly 15 clients.

Correctness was verified for all fleet sizes. The maximum floating point error between masked and unmasked aggregation was below 10^{-3} in every configuration, well within float32 precision.

The asymptotic cost of the compared aggregation choices is also useful for deployment planning. Let $d = 4{,}505{,}600$ denote the number of trainable LoRA parameters and let n denote the number of clients. FedAvg has server side aggregation cost $O(nd)$ per round and client upload volume $O(d)$ per client. DP-FedAvg keeps the same aggregation and communication order, but adds per step clipping and Gaussian perturbation during local training. In our implementation this cost is dominated by the ordinary LoRA backward pass rather than by server aggregation. Pairwise secure aggregation adds $O(n^2)$ key agreements and $O(n^2 d)$ mask generation across the fleet, while preserving the same upload size because the masked update has the same dimension as the unmasked LoRA update. Krum adds $O(n^2 d)$ server side distance computations because every client update is compared with every other update. Coordinate wise trimmed mean costs $O(d\, n \log n)$ with sorting per coordinate, or $O(dn)$ if selection is implemented without a full sort. Thus the full framework has the same LoRA training cost as FedAvg but adds the quadratic secure aggregation and Krum terms. This is acceptable for the five client fleet studied here, yet it explains why larger autonomous aerial vehicle deployments would need lower overhead secure aggregation and possibly more scalable robust aggregation rules.

5.6. Byzantine Resilience Without Differential Privacy

Before combining all defense layers, we isolate the impact of Byzantine attacks on federated LoRA fine tuning without any DP noise. We test three aggregation rules (FedAvg, Krum, trimmed mean) against three attack types (random noise with $\sigma_{attack}=5.0$, sign flip with $\lambda=2.0$, targeted poisoning via emergency navigation label swap) with 0, 1, or 2 Byzantine clients out of 5. This yields 21 experimental configurations in total. The two Byzantine case should be read with this fleet size in mind: for trimmed mean, $n=5$ and $f=2$ leaves only $n-2f = 1$ coordinate value after trimming, so the rule largely loses the averaging effect it would have in a larger fleet.

Fig. 7 shows the BLEU comparison across aggregation methods and attack types. Fig. 8 shows the training loss convergence under sign flip attacks specifically, which is the strongest generic untargeted attack in our test suite. In both figures, the 1 Byzantine and 2 Byzantine settings correspond to 20% and 40% compromised clients in a five client fleet. This distinction is important because one additional compromised drone changes the corruption level substantially in such a small autonomous aerial vehicle group.

The results reveal a stark asymmetry. FedAvg offers essentially zero protection against random noise injection. This statement refers specifically to the random noise attack with $\sigma_{attack}=5.0$, evaluated at round 15 in Fig. 7. A single Byzantine client with $\sigma_{attack} = 5.0$ drives FedAvg's BLEU to 0.0272 with 0% JSON validity by round 15. The model produces gibberish. Sign flip with one attacker is partially tolerated by FedAvg (BLEU 0.8432 at round 15), likely because the sign flip perturbation at $\lambda = 2.0$ with only one adversary among four honest contributions is not large enough to fully overwhelm the weighted average. But with

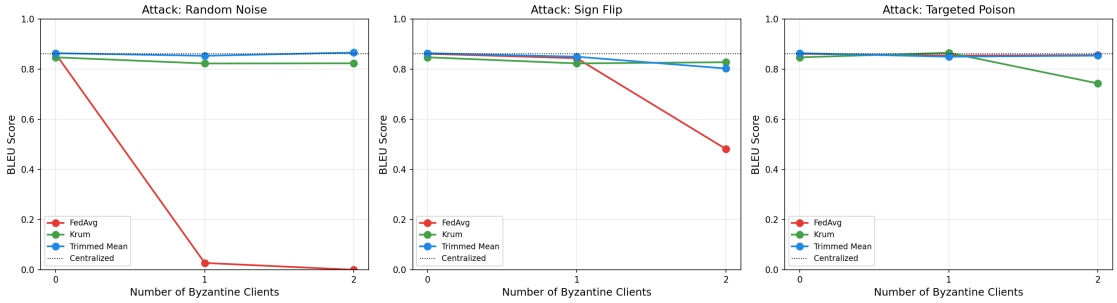

Figure 7: BLEU Scores By Aggregation Method And Attack Type At Round 15. The compared aggregation methods are FedAvg [5], Krum [10], and Trimmed Mean [11]. FedAvg Collapses Under Random Noise (BLEU $\approx$ 0.03) While Krum And Trimmed Mean Remain Stable Across All Attack Types And Corruption Levels.

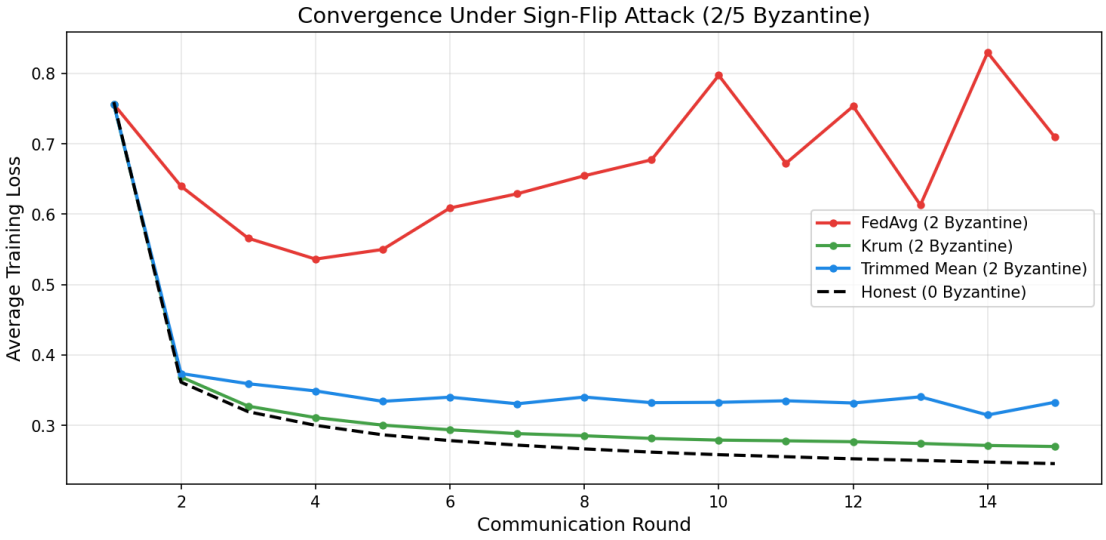

Figure 8: Training Loss Convergence Under Sign Flip Attack With Varying Byzantine Clients. FedAvg [5], Krum [10], and Trimmed Mean [11] are compared. FedAvg Diverges While Krum And Trimmed Mean Converge Comparably To The Honest Baseline.

two sign flip attackers, FedAvg performance degrades more substantially.

Krum and trimmed mean both survive all three attack types at both 20% (1/5) and 40% (2/5) corruption rates. In the no attack baseline, trimmed mean (BLEU 0.8631) slightly outperforms Krum (BLEU 0.8467). This gap exists because Krum discards four out of five client contributions each round, using only a single update, which slows convergence. Trimmed mean retains more honest information per round.

Under 2 Byzantine sign flip clients (40% corruption), Krum and trimmed mean perform comparably. This makes sense for a reason specific to the small fleet size. With $n=5$ and $f = 2$, trimmed mean retains only $n - 2f = 1$ value per coordinate after removing extremes, losing the averaging benefit that normally gives it an advantage over Krum.

Targeted poisoning (emergency navigation label swap) is the most subtle attack. All three aggregation methods show some performance reduction, but Krum and trimmed mean keep the model functional. This type of attack produces updates that look structurally similar to honest ones, making geometric detection harder.

5.7. Full Framework Results

We now activate all four defense layers simultaneously. The full framework combines per step DP noise ($\varepsilon = 8$, $\delta = 10^{-5}$), ECC based secure aggregation, and Byzantine resilient aggregation under sign flip attack ($\lambda = 2.0$) with $\alpha = 0.5$ data heterogeneity. Six experiments were run, with the primary results evaluated on 500 test samples for higher statistical reliability.

Table 9 presents the main results of this paper. Fig. 9 shows the convergence behaviour across configurations, and Fig. 10 presents the ablation across defense layers. The final row in Table 9 is the only configuration where JSON validity drops sharply, reaching 76.6% for DP + FedAvg with two Byzantine clients and no Byzantine resilient aggregation.

We do not report Wilcoxon or Friedman rank tests because the study is not a repeated seed or multi dataset benchmark. Each full training configuration was executed once, and some baseline rows use different evaluation sizes. Applying rank tests to aggregate table entries would create pseudo replication rather than stronger evidence. For the discrete JSON validity endpoint in the directly comparable

Table 9
Main Experimental Results. Full Framework Rows Use 500 Sample Evaluation With All Defenses Active ($\varepsilon = 8$, $\alpha = 0.5$, Sign Flip At $\lambda = 2.0$). Baselines Use 100 Sample Evaluation Except The Centralized Model (Full Test Set, 1,210 Samples). JSON Validity ↑ Is 100.0% Unless Noted. Method abbreviations refer to FedAvg [5], DP-SGD [8], SecAgg [9], Krum [10], and Trimmed Mean [11]. Boldface marks the best value among the comparable full framework rows.

Configuration	Defenses Active	Byz.	ε	BLEU ↑	CmdType% ↑	Action% ↑	Eval n
Centralized baseline	None	0	—	0.8618	99.4	88.4	1,210
FedAvg ($\alpha = 0.5$)	FL only	0	—	0.8595	98.0	84.0	100
FedAvg ($\alpha = 0.1$)	FL only	0	—	0.8642	99.0	86.0	100
DP-FedAvg ($\varepsilon = 1$)	FL + DP	0	1	0.8662	98.0	81.0	100
DP-FedAvg ($\varepsilon = 8$)	FL + DP	0	8	0.8626	98.0	84.0	100
DP + SecAgg + FedAvg (0 Byz)	FL + DP + SecAgg	0	8	**0.8701**	**99.2**	**89.8**	500
DP + SecAgg + Krum (1 Byz)	FL + DP + SecAgg + Byz	1	8	0.8608	97.8	86.2	500
DP + SecAgg + TrimmedMean (1 Byz)	FL + DP + SecAgg + Byz	1	8	0.8543	95.6	86.8	500
DP + SecAgg + Krum (2 Byz)	FL + DP + SecAgg + Byz	2	8	0.8213	87.8	77.0	500
DP + SecAgg + TrimmedMean (2 Byz)	FL + DP + SecAgg + Byz	2	8	0.8022†	80.4	74.4	500
DP + FedAvg (2 Byz, no defense)	FL + DP + SecAgg	2	8	0.6202‡	51.7	50.7	500

†JSON validity 99.2%. ‡JSON validity 76.6%. All other configurations achieve 100% JSON validity.

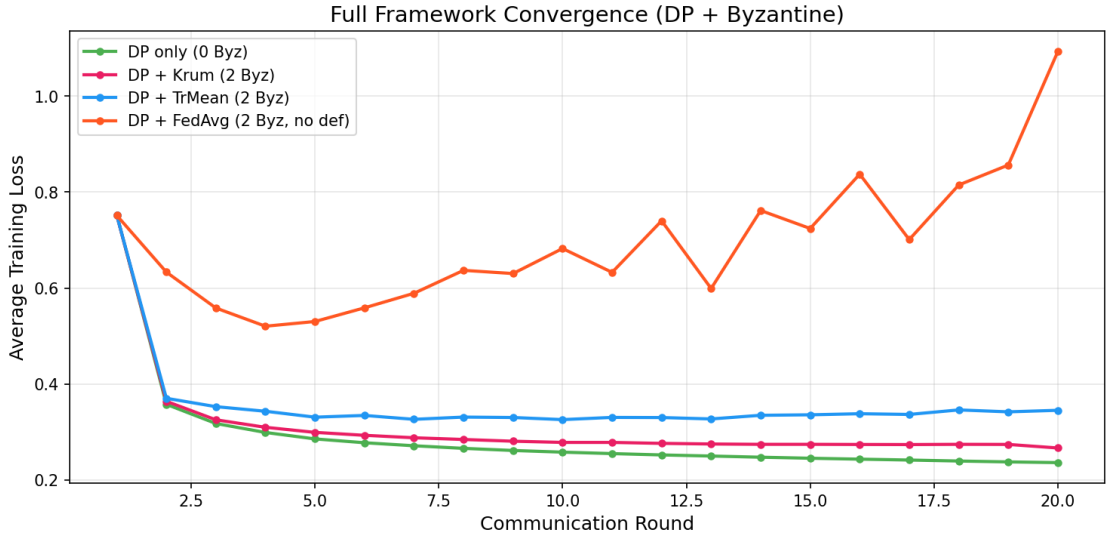

Figure 9: Full Framework Training Loss Over 20 Rounds. The 0 Byzantine DP Baseline Converges Smoothly. Krum And Trimmed Mean With 2 Byzantine Clients Show Minor Instability. Undefended FedAvg Diverges To Loss 1.09 By Round 20.

500 sample full framework rows, however, the effect can be checked from counts. With two Byzantine clients, DP + FedAvg without Byzantine filtering produces 383 valid JSON outputs out of 500, while DP + SecAgg + Krum produces 500 valid outputs out of 500. A two sided Fisher exact test for these counts gives $p = 5.43 \times 10^{-39}$, indicating that the JSON validity collapse is not a small sampling fluctuation. Close BLEU differences in the privacy budget sweep are still treated descriptively, since those runs were not repeated across independent random seeds.

The ablation shown by the main results table and the full framework ablation figure should be read as a security utility ablation rather than a search for higher BLEU alone. Federated LoRA is the first useful layer, since FedAvg at $\alpha = 0.5$ reaches BLEU 0.8595, close to the centralized baseline of 0.8618, while keeping raw mission data local. Adding DP keeps the score essentially unchanged, with DP-FedAvg at $\varepsilon = 8$ reaching BLEU 0.8626, so the privacy layer gives a formal leakage bound without a visible utility penalty in this parameter regime. Adding secure aggregation does not aim to improve BLEU, but the DP + SecAgg + FedAvg row still reaches BLEU 0.8701 with 100% JSON validity, which suggests that masking individual updates does not disturb training when the masks cancel correctly. The largest performance benefit appears when Byzantine resilient aggregation is activated under attack. With two Byzantine clients, DP + SecAgg + Krum reaches BLEU 0.8213 and 100% JSON validity, whereas the corresponding FedAvg configuration without Byzantine filtering falls to BLEU 0.6202 and 76.6% JSON validity. Taken together, the ablation indicates that the proposed layers add privacy, update confidentiality, and poisoning tolerance while retaining near baseline command generation quality whenever the attack level remains within the tested small fleet setting.

We organize the analysis around five observations that emerge from Table 9 and the accompanying figures.

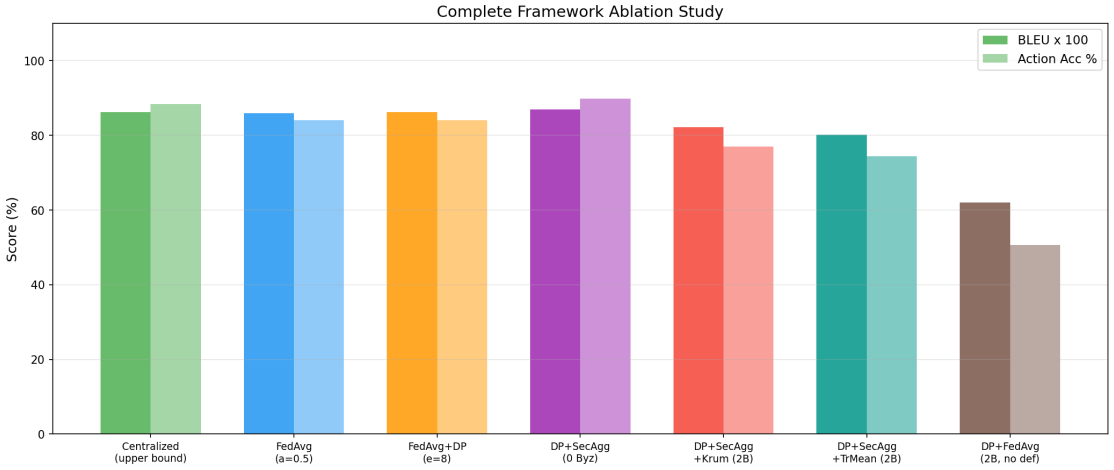

Figure 10: Defense Layer Ablation Showing BLEU (×100) And Action Accuracy. Performance Remains Stable As Privacy Layers Are Added But Degrades Under Byzantine Corruption. The Undefended FedAvg Configuration Confirms Collapse Without Byzantine Resilient Aggregation.

The privacy cost is small. With all layers active and no Byzantine adversary (DP + SecAgg + FedAvg, 0 Byz), the model achieves BLEU 0.8701 on the 500 sample evaluation, with 100% JSON validity, 99.2% command type accuracy, and 89.8% action accuracy. This slightly exceeds the centralized baseline of 0.8618, though the difference is likely an artifact of evaluation sample size (500 vs 1,210) and random seed variation rather than a genuine improvement from DP. The important point is that four layers of defense impose no measurable cost on model quality in this configuration.

One Byzantine attacker is well tolerated. Under a single sign flip attacker ($\lambda = 2.0$), the Krum based configuration achieves BLEU 0.8608 with 100% JSON validity and 97.8% command type accuracy. Trimmed mean reaches BLEU 0.8543 under the same conditions. Both retain full operational viability. The gap between the two is small, with Krum producing slightly higher BLEU and trimmed mean showing marginally better action accuracy (86.8% vs. 86.2%).

Two Byzantine attackers cause meaningful but bounded degradation. At 40% corruption (2 of 5 clients compromised), Krum maintains 100% JSON validity but BLEU drops to 0.8213 and command type accuracy falls to 87.8%. Trimmed mean suffers more, with BLEU at 0.8022, JSON validity dropping to 99.2%, and command type accuracy at 80.4%. The convergence behavior in Fig. 9 confirms that the 2 Byzantine trimmed mean configuration never fully stabilizes, with oscillating training loss in later rounds. Krum performs better here because it selects one complete honest update rather than averaging per coordinate values from a pool that is 40% poisoned. With $n = 5$ and $f = 2$, trimmed mean retains only a single value per dimension after discarding both extremes, losing the statistical averaging that gives it an advantage in less adversarial settings.

Without Byzantine defense, DP alone cannot prevent collapse. The starkest result in Table 9 is the final row. DP + FedAvg with 2 Byzantine attackers and no resilient aggregation produces BLEU 0.6202, JSON validity 76.6%, and command type accuracy barely above chance at 51.7%. Fig. 9 tells the story clearly. Training loss for this configuration diverges from 0.52 at round 4 to 1.09 by round 20, while Krum and trimmed mean converge steadily under the identical attack. This confirms that differential privacy, despite providing formal guarantees against inference attacks, offers no protection whatsoever against model poisoning. The two defenses address fundamentally different threat classes and cannot substitute for one another.

DP noise does not interfere with Byzantine detection. Comparing the full framework results (with DP) against the isolated Byzantine experiments in Section 5.6 (without DP), adding per step DP noise does not appear to degrade the effectiveness of Krum or trimmed mean. The per parameter noise magnitude of approximately 10^{-6} is several orders of magnitude below the perturbations introduced by sign flip attacks ($\sim 10^{0}$ to 10^{1}), so honest updates remain clearly distinguishable from malicious ones in the distance computations used by both aggregation rules.

5.8. Key Findings

Four principal findings emerge from the experiments. First, federated LoRA fine tuning with FedAvg closely matches centralized training across all data heterogeneity levels tested, with BLEU differences below 0.006, though convergence speed is significantly slower under severe non IID conditions ($\alpha = 0.1$). Second, the privacy utility tradeoff under per step DP-SGD is negligible in this parameter regime because LoRA's low rank structure and subsampling amplification reduce per parameter noise to the order of 10^{-6}, making the Dirichlet concentration α the dominant quality factor rather than the privacy budget ε. Third, secure aggregation imposes an acceptable overhead of 58.3 s across 20 rounds for a five client fleet with zero additional bandwidth cost, though $O(n^2)$ scaling becomes a bottleneck beyond 15 clients. Fourth, the full framework with DP ($\varepsilon = 8$), secure aggregation, and Krum retains 99.9% of

the centralized BLEU under a single Byzantine sign flip attacker, while FedAvg without Byzantine defense collapses to BLEU 0.6202 and 76.6% JSON validity under the same attack, confirming that privacy and Byzantine resilience address orthogonal threats and must be deployed together.

6. Discussion

This section interprets the experimental findings in context, identifies the limitations of the current study, and considers the practical implications for deploying privacy-preserving LLM driven Autonomous Aerial Vehicles systems.

6.1. Interpretation of Findings

The results in Section 5 tell a story that is partly expected and partly not. We anticipated that layering multiple defenses would degrade model quality relative to the centralized baseline. What we did not expect was how little that degradation would be under realistic threat conditions.

The near invisible privacy utility tradeoff deserves careful interpretation. At $\varepsilon = 1$, a guarantee considered strong even by conservative standards [41], our model achieves BLEU 0.8662, virtually identical to the non private federated baseline. This outcome may seem too good to be true, and in a sense it is regime specific rather than general. The mechanism behind it is the interaction between LoRA's low rank structure and the subsampled Gaussian mechanism. With only 4.5 million trainable parameters (0.73% of the model), a mini batch sampling rate of $q \approx 0.027$, and RDP composition over 740 steps, the per parameter noise lands in the range of 10^{-6} to 10^{-5}. That is several orders of magnitude below the typical gradient update magnitude. We expect this finding would not hold for full parameter fine tuning, where the noise must be spread across hundreds of millions of parameters without the concentration benefit that low rank adaptation provides. Sun et al. [12] hinted at something similar when they observed that freezing one LoRA matrix reduces DP noise amplification, but our results quantify the effect more directly and under federated non IID conditions.

The dominance of data heterogeneity over DP noise (Section 5.4, Experiment B) connects to an open question raised by Kairouz et al. [30] about the interaction between non IID distributions and differential privacy in federated settings. Our experiments suggest that for LoRA based LLM fine tuning, the Dirichlet concentration α is the primary lever affecting model quality, while ε plays a secondary role at best. This does not mean privacy should be treated as secondary in every deployment. A public safety or defense operator may have a fixed privacy requirement from the start. In that case, the practical message is narrower: once the required privacy budget is fixed, better balancing of local command distributions may be the more productive lever for recovering utility.

The orthogonality between DP and Byzantine resilience is perhaps the cleanest finding of the paper. The collapse of FedAvg to BLEU 0.6202 under two sign flip attackers, even with per step DP noise active, shows unambiguously that privacy and integrity are distinct security properties. DP bounds what an adversary can learn from observing updates. It says nothing about what a compromised client can inject. Conversely, Krum and trimmed mean filter malicious updates based on geometric proximity but offer no inference protection. The fact that per step DP noise at $\sim 10^{-6}$ per parameter does not interfere with Krum's distance based selection (which operates on perturbations at $\sim 10^{0}$ to 10^{1}) confirms that these two defense classes compose cleanly in the LoRA regime, at least at the noise levels we tested. Whether this composability would hold at much smaller ε (larger noise) is an open question we return to in Section 6.2.

This finding also places the proposed framework within a broader shift in secure and resilient FL. Recent systems such as BVDFed [13], RSAM [21], DRL-PBFL [22], and FedDProc [14] no longer treat privacy, aggregation correctness, and malicious update filtering as separate engineering tasks. They combine ideas such as verifiable aggregation, private median estimation, adaptive update weighting, and communication aware defense. Our result supports the same direction, but in a different operating regime. The object being protected here is not a small perception model but a LoRA adapter for an LLM whose output must remain syntactically valid JSON. The empirical point is modest but useful. In this setting, privacy noise can coexist with Byzantine filtering when the DP perturbation remains far below the attack magnitude.

The choice between Krum and trimmed mean is also context dependent. At 20% corruption (1 of 5 Byzantine clients), both perform comparably. At 40% corruption, Krum outperforms trimmed mean because it selects one complete honest update rather than averaging per coordinate residuals from a heavily corrupted pool. With $n = 5$ and $f = 2$, trimmed mean keeps only a single value per dimension after discarding extremes, effectively eliminating the statistical benefit of averaging. In larger fleets where $n \gg 2f$, trimmed mean would likely recover its advantage.

6.2. Limitations

We identify several limitations that should temper the conclusions drawn from this work.

Simulated federation. All federated experiments run on a single GPU simulating five clients sequentially rather than on physically distributed autonomous aerial vehicle hardware with real wireless links. Communication latency, packet loss, and bandwidth constraints in actual air to ground channels are not modeled. The secure aggregation overhead of 2.9 s per round was measured on CPU and would differ on embedded autonomous aerial vehicle processors. A field deployment would need to account for these factors.

Synthetic dataset. The 14,446 sample dataset was generated via the GPT-4o-mini API rather than collected from real autonomous aerial vehicle operations. While we enforced realistic parameter ranges and validated all samples for JSON correctness and schema adherence, synthetic data cannot fully capture the distribution of commands that arise

in operational settings. The language patterns, the frequency of ambiguous instructions, and the edge cases in real missions may differ substantially from what an LLM generates when prompted to simulate them.

Small fleet size. Five clients is sufficient for demonstrating the framework and stress testing at 20% and 40% Byzantine corruption, but real autonomous aerial vehicle fleets may involve tens or hundreds of drones. The $O(n^2)$ scaling of our secure aggregation protocol becomes a practical bottleneck beyond roughly 15 to 20 clients, as the benchmarks in Section 5.5 indicate. Poly logarithmic protocols such as those proposed by Bell et al. [33] would be needed for larger deployments.

Fixed attack strength. We tested sign flip at $\lambda = 2.0$, random noise at $\sigma_{attack} = 5.0$, and targeted poisoning with label swaps. Adaptive adversaries that observe the aggregation rule and tailor their attack accordingly, for example, the inner product manipulation strategy of Xie et al. [42], were not evaluated. It is plausible that more sophisticated attacks could evade Krum or trimmed mean under DP noise, particularly if the noise level were higher (smaller ε).

Single model architecture. All experiments use the 1.1B parameter TinyLlama model with a fixed LoRA configuration ($r = 16$). The near zero DP cost we observe is partly a consequence of this particular parameter count and rank. Larger models (7B, 13B) or higher LoRA ranks would increase the trainable parameter space, potentially shifting the noise to signal balance. Whether the composability between DP and Byzantine defenses holds at those scales remains to be tested.

BLEU as evaluation metric. While BLEU captures structural similarity between predicted and reference JSON strings, it does not evaluate the semantic correctness of parameter values. A command with a latitude off by 0.01 degrees would score nearly perfectly on BLEU but send the drone to the wrong location. Our command type and action accuracy metrics partially address this, but a full evaluation of parameter level precision would strengthen future work.

6.3. Practical Implications

Despite the limitations above, several practical takeaways emerge for organizations considering privacy preserving LLM fine tuning for autonomous aerial vehicle fleets.

First, LoRA based federated fine tuning is ready for deployment in bandwidth constrained autonomous aerial vehicle settings. The 18 MB per round communication cost is manageable over LTE or dedicated radio links, and the utility loss relative to centralized training is negligible across all heterogeneity levels we tested. Organizations can train fleet wide command generation models without centralizing sensitive operational data.

Second, strong DP guarantees ($\varepsilon \leq 1$) can be achieved essentially for free in the LoRA regime. This is a useful result for regulatory compliance, particularly in jurisdictions where data protection laws apply to drone surveillance operations. The cost of privacy is not in model quality but rather in the engineering complexity of correctly implementing per step noise injection with proper RDP accounting.

Third, Byzantine resilient aggregation should be treated as a non optional defense layer for any Autonomous Aerial Vehicles fleet operating in contested or adversarial environments. Our results show that a single compromised drone running a sign flip attack can degrade an undefended model's JSON validity to 76.6%, which in operational terms means roughly one in four commands would fail to parse. Krum is the safer default for small fleets ($n \leq 10$) because it preserves internal update consistency even under high corruption ratios.

Fourth, the composable architecture of the framework allows operators to activate or deactivate individual layers based on the threat environment. A fleet operating in a trusted environment might run only FedAvg with secure aggregation. A fleet in contested airspace would add DP and Byzantine resilience. The modular design means each layer imposes its own cost independently, and the costs are additive rather than multiplicative.

6.4. Data and Code Availability

The author created UAV command generation dataset used in this study is publicly available on Kaggle as the UAV Command Generation: Natural Language to Drone dataset [38]. The experiment materials are also publicly accessible in a GitHub repository at https://github.com/csshafiqahmed/Privacy-Preserving-Federated-Fine-Tuning-of-Large-Language-Models-for-UAVs-Command-Generation. The repository contains six experiment notebooks covering the centralized baseline, federated baseline, DP-FedAvg, secure aggregation, Byzantine aggregation, and full framework evaluation, along with the dataset generation notebook.

7. Conclusion

This manuscript evaluated a privacy preserving federated fine tuning framework for large language model driven Autonomous Aerial Vehicles command generation. The framework combines LoRA based FedAvg, per step DP-SGD with Rényi accounting, ECC secure aggregation with forward secrecy, and Byzantine resilient aggregation. On a 14,446 sample dataset with five simulated clients, the full configuration at $\varepsilon = 8$ retained near centralized quality, reaching BLEU 0.8608 versus 0.8618 for the centralized baseline and 100% JSON validity under a Byzantine sign flip attacker. FedAvg without Byzantine defense fell to BLEU 0.6202 and 76.6% JSON validity, confirming that privacy and integrity controls solve different problems. The experiments also suggest that LoRA and subsampling make per step DP noise small in this setting, while Dirichlet heterogeneity remains the main utility driver. Secure aggregation added 58.3 seconds over 20 rounds for five clients without extra bandwidth. Several limits remain. Scaling to larger fleets will require replacing the $O(n^2)$ pairwise masking protocol with poly logarithmic alternatives. In practical terms, the current pairwise masking design should

be treated as a small fleet solution, since Section 5.5 shows that overhead becomes problematic beyond roughly 15 clients. Future work should test adaptive Byzantine attacks, larger models and LoRA ranks, and deployment on physical Autonomous Aerial Vehicles links where latency and bandwidth constraints are less ideal than simulation.

References

[1] MarketsandMarkets, Unmanned Aerial Vehicle (UAV) Market – Global Forecast to 2030, Technical Report, MarketsandMarkets Research Pvt. Ltd., 2025. URL: https://www.marketsandmarkets.com/Market-Reports/unmanned-aerial-vehicles-uav-market-662.html, accessed: March 2026.

[2] M. L. Tazir, M. Mancas, T. Dutoit, From words to flight: Integrating OpenAI ChatGPT with PX4/Gazebo for natural language-based drone control, in: Proceedings of 2023 the 13th International Workshop on Computer Science and Engineering (WCSE 2023), WCSE, 2023, pp. 215–222. doi:10.18178/wcse.2023.06.031.

[3] G. Chen, X. Yu, N. Ling, L. Zhong, TypeFly: Flying drones with large language model, 2024. URL: https://arxiv.org/abs/2312.14950. arXiv:2312.14950.

[4] S. Javaid, H. Fahim, B. He, N. Saeed, Large language models for UAVs: Current state and pathways to the future, IEEE Open Journal of Vehicular Technology 5 (2024) 1166–1192.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[6] L. Zhu, Z. Liu, S. Han, Deep leakage from gradients, Advances in Neural Information Processing Systems 32 (2019).

[7] Y. Gao, Y. Xie, H. Deng, Z. Zhu, Gradient inversion attack in federated learning: Exposing text data through discrete optimization, in: Proceedings of the 31st International Conference on Computational Linguistics, 2025, pp. 2582–2591.

[8] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 308–318. doi:10.1145/2976749.2978318.

[9] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for privacy-preserving machine learning, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1175–1191. doi:10.1145/3133956.3133982.

[10] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries: Byzantine tolerant gradient descent, Advances in Neural Information Processing Systems 30 (2017).

[11] D. Yin, Y. Chen, K. Ramchandran, P. Bartlett, Byzantine-robust distributed learning: Towards optimal statistical rates, in: International Conference on Machine Learning, PMLR, 2018, pp. 5650–5659.

[12] Y. Sun, Z. Li, Y. Li, B. Ding, Improving LoRA in privacy-preserving federated learning, 2024. URL: https://arxiv.org/abs/2403.12313. arXiv:2403.12313.

[13] X. Gao, S. Fu, L. Liu, Y. Luo, BVDFed: Byzantine-resilient and verifiable aggregation for differentially private federated learning, Frontiers of Computer Science 18 (2024) 185810.

[14] Y. Xia, T. Jahani-Nezhad, R. Bitar, Beyond trade-offs: A unified framework for privacy, robustness, and communication efficiency in federated learning, 2025. URL: https://arxiv.org/abs/2508.12978. arXiv:2508.12978.

[15] J. W. Ayana, Q. Huang, PrivLLMSwarm: Privacy-preserving LLM-driven UAV swarms for secure IoT surveillance, arXiv preprint arXiv:2512.06747 (2025).

[16] P. Zhang, G. Zeng, T. Wang, W. Lu, TinyLlama: An open-source small language model, 2024. URL: https://arxiv.org/abs/2401.02385. arXiv:2401.02385.

[17] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: International Conference on Learning Representations (ICLR), 2022. URL: https://openreview.net/forum?id=nZeVKeeFYf9.

[18] J. Bai, D. Chen, B. Qian, L. Yao, Y. Li, Federated fine-tuning of large language models under heterogeneous tasks and client resources, Advances in Neural Information Processing Systems 37 (2024) 14457–14483.

[19] Z. Xiang, T. Wang, W. Lin, D. Wang, Practical differentially private and Byzantine-resilient federated learning, Proceedings of the ACM on Management of Data 1 (2023) 1–26.

[20] X. Feng, H. Liu, H. Yang, Q. Xie, L. Wang, Batch-Aggregate: Efficient aggregation for private federated learning in VANETs, IEEE Transactions on Dependable and Secure Computing 21 (2024) 4939–4952.

[21] Y. He, P. Li, J. Ni, X. Deng, H. Lu, J. Zhang, L. T. Yang, RSAM: Byzantine-robust and secure model aggregation in federated learning for Internet of Vehicles using private approximate median, IEEE Transactions on Vehicular Technology 73 (2024) 6714–6726.

[22] Y. Pan, Z. Su, Y. Wang, J. Zhou, M. Mahmoud, Privacy-preserving Byzantine-robust federated learning via deep reinforcement learning in vehicular networks, IEEE Transactions on Vehicular Technology 74 (2025) 9461–9475.

[23] T. Ahmad, A. A. H. Elnour, M. U. Hadi, V. Vassiliou, L. Dimitriou, X. J. Li, D. Trihinas, T. R. Gadekallu, AeroVeil-FL: Privacy-preserving federated learning framework for UAV-assisted real-time disaster detection, Computers and Electrical Engineering 129 (2026) 110853.

[24] S. Deng, J. Zhang, L. Tao, X. Jiang, F. Wang, LSBlocFL: A secure federated learning model combining blockchain and lightweight cryptographic solutions, Computers and Electrical Engineering 111 (2023) 108986.

[25] Z. Wang, X. Yu, H. Wang, P. Xue, A federated learning scheme for hierarchical protection and multiple aggregation, Computers and Electrical Engineering 117 (2024) 109240.

[26] J. Zhang, S. Vahidian, M. Kuo, C. Li, R. Zhang, T. Yu, G. Wang, Y. Chen, Towards building the Federated GPT: Federated instruction tuning, in: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2024, pp. 6915–6919. doi:10.1109/ICASSP48485.2024.10447454.

[27] Y. Yang, G. Long, Q. Lu, L. Zhu, J. Jiang, C. Zhang, Federated low-rank adaptation for foundation models: A survey, in: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, 2025, pp. 10779–10787. doi:10.24963/ijcai.2025/1196.

[28] I. Mironov, Rényi differential privacy, in: 2017 IEEE 30th Computer Security Foundations Symposium (CSF), IEEE, 2017, pp. 263–275. doi:10.1109/CSF.2017.11.

[29] H. B. McMahan, D. Ramage, K. Talwar, L. Zhang, Learning differentially private recurrent language models, 2018. URL: https://arxiv.org/abs/1710.06963. arXiv:1710.06963.

[30] P. Kairouz, H. B. McMahan, et al., Advances and open problems in federated learning, Foundations and Trends in Machine Learning 14 (2021) 1–210.

[31] Y. Wang, Y. Lin, X. Zeng, G. Zhang, PrivateLoRA for efficient privacy preserving LLM, 2023. URL: https://arxiv.org/abs/2311.14030. arXiv:2311.14030.

[32] S. Zhang, J. Huang, P. Li, C. Liang, Differentially private federated learning with local momentum updates and gradients filtering, Information Sciences 680 (2024) 120960.

[33] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, M. Raykova, Secure single-server aggregation with (poly)logarithmic overhead, in: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 1253–1269. doi:10.1145/3372297.3417885.

[34] Y. Ma, J. Woods, S. Angel, A. Polychroniadou, T. Rabin, Flamingo: Multi-round single-server secure aggregation with applications to private federated learning, in: 2023 IEEE Symposium on Security

and Privacy (SP), IEEE, 2023, pp. 477–496. doi:10.1109/SP46215.2023.10179434.

[35] M. A. Ferrag, A. Lakas, M. Debbah, UAVBench: An open benchmark dataset for autonomous and agentic AI UAV systems via LLM-generated flight scenarios, 2025. URL: https://arxiv.org/abs/2511.11252. arXiv:2511.11252.

[36] T.-M. H. Hsu, H. Qi, M. Brown, Measuring the effects of non-identical data distribution for federated visual classification, 2019. URL: https://arxiv.org/abs/1909.06335. arXiv:1909.06335.

[37] D. Dolev, A. C.-C. Yao, On the security of public key protocols, IEEE Transactions on Information Theory 29 (1983) 198–208.

[38] S. Ahmed, UAV Command Generation: Natural Language to Drone, Kaggle dataset, 2026. URL: https://www.kaggle.com/datasets/csshafiqahmed/uav-llm-command-dataset, accessed: 27 April 2026.

[39] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, QLoRA: Efficient finetuning of quantized LLMs, Advances in Neural Information Processing Systems 36 (2023) 10088–10115.

[40] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002, pp. 311–318. doi:10.3115/1073083.1073135.

[41] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Foundations and Trends® in Theoretical Computer Science 9 (2014) 211–487.

[42] C. Xie, O. Koyejo, I. Gupta, Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation, in: Uncertainty in Artificial Intelligence, PMLR, 2020, pp. 261–270.