Aligned LoRA Updates via Intrinsic Geometry for Federated Low-Rank Adaptation

Yihao Yang[1,2,†], **Jian Liang**[1,†], **Wenke Huang**[3], **He Li**[1], **Xingyu Gao**[4,5,*], **Mang Ye**[1,2,*]
[1]School of Computer Science, Wuhan University
[2]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)
[3]College of Computing and Data Science, Nanyang Technological University
[4]Institute of Microelectronics, Chinese Academy of Sciences
[5]University of Chinese Academy of Sciences
{yihaoyang, yemang}@whu.edu.cn {gxy9910}@gmail.com

Abstract

Federated Low-Rank Adaptation (FedLoRA) enables efficient fine-tuning of large language models by exchanging low-rank adaptation parameters across clients. Several studies mainly focus on aggregation bias and design improved aggregation schemes. We argue that aggregation bias is not the fundamental bottleneck of FedLoRA. Even under ideal aggregation, heterogeneous data induce directional conflicts among client-specific LoRA updates. We show that these conflicts arise from misaligned low-rank adaptation subspaces, leading to destructive interference during collaboration. To diagnose this phenomenon, we introduce a geometry-aware metric that characterizes client updates at the subspace level. Our analysis reveals that subspace conflicts intensify with increasing heterogeneity and are concentrated in the column spaces of LoRA updates. Based on this insight, we propose ALIGN, a geometry-aware federated adaptation framework that aligns client LoRA subspaces during aggregation and constrains local updates to mitigate conflicts. Comprehensive empirical results confirm that mitigating subspace misalignment is crucial for FedLoRA, and that ALIGN consistently improves downstream performance over existing methods. The codes are released at https://github.com/MoratalYang/ALIGN.

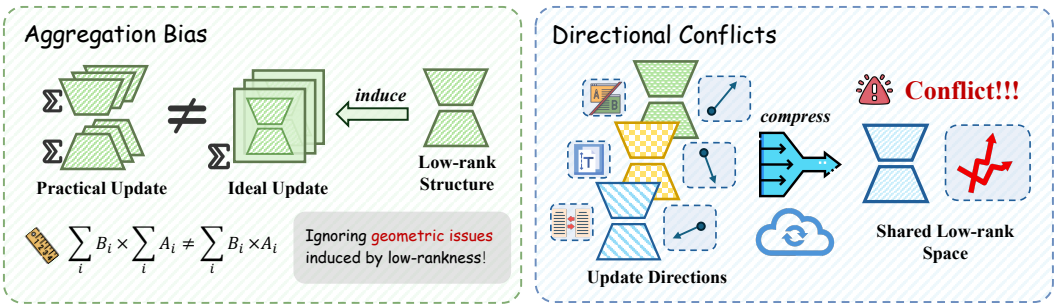
1 Introduction

Federated Learning (FL) is a distributed paradigm [29, 17, 18, 30, 14, 13], enabling collaborative model training across decentralized clients without exchanging private data. Large Language Models (LLMs) [31, 35, 6, 7] have demonstrated strong representation and generalization capabilities across a wide range of downstream tasks by leveraging rich prior knowledge. However, deploying such models in federated settings remains challenging due to their massive parameter scale, resulting in prohibitive communication as well as computation costs for full fine-tuning. Meanwhile, Low-Rank Adaptation (LoRA) [12, 26, 19, 46] has emerged as a parameter-efficient fine-tuning paradigm that adapts large models by learning low-rank update matrices while keeping the pre-trained weights frozen. Thus, Federated LoRA (FedLoRA) [42, 45, 41, 43] offers a communication-efficient and privacy-preserving framework for collaboratively fine-tuning large models in federated environments, where only low-rank adaptation parameters are exchanged among clients.

Recent studies on FedLoRA have investigated aggregation challenges introduced by the low-rank structure of LoRA. A line of work focuses on aggregation bias [32], which arises from aggregating

[†] Equal Contribution.
[*] Corresponding Author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Existing FedLoRA (b) Our Focus

Figure 1: **Low-Rank Structural Challenges in FedLoRA.** (a) Existing FedLoRA methods mainly focus on aggregation bias, caused by aggregating factorized low-rank updates instead of full-parameter updates. (b) In contrast, we highlight directional conflicts as a more fundamental issue. When diverse client-specific low-rank updates are compressed into a shared low-rank space, incompatible directions interfere with each other, limiting the effectiveness of aggregation.

client updates in a factorized low-rank form. Aggregating factorized low-rank components across clients is generally not equivalent to aggregating the corresponding full-parameter updates, which can lead to a global update that deviates from the ideal update in the full parameter space. To address this issue, a variety of strategies have been proposed, including alternative aggregation schemes [36, 3], selective parameter sharing [34, 9] and alternating the training of modules [23, 5]. Despite their differences, these approaches largely view aggregation bias as an error source to be mitigated in order to better approximate a presumed ideal global update.

However, as shown in Fig. 1, these studies overlooked another geometric challenge inherently introduced by the low-rank structure. Specifically, in FedLoRA, each client learns its own low-rank adaptation based on local data, resulting in LoRA updates whose dominant directions reflect client-specific optimization trajectories. During aggregation, these independently learned LoRA modules are required to be merged and compressed into a shared low-rank space [48]. However, due to the limited representational capacity of this space, low-rank directions that are individually meaningful on different clients may become misaligned or incompatible when projected together. This can give rise to **Directional Conflicts**, where competing directions interfere with each other and weaken the effectiveness of the aggregated update. Notably, this issue arises from the geometric interaction among diverse low-rank subspaces and can persist even when aggregation accurately preserves the intended update form. Therefore, beyond aggregation bias, directional conflicts induced by low-rank compression represent an additional and previously underexplored bottleneck in FedLoRA training.

Motivated by this insight, we introduce a geometry-aware conflict metric that characterizes client updates at the subspace level. Rather than comparing client updates as raw parameter vectors, the diagnostic characterizes each client update by the low-dimensional subspace that captures its dominant functional variation. Based on this metric, we further conduct a systematic analysis of its effect on federated training. Empirically, we reveal that as data heterogeneity [47, 16, 21] increases, update directions across different subspaces exhibit increasingly severe conflicts, and notably, such conflicts predominantly reside in the column space of the low-rank adaptations. Therefore, this raises two challenges: **I) How to coordinate diverse LoRA directions across clients during aggregation?** Simply averaging or faithfully preserving client updates does not prevent destructive interference when their underlying subspaces are misaligned, especially under strong data heterogeneity. **II) How to properly constrain local LoRA updates to learn a shared space?** Without appropriate constraints at the client level, conflicting subspace directions may continue to emerge, placing an excessive burden on server-side aggregation to resolve incompatibilities. A more detailed discussion of these phenomena is provided in Sec. 4.

To address these challenges, we propose **A**ligned **L**oRA updates via **I**ntrinsic **G**eometry under data heteroge**N**eity (ALIGN), a geometry-aware federated framework for low-rank adaptation. ALIGN views the LoRA update of each client not as raw parameters, but as a low-dimensional subspace that captures its dominant adaptation direction. This representation enables invariant comparison across clients and provides a principled basis for reasoning about directional compatibility under heterogeneity. Building on this perspective, ALIGN addresses directional conflicts from both the server and client sides. To tackle **I**), ALIGN introduces a Geometry-Aware Aggregation strategy that explicitly

decomposes client updates into components aligned with a shared consensus subspace and orthogonal residuals. The aligned components preserve consensus directions, while the orthogonal residuals are robustly absorbed via geometry-aware fusion to suppress destructive interference. Meanwhile, to address **II**), ALIGN further incorporates Direction Alignment Regularization into local training, softly biasing client updates toward the shared geometry to prevent highly conflicting subspaces from emerging during local optimization. Through this joint client–server design, ALIGN resolves directional conflicts at their source rather than correcting them post hoc, enabling more stable and effective collaborative training in heterogeneous data settings. To validate our approach, we conduct comprehensive experiments across multiple architectures, demonstrating superior performance in mitigating directional conflicts. Our contributions are summarized as follows:

❶ ***Revealing Directional Conflicts.*** We identify directional conflicts in Federated LoRA, where independently learned low-rank updates become geometrically incompatible under data heterogeneity, leading to destructive interference beyond aggregation bias. To address this issue, we adopt a subspace-level geometric perspective to characterize and analyze such conflicts.

❷ ***Geometry-Aware FedLoRA Framework.*** We propose ALIGN, a geometry-aware federated adaptation framework that coordinates client updates at the subspace level and constrains local LoRA adaptation to mitigate directional conflicts during aggregation.

❸ ***Comprehensive Empirical Validation.*** Extensive experiments across multiple architectures demonstrate that ALIGN achieves the best average performance and remains competitive across tasks.

2 Related Work

2.1 Parameter-Efficient Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) methods [38, 28] aim to adapt pretrained models using a small number of additional parameters. Representative approaches include adapters, prompt tuning [20, 49, 33], prefix tuning [22], and Low-Rank Adaptation (LoRA) [12, 25, 24]. Among these methods, LoRA has gained particular popularity due to its simplicity and strong empirical performance. It represents weight updates using low-rank matrices while keeping the pretrained weights frozen, thereby substantially reducing both memory overhead and communication cost. LoRA and its variants [26, 19, 46] have been successfully applied to large language models across diverse tasks, and have become a standard tool for efficient adaptation. These characteristics make LoRA particularly well-suited for federated learning with large language models.

2.2 Federated Fine-Tuning with LoRA

Building on the efficiency of LoRA, Federated LoRA (FedLoRA) [2, 43, 11] has emerged as a natural paradigm for enabling parameter-efficient fine-tuning [44] in federated learning. FedIT [45] incorporates LoRA into the federated learning pipeline, allowing clients to collaboratively fine-tune models while keeping raw data local. However, recent work [36] has shown that naively aggregating factorized low-rank updates can introduce aggregation bias, since averaging low-rank factors is generally not equivalent to averaging the corresponding full-rank updates. To address this issue, a variety of strategies have been proposed, including alternative aggregation schemes [36, 3], selective parameter sharing [34, 9], and alternating the training of modules [23, 5]. These methods primarily aim to reduce the discrepancy between aggregated low-rank updates and an idealized full-parameter global update. Beyond aggregation bias, we focus on the directional conflicts arising from low-rank compression and examine how they hinder effective FedLoRA training.

3 Problem Setup

Low-Rank Adaptation. LoRA is a parameter-efficient fine-tuning method that enables effective adaptation of large pre-trained models with a small number of trainable parameters. Consider a linear transformation parameterized by a weight matrix $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. Instead of directly updating $\mathbf{W}$, LoRA introduces an additive low-rank update:

$$\mathbf{W}' = \mathbf{W} + \Delta\mathbf{W}, \quad \Delta\mathbf{W} = \mathbf{B}\mathbf{A}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{r \times d_{\text{in}}}$ and $\mathbf{B} \in \mathbb{R}^{d_{\text{out}} \times r}$ with rank $r \ll \min(d_{\text{out}}, d_{\text{in}})$. The pre-trained weight $\mathbf{W}$ is kept frozen, and only the low-rank matrices $\mathbf{A}$ and $\mathbf{B}$ are optimized during training.

Federated LoRA. We consider a federated learning setting with K clients, where each client $k \in \{1, \ldots, K\}$ holds a local dataset $\mathcal{D}_k$ drawn from a potentially heterogeneous data distribution. In FedLoRA, the backbone model parameters are shared and frozen across all clients, while adaptation is performed by learning LoRA parameters. At each communication round, the server maintains global LoRA parameters $(\mathbf{A}_g, \mathbf{B}_g)$ and broadcasts them to participating clients. Each client k performs local optimization on its dataset $\mathcal{D}_k$ by minimizing the local objective:

$$\min_{\mathbf{A},\mathbf{B}} \ \mathcal{L}_k\big(\mathbf{W} + \mathbf{B}\mathbf{A}\big), \tag{2}$$

initialized from the global parameters $(\mathbf{A}_g, \mathbf{B}_g)$. After local training, client k obtains updated LoRA parameters $(\mathbf{A}_k, \mathbf{B}_k)$. The server then aggregates client updates to form new global parameters, commonly via weighted averaging using the normalized proportions of client data:

$$\mathbf{A}_g = \sum_{k=1}^{K} p_k \mathbf{A}_k, \quad \mathbf{B}_g = \sum_{k=1}^{K} p_k \mathbf{B}_k, \tag{3}$$

where p_k denotes the relative weight of client k, typically proportional to $|\mathcal{D}_k|$.

4 Subspace Disagreement in FedLoRA

4.1 Quantifying LoRA Directional Disagreement

In FedLoRA, different clients may adapt the shared model along distinct low-dimensional directions. Rather than comparing raw LoRA update matrices in Euclidean space, we aim to quantify whether different clients learn aligned dominant update directions. This motivates a geometry-aware measure of inter-client heterogeneity, defined on the low-rank subspaces induced by client updates.

LoRA Update Subspaces. At each communication round, the LoRA update of client k is represented by a matrix $\Delta\mathbf{W}_k \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. We characterize its dominant behavior by the rank-r output subspace spanned by its leading left singular vectors:

$$\Delta\mathbf{W}_k \approx \mathbf{U}_k \Sigma_k \mathbf{V}_k^\top, \quad \mathbf{U}_k \in \mathbb{R}^{d_{\text{out}} \times r}, \quad \mathbf{U}_k^\top \mathbf{U}_k = I_r, \tag{4}$$

which defines a point on the Grassmann manifold $\mathrm{Gr}(r, d_{\text{out}})$ [8]. For clarity, we present the formulation using output subspaces. An entirely analogous construction applies to the input subspaces spanned by the right singular vectors $\mathbf{V}_k$.

Extrinsic Fréchet Mean under the Projection Metric. Given normalized client weights $\{p_k\}_{k=1}^K$, we measure subspace disagreement using the projection metric [10, 15], which quantifies the misalignment between two subspaces as $r - \|\mathbf{U}_k^\top \mathbf{U}\|_F^2$. This metric is invariant to rotations within each subspace and depends only on their relative orientations. Under this metric, we define a consensus subspace $\mathbf{U}_\star \in \mathrm{Gr}(r, d_{\text{out}})$ as the weighted Fréchet mean of the client update subspaces:

$$\mathbf{U}_\star = \arg\min_{\mathbf{U} \in \mathrm{Gr}(r, d_{\text{out}})} \sum_{k=1}^{K} p_k \big(r - \|\mathbf{U}_k^\top \mathbf{U}\|_F^2\big). \tag{5}$$

Although defined on the Grassmann manifold, the projection-based objective admits a closed-form solution [4]. Representing each client subspace by its orthogonal projection matrix $\mathbf{U}_k\mathbf{U}_k^\top$, the optimization reduces to finding the top-r eigenspace of their weighted average. Since this average equals the product of the weighted concatenation matrix and its transpose, the consensus subspace $\mathbf{U}_\star$ can be computed from the top-r left singular vectors of:

$$\big[\sqrt{p_1}\mathbf{U}_1, \ \ldots, \ \sqrt{p_K}\mathbf{U}_K\big]. \tag{6}$$

A formal derivation based on a Ky Fan maximum principle is provided in Appendix D.

Subspace Disagreement Definition. We then define the corresponding variance as:

$$\mathrm{FV} = \sum_{k=1}^{K} p_k \big(r - \|\mathbf{U}_k^\top \mathbf{U}_\star\|_F^2\big), \tag{7}$$

which we refer to as the Grassmann Fréchet Variance. Intuitively, this quantity measures how much, on average, the dominant update subspace of each client deviates from the consensus subspace. Under the projection metric, the variance reduces to a rotation-invariant measure of subspace misalignment and admits a simple closed-form computation.

4.2 Empirical Analysis

As shown in Fig. 2, we empirically analyze subspace-level disagreement in FedLoRA by measuring Fréchet Variance (Eq. (7)) of LoRA updates across layers. We report Fréchet Variance separately for the output subspaces spanned by the left singular vectors $\mathbf{U}$ and the input subspaces spanned by the right singular vectors $\mathbf{V}$, under both homogeneous and heterogeneous data distributions. Blue lines correspond to heterogeneous data, while red lines denote homogeneous data.

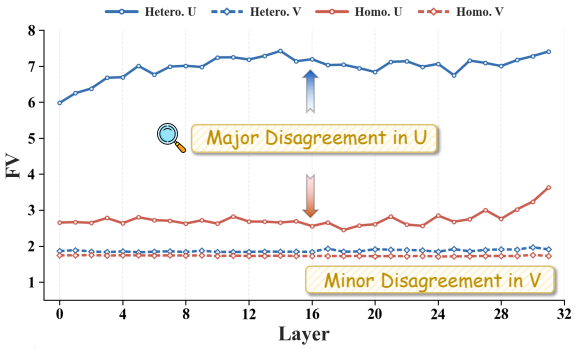


Figure 2: **Fréchet Variance across Layers.** See Sec. 4.

A first observation is that data heterogeneity substantially amplifies subspace disagreement. Under heterogeneous data, the FV of client update subspaces is consistently higher than that observed under homogeneous data across almost all layers. This confirms that client-specific data distributions induce divergent optimization trajectories that manifest as misaligned low-rank subspaces.

More importantly, the disagreement is highly asymmetric between the two sides of the LoRA decomposition. Under heterogeneous data, the Fréchet variance associated with the output subspaces $\mathbf{U}$ is dramatically larger than that of the input subspaces $\mathbf{V}$, often by a wide margin. In contrast, under homogeneous data, both $\mathbf{U}$ and $\mathbf{V}$ subspaces exhibit relatively low and comparable variance. This indicates that the dominant source of directional conflicts in FedLoRA primarily resides in the column space of the low-rank updates, rather than the row space.

These results suggest that FedLoRA suffers from a fundamentally geometric issue: incompatible low-rank subspaces learned by different clients interfere when compressed into a shared space, particularly in the column space of LoRA updates. Parameter-level aggregation alone cannot resolve such directional conflicts. This motivates a geometry-aware approach that treats each LoRA update as a low-dimensional subspace rather than as raw parameters.

5 Proposed Method

We propose ALIGN, a geometry-aware federated LoRA framework designed to mitigate directional conflicts among client updates through (i) Geometry-Aware Aggregation on the server and (ii) Direction Alignment Regularization on clients. The architecture illustration is shown in Fig. 3.

5.1 Geometry-Aware Aggregation

Motivation. Under heterogeneous data distributions, naive aggregation of low-rank updates can lead to severe directional interference. Our empirical analysis in Sec. 4 reveals that such interference is highly asymmetric. Specifically, directional disagreement concentrates primarily in the column subspaces of LoRA updates, while row factors remain comparatively stable. This observation suggests that treating all update directions equally during aggregation is suboptimal. From a geometric perspective, since the column subspace of a LoRA update ΔW is contained in $\mathrm{col}(\mathbf{B})$, misalignment among client updates manifests predominantly as inconsistency across the column subspaces of $\mathbf{B}$. Motivated by this insight, we introduce a geometry-aware aggregation strategy that decomposes client updates relative to a shared reference geometry and aggregates them in a geometry-aware manner.

Consensus Subspace. At each round, the server computes a per-layer consensus subspace via Eq. (5):

$$\mathbf{U}_\star \in \mathbb{R}^{d_{\text{out}} \times r},\ \mathbf{U}_\star^\top \mathbf{U}_\star = \mathbf{I}, \tag{8}$$

which represents the shared principal adaptation directions across clients. In practice, $\mathbf{U}_\star$ can be instantiated as the Fréchet mean subspace under the projection metric (Sec. 4), and is communicated

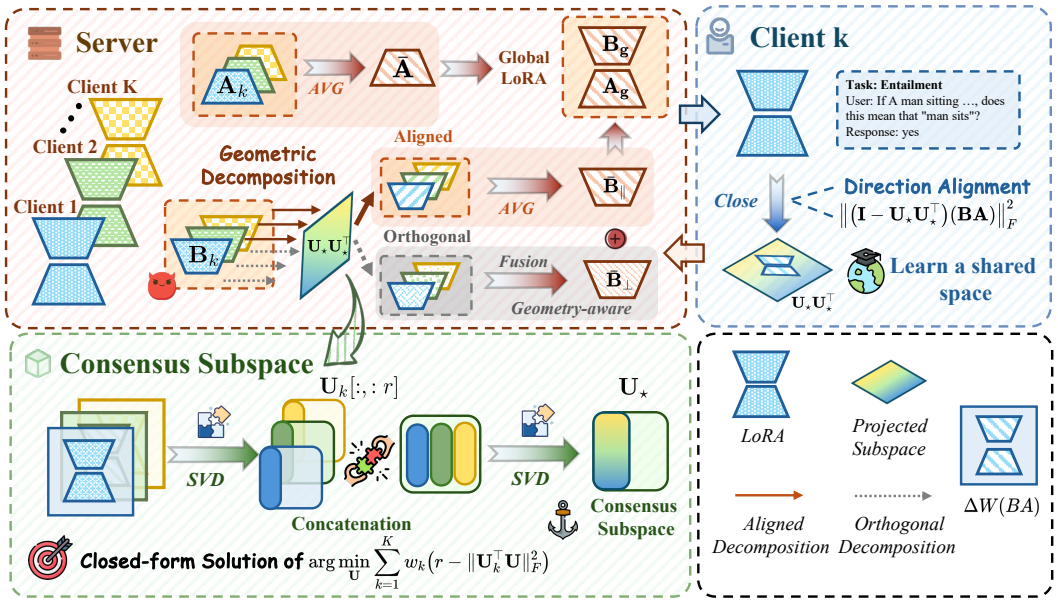


Figure 3: **Architecture Illustration.** The server first computes a consensus subspace from client LoRA updates, and then performs geometry-aware aggregation to combine aligned components and robustly fuse orthogonal residuals. Meanwhile, clients incorporate direction alignment regularization during local training to softly bias updates toward the shared subspace.

to clients as the geometry context for the next local update. For clarity, we focus our presentation on a single LoRA-injected layer, with the multi-layer case handled analogously.

Geometric Decomposition. Given a participating client set, each client uploads its LoRA factors $(\mathbf{A}_k, \mathbf{B}_k)$ to the server. ALIGN then decomposes the uploaded $\mathbf{B}_k$ into the projection onto the consensus subspace and its orthogonal residual:

$$\mathbf{B}_k = \underbrace{\mathbf{U}_\star \mathbf{U}_\star^\top \mathbf{B}_k}_{Aligned\ \mathbf{B}_{k,\parallel}} + \underbrace{\left(\mathbf{I} - \mathbf{U}_\star \mathbf{U}_\star^\top\right)\mathbf{B}_k}_{Orthogonal\ \mathbf{B}_{k,\perp}}. \tag{9}$$

The projected part $\mathbf{B}_{k,\parallel}$ captures update directions compatible with the shared subspace, while $\mathbf{B}_{k,\perp}$ captures client-specific directions that potentially conflict across clients.

Consensus Aggregation for Aligned Components. For the aligned components, we perform standard FedAvg with normalized dataset weights p_k:

$$\bar{\mathbf{A}} = \sum_{k=1}^{K} p_k \mathbf{A}_k,\ \bar{\mathbf{B}}_\parallel = \sum_{k=1}^{K} p_k \mathbf{B}_{k,\parallel}. \tag{10}$$

This preserves the consensus directions and mitigates destructive interference caused by misalignment in the dominant shared space during aggregation.

Geometry-Aware Fusion for Orthogonal Residuals. Directly discarding the orthogonal components $\mathbf{B}_{k,\perp}$ may eliminate informative client-specific adaptation signals, while naively averaging them can amplify conflicting directions under heterogeneity. An effective aggregation rule should therefore attenuate highly misaligned residuals while still permitting moderately deviating but informative updates to contribute. To this end, we aggregate orthogonal residuals using a geometry-aware fusion scheme. Specifically, for each client k, we quantify the deviation of its update subspace from the global reference $\mathbf{U}_\star$ by computing the projection distance:

$$\begin{aligned} d_k &= \sqrt{r - \left\|\mathbf{U}_k^\top \mathbf{U}_\star\right\|_F^2}, \\ \mathbf{U}_k &= \mathrm{topU}_r(\mathbf{B}_k \mathbf{A}_k), \end{aligned} \tag{11}$$

where $\mathrm{topU}_r(\cdot)$ denotes the top-r left singular vectors. This distance provides a geometry-consistent measure of directional misalignment between the client update and the shared subspace, independent

of the basis chosen for either subspace. We then map d_k to a residual weight via:

$$\omega_k = \frac{1}{1 + d_k/\tau}, \tag{12}$$

where $\tau > 0$ is a temperature parameter controlling the rate at which contributions from increasingly misaligned updates are downweighted. The resulting weights are combined with normalized dataset proportions p_k to form the final aggregation coefficients:

$$\tilde{\omega}_k = \frac{p_k \omega_k}{\sum_{j=1}^{K} p_j \omega_j}. \tag{13}$$

Finally, the orthogonal components are aggregated as:

$$\bar{\mathbf{B}}_\perp = \sum_{k=1}^{K} \tilde{\omega}_k \, \mathbf{B}_{k,\perp}. \tag{14}$$

This design suppresses the influence of highly misaligned residual directions while retaining useful client-specific variations. The final aggregated LoRA parameters are given by

$$\mathbf{A}_g = \bar{\mathbf{A}}, \; \mathbf{B}_g = \bar{\mathbf{B}}_\parallel + \bar{\mathbf{B}}_\perp. \tag{15}$$

5.2 Direction Alignment Regularization

Motivation. While geometry-aware server aggregation can mitigate directional conflicts after model updates are collected, it does not prevent strongly misaligned update directions from being generated during local training. Under severe data heterogeneity, purely local optimization may encourage clients to explore update subspaces that deviate substantially from the shared global geometry, increasing the burden on server-side coordination. To address this issue proactively, it is desirable to gently bias local adaptation toward directions that are compatible with the globally shared subspace, without enforcing strict constraints that could hinder personalization or convergence. Motivated by this consideration, we introduce direction alignment regularization that penalizes the energy of update components orthogonal to the consensus subspace during client-side training.

Direction Alignment Regularization. Given the consensus subspace $\mathbf{U}_\star$, each client augments its local objective with a direction alignment penalty:

$$\mathcal{L}_{\text{dir}} = \left\| (\mathbf{I} - \mathbf{U}_\star \mathbf{U}_\star^\top)(\mathbf{B}\mathbf{A}) \right\|_F^2. \tag{16}$$

This regularization penalizes the energy of update components lying outside the shared principal subspace, serving as a relaxed surrogate for explicit subspace alignment. Rather than enforcing exact agreement, it softly biases local updates toward directions compatible with the global geometry.

Local Optimization Objective. Each client k solves the following local optimization problem:

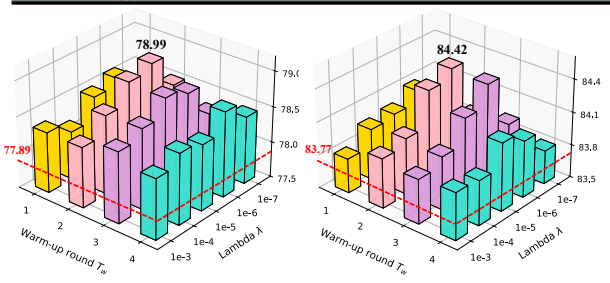
$$\min_{\mathbf{A},\mathbf{B}} \; \mathcal{L}_k(\mathbf{A}, \mathbf{B}) + \lambda \mathcal{L}_{\text{dir}}, \tag{17}$$

where $\mathcal{L}_k$ denotes the task loss on client k and λ is the regularization coefficient. To improve optimization stability, we apply this regularization after a warm-up phase. Specifically, the direction alignment penalty is activated after T_w communication rounds, allowing clients to adapt freely to their local data before being guided toward the shared subspace. This warm-up strategy prevents overly restrictive bias in early training, when the consensus subspace may still be inaccurate.

Together with the geometry-aware server aggregation, the proposed direction alignment regularization forms a complementary client–server mechanism. While server-side aggregation resolves directional conflicts at the subspace level, client-side regularization proactively limits the emergence of highly misaligned updates during local optimization. Overall, the combined mechanism mitigates directional conflicts at both optimization and aggregation stages, ensuring stable and consistent low-rank coordination across heterogeneous clients while preserving flexibility during local optimization.

Table 1: **Comparison with the state-of-the-art solutions on Federated Dataset 1.** The best results are highlighted in bold. Please refer to Sec. 6.2 for detailed analysis.

Methods	Paraphrase	Entailment	Text Formatting	Struct2Text	Ling. Acc.	Word Dis.	Coreference	Question CLS	Average
LLaMA2-7B									
Zero-shot	4.70	9.20	39.19	60.60	6.86	7.67	10.10	5.96	18.03
FedIT	76.00	81.50	68.85	96.94	70.00	64.50	73.37	92.00	77.89
FFA-LoRA	65.00	35.00	59.50	96.09	69.50	49.00	62.67	73.00	63.72
FlexLoRA	73.50	81.00	72.06	96.80	72.00	65.00	74.80	92.50	78.46
LoRA-FAIR	74.00	84.00	70.31	96.67	73.00	65.50	75.87	92.00	78.92
ALIGN	75.50	83.50	70.12	96.88	75.00	64.50	76.44	93.00	**79.37**
Qwen2.5-7B									
Zero-shot	2.35	6.09	60.67	41.78	2.94	5.33	6.00	3.19	16.04
FedIT	79.00	91.50	97.39	78.76	80.50	73.50	78.03	91.50	83.77
FFA-LoRA	80.50	91.50	97.04	78.25	82.00	71.50	75.12	93.00	83.61
FlexLoRA	82.00	91.50	96.73	77.35	79.50	70.00	80.65	90.50	83.53
LoRA-FAIR	78.50	92.00	97.01	77.15	79.50	77.00	77.58	92.50	83.91
ALIGN	80.00	91.50	97.41	77.59	80.50	75.00	81.38	93.00	**84.55**



(a) LLaMA2-7B (b) Qwen2.5-7B

Figure 4: **Hyperparameter Study** for regularization coefficient λ and warm-up rounds T_w. Refer to Sec. 6.3.

Table 2: **Ablation Study of Key Components** for ALIGN, showing the effects of Geometry-Aware Aggregation (GAA) and Direction Alignment Regularization (DAR). Refer to Sec. 6.3.

GAA	DAR	LLaMA2	Qwen2.5
		77.89	83.77
✓		78.93 ↑1.04	84.33 ↑0.56
	✓	78.99 ↑1.10	84.42 ↑0.65
✓	✓	**79.37** ↑1.48	**84.55** ↑0.78

6 Experiments

6.1 Experimental Setup

Datasets and Architecture. We follow FedDPA [40] to simulate heterogeneous client data using FLAN [37], a large-scale instruction-tuning benchmark with over 60 NLP datasets. Following prior work [40], two federated datasets are considered in our experiments, referred to as Federated Dataset 1 and Federated Dataset 2. In the main experiments, we adopt Federated Dataset 1 as the primary benchmark, while additional experiments on Federated Dataset 2 are provided in the Appendix B to further evaluate the robustness of our method. For each dataset, 8 different NLP tasks are randomly selected, with each task treated as an individual client, and the datasets are further downsampled to accommodate the federated learning setting. Additional details are provided in the Appendix A. We adopt LLaMA2-7B [35] and Qwen2.5-7B [39] as backbone models. Both model are initialized from the publicly released pretrained weights and fine-tuned under the federated learning setting. ROUGE-1 is used as the primary evaluation metric for all tasks and both backbone architectures.

Implementation Details. To ensure a fair comparison, all methods are evaluated under the same training configuration. The batch size for local training is set to 128. In each communication round, each client performs three local update steps before transmitting model updates to the server. The training process is conducted for 20 communication rounds. AdamW [27] is used as the optimizer with an initial learning rate of 3e-4 across all experiments. The LoRA rank is set to 8 and is applied only to W_q and W_v. We conduct experiments on two NVIDIA RTX 4090 GPUs.

Comparison Baselines. We compare our method with several recent federated learning baselines:

- **FedIT** [ICASSP'24] [45] applies LoRA to federated instruction tuning, with clients optimizing lightweight adapters on local data and the server averaging their parameters through FedAvg [29].

- **FFA-LoRA** [ICLR'24] [34] fixes the randomly initialized non-zero A matrices and fine-tunes only the zero-initialized B matrices across training rounds, aggregating B matrices on the server.
- **FlexLoRA** [NeurIPS'24] [1] merges client LoRA updates into a global weight update and reconstructs local low-rank adapters with different ranks through singular value decomposition.
- **LoRA-FAIR** [ICCV'25] [3] reduces aggregation bias through refined server-side aggregation and improved initialization of LoRA modules across rounds.

6.2 Comparison to State-of-the-Arts

We evaluate ALIGN on a federated NLP benchmark with different backbone models, covering classification, reasoning, and generation tasks. As shown in Tab. 1, ALIGN achieves the best overall performance on both backbones, demonstrating its effectiveness and generality under heterogeneous federated data distributions. A key observation is that the advantage of ALIGN is consistent across LLaMA2-7B and Qwen2.5-7B, although the two backbones show different task preferences. This indicates that ALIGN does not rely on model-specific characteristics, but provides a generally effective optimization strategy for FedLoRA. Overall, these results highlight the robustness and effectiveness of ALIGN in federated LoRA tuning across the two backbone architectures.

6.3 Diagnostic Analysis

Hyperparameter Study. We show the impact of the regularization coefficient λ and the number of warm-up rounds T_w on the performance in Fig. 4. All configurations consistently outperform the FedIT baseline, demonstrating the robustness of our method across different hyperparameter settings. Both backbones exhibit similar trends, with stronger performance concentrated around a balanced regularization strength and a short warm-up schedule. The best performance is achieved with $\lambda = 10^{-6}$ and $T_w = 2$, which are used in all subsequent experiments.

Ablation of Key Components. We conduct an ablation study to evaluate the contribution of the two core components in ALIGN: Geometry-Aware Aggregation (GAA) and Direction Alignment Regularization (DAR), as reported in Tab. 2. Compared with the baseline, both GAA and DAR consistently improve performance across different backbone models, demonstrating their effectiveness in mitigating subspace misalignment. When combined, GAA and DAR achieve the best overall performance, highlighting their complementary roles in coordinating low-rank adaptations under heterogeneous federated settings across different backbones.

6.4 Limitation

ALIGN relies on estimating a consensus subspace from client LoRA updates, which inevitably incurs additional communication and server-side computation. To better characterize the practical cost, we provide a quantitative comparison with baselines that involve server-side operations. As shown in Tab. 3, we report the communication cost as the total upload and download size per round, and the computation cost as the server-side FLOPs per round. ALIGN incurs only modest communication overhead and substantially less server-side computation than LoRA-FAIR. Despite this limited overhead, ALIGN achieves the best overall result. These results demonstrate that ALIGN incurs acceptable practical overhead while providing a favorable efficiency–effectiveness trade-off.

Table 3: **Communication and Computation Overhead Comparison**. See details in Sec. 6.4.

	FlexLoRA	LoRA-FAIR	ALIGN
Comm. Cost (MB)	256	256	288
Server FLOPs	0.30	13.04	5.91
Result	78.46	78.92	79.37

7 Conclusion

This paper identifies directional conflicts as a key challenge in FedLoRA, where different clients learn misaligned low-rank updates that interfere during aggregation. Unlike aggregation bias, this issue stems from incompatible low-rank subspaces under data heterogeneity. We propose ALIGN, a geometry-aware federated adaptation framework that mitigates destructive interference through (i) Geometry-Aware Aggregation on the server and (ii) Direction Alignment Regularization on clients. Experiments show that ALIGN significantly improves performance, highlighting the importance of subspace geometry in FedLoRA, especially when clients learn diverse adaptation directions.

Acknowledgement

This work is funded by National Natural Science Foundation of China under Grant (62361166629, 623B2080, 62376264), the Major Project of Science and Technology Innovation of Hubei Province (2024BCA003), Key Research and Development Program of Wuhan (2025061202030423), the Innovative Research Group Project of Hubei Province under Grants 2024AFA017, the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) under Grant No. GMLKF-24-10, the Independent Research Project of the National Key Laboratory of Big Data and Decision, Brain Science and Brain-like Intelligence Technology—National Science and Technology Major Project under Grant 2022ZD0208700, the National Research Foundation (NRF), Singapore, under Grant ID NRF-PA2025-NTU, and by Nanyang Technological University (NTU). The supercomputing system at the Supercomputing Center of Wuhan University supported the numerical calculations in this paper.

References

[1] Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In *NeurIPS*, 2024.

[2] Jieming Bian, Lei Wang, Letian Zhang, and Jie Xu. Fedalt: Federated fine-tuning through adaptive local training with rest-of-world lora. *arXiv preprint arXiv:2503.11880*, 2025.

[3] Jieming Bian, Lei Wang, Letian Zhang, and Jie Xu. Lora-fair: Federated lora fine-tuning with aggregation and initialization refinement. In *ICCV*, pages 3737–3746, 2025.

[4] Rudrasis Chakraborty and Baba C Vemuri. Recursive frechet mean computation on the grassmannian and its applications to computer vision. In *ICCV*, pages 4229–4237, 2015.

[5] Shuangyi Chen, Yue Ju, Hardik Dalal, Zhongwen Zhu, and Ashish J Khisti. Robust federated finetuning of foundation models via alternating minimization of loRA. In *NeurIPS*, 2025.

[6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM*, 20(2):303–353, 1998.

[9] Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *ICLR*, 2025.

[10] Jihun Hamm and Daniel D Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *ICML*, pages 376–383, 2008.

[11] Wenwen He, Wenke Huang, Yi Liu, Jian Liang, Xirui Li, Guansong Pang, and Mang Ye. Fedpissa: Towards federated personalized adaptation of foundation models via lora subspace mapping. In *ICML*, 2026.

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, pages 1–26, 2022.

[13] Wenke Huang, Yangxu Liao, Wenjia Lei, Guancheng Wan, Xuankun Rong, Chi Wen, He Li, Mang Ye, Qingqing Wu, and Bo Du. Federated clinical concept and disease semantic learning for congenital heart disease diagnosis. *npj Digital Medicine*, 2026.

[14] Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, Bo Du, and Qiang Yang. Federated learning for generalization, robustness, fairness: A survey and benchmark. *IEEE PAMI*, 2024.

[15] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Projection metric learning on grassmann manifold with application to video based face recognition. In *CVPR*, pages 140–149, 2015.

[16] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, pages 1–210, 2021.

[17] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, pages 1–38, 2016.

[18] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[19] Dawid J Kopiczko, Tijmen Blankevoort, and Yuki M Asano. Vera: Vector-based random matrix adaptation. In *ICLR*, 2023.

[20] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021.

[21] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE SPM*, pages 50–60, 2020.

[22] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[23] Zhixuan Li, Binqian Xu, Xiangbo Shu, Jiachao Zhang, Yazhou Yao, Guo-Sen Xie, and Jinhui Tang. Tensor-aggregated lora in federated fine-tuning. In *ICCV*, pages 1058–1067, October 2025.

[24] Jian Liang, Wenke Huang, Xianda Guo, Guancheng Wan, Bo Du, and Mang Ye. Lorasculpt: Harmonious low-rank adaptation for multimodal large language models. *IEEE PAMI*, 2026.

[25] Jian Liang, Wenke Huang, Guancheng Wan, Qu Yang, and Mang Ye. Lorasculpt: Sculpting lora for harmonizing general and specialized knowledge in multimodal large language models. In *CVPR*, pages 26170–26180, 2025.

[26] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *ICML*, 2024.

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[28] Gen Luo, Minglang Huang, Yiyi Zhou, Xiaoshuai Sun, Guannan Jiang, Zhiyu Wang, and Rongrong Ji. Towards efficient visual adaption via structural re-parameterization. *arXiv preprint arXiv:2302.08106*, 2023.

[29] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282. PMLR, 2017.

[30] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *IEEE SSP*, pages 19–38, 2017.

[31] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[32] George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. *ICLR*, 2025.

[33] Ximeng Sun, Ping Hu, and Kate Saenko. Dualcoop: Fast adaptation to multi-label recognition with limited annotations. In *NeurIPS*, pages 30569–30582, 2022.

[34] Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving loRA in privacy-preserving federated learning. In *ICLR*, 2024.

[35] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[36] Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In *NeurIPS*, 2024.

[37] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *ICLR*, 2022.

[38] Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*, 2023.

[39] An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, et al. Qwen2. 5-1m technical report. *arXiv preprint arXiv:2501.15383*, 2025.

[40] Yiyuan Yang, Guodong Long, Tao Shen, Jing Jiang, and Michael Blumenstein. Dual-personalizing adapter for federated foundation models. In *NeurIPS*, pages 39409–39433, 2024.

[41] Rui Ye, Rui Ge, Xinyu Zhu, Jingyi Chai, Du Yaxin, Yang Liu, Yanfeng Wang, and Siheng Chen. Fedllm-bench: Realistic benchmarks for federated learning of large language models. In *NeurIPS*, volume 37, pages 111106–111130, 2025.

[42] Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. Openfedllm: Training large language models on decentralized private data via federated learning. In *ACM SIGKDD*, 2024.

[43] Liping Yi, Han Yu, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. pfedlora: Model-heterogeneous personalized federated learning with lora tuning. *arXiv preprint arXiv:2310.13283*, 2023.

[44] Dan Zhang, Tao Feng, Lilong Xue, Yuandong Wang, Yuxiao Dong, and Jie Tang. Parameter-efficient fine-tuning for foundation models. *arXiv preprint arXiv:2501.13787*, 2025.

[45] Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP*, 2024.

[46] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

[47] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, pages 1–12, 2018.

[48] Junchao Zhou, Junkang Liu, and Fanhua Shang. Ilora: Federated learning with low-rank adaptation for heterogeneous client aggregation. *arXiv preprint arXiv:2511.16069*, 2025.

[49] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022.

APPENDIX

A Datasets Details

Following the experimental setting of FedDPA [40], two federated learning benchmarks derived from the Flan [37] dataset are used in this work, referred to as **Federated Dataset 1** and **Federated Dataset 2**. In the main experiments, we adopt Federated Dataset1 as the primary benchmark, while additional experiments on Federated Dataset2 are provided in the appendix to further evaluate the robustness and generality of our method. Flan covers a diverse set of NLP tasks with substantial distributional heterogeneity across datasets. Under this setting, one dataset is randomly selected from each of eight distinct task categories, with each selected dataset treated as the local data of an individual client, resulting in eight clients per federated dataset. To simulate limited local data availability, each client dataset is downsampled to 300 training samples and 200 testing samples. As a result, each federated dataset contains 2,400 training instances and 1,600 testing instances in total. The detailed task and dataset composition is reported in Tab. 4. The benchmarks include both generation and classification tasks. All tasks are reformulated into a unified generative, instruction-following format with task-specific prompts for large language models. Representative examples and the input template are provided in Tab. 5.

Table 4: Tasks and datasets of constructed federated datasets.

Federated Dataset 1		**Federated Dataset 2**	
Task	Dataset	Task	Dataset
Paraphrase	glue_qqp	Paraphrase	paws_wiki
Entailment	snli	Commonsense	hellaswag
Structure to text	web_nlg_en	Entailment	qnli
Text formatting	fix_punct	Text formatting	word_segment
Linguistic acceptability	cola	Summarization	gigaword
Word disambiguation	wic	Reading comprehension	bool_q
Coreference	definite_pronoun_resolution	Sentiment	sentiment140
Question classification	trec	Open-domain QA	acr_easy

Table 5: Examples of data in federated datasets.

Data	**Examples**
Input:	The father convinced his son that it is possible for him to one day become a knight, but he may never achieve such status coming from a peasant family. Who is "he"? **OPTIONS:** – The father – his son
Output:	His son
Input:	Police are seeking a former village chief in north China for allegedly killing his political rivals in an attack apparently motivated by local power plays, state press reported Monday. Can you generate a short summary of the above paragraph?
Output:	Former Chinese village head wanted for political murders

B Additional Experiments

We further evaluate ALIGN on a constructed federated benchmark, denoted as federated dataset 2. Compared with federated dataset 1, federated dataset 2 introduces stronger heterogeneity by covering more diverse task distributions, including classification, reasoning, reading comprehension, summarization and so on. Such diversity leads to more heterogeneous client objectives and update directions, making federated optimization more challenging. As shown in Tab. 6, ALIGN achieves the best average performance on federated dataset 2 with both LLaMA2-7B and Qwen2.5-7B. This

Table 6: **Comparison with the state-of-the-art solutions on federated dataset 2.**

Methods	Commonsense	Paraphrase	Entailment	Reading Com	Text Formatting	Open QA	Summarization	Sentiment	Average
LLaMA2-7B									
Zero-shot	27.27	2.93	3.25	2.76	19.14	14.05	15.99	6.89	11.54
FedIT	75.13	83.51	79.00	80.50	88.58	81.08	31.00	74.50	74.16
FFA-LoRA	55.52	58.00	54.00	74.00	72.60	79.52	24.72	67.50	60.73
FlexLoRA	68.38	81.50	77.50	76.50	88.71	83.14	31.02	74.00	72.59
LoRA-FAIR	74.20	83.50	78.50	83.00	90.14	83.22	31.00	72.00	74.45
ALIGN	72.90	83.59	79.01	81.50	89.65	85.64	31.36	74.50	**74.77**
Qwen2.5-7B									
Zero-shot	52.10	2.55	2.81	2.65	63.46	9.50	16.38	2.69	19.02
FedIT	89.98	89.50	91.50	81.00	94.83	97.20	31.82	73.00	81.10
FFA-LoRA	90.24	88.50	91.50	82.50	93.34	96.34	29.61	76.00	81.00
FlexLoRA	90.14	87.50	89.50	81.50	95.33	96.42	31.63	75.00	80.88
LoRA-FAIR	89.75	87.50	90.00	82.50	94.10	96.70	32.08	77.00	81.20
ALIGN	90.37	87.50	93.00	82.50	94.88	97.90	31.53	73.60	**81.41**

consistent advantage shows that ALIGN is not only effective on the original federated dataset, but also remains robust under stronger data heterogeneity. Overall, these results further demonstrate the robustness and generality of ALIGN in highly heterogeneous federated scenarios.

C Algorithm

An overview of the complete ALIGN pipeline is presented in Algorithm 1

D Closed-form Fréchet Mean on the Grassmann Manifold

Proposition 1 (Closed-form extrinsic Fréchet mean on the Grassmann manifold)**.** *Let* $[U_k] \in \mathrm{Gr}(r, d)$ *denote the* r*-dimensional client subspace, where* $U_k \in \mathbb{R}^{d\times r}$ *is any orthonormal basis satisfying* $U_k^\top U_k = I_r$*. Let* $p_k \geq 0$ *and* $\sum_{k=1}^K p_k = 1$*. Consider the squared projection distance*

$$d^2_{\mathrm{proj}}([U_k], [U]) = r - \|U_k^\top U\|_F^2, \tag{18}$$

where $U \in \mathbb{R}^{d\times r}$ *is any orthonormal basis of a candidate subspace* $[U] \in \mathrm{Gr}(r, d)$*. Define the weighted Fréchet objective*

$$\mathcal{F}([U]) = \sum_{k=1}^K p_k d^2_{\mathrm{proj}}([U_k], [U]). \tag{19}$$

Let

$$\bar{P} = \sum_{k=1}^K p_k U_k U_k^\top \tag{20}$$

and let $q_1, \ldots, q_r$ *be eigenvectors of* $\bar{P}$ *corresponding to its largest* r *eigenvalues. Then any subspace*

$$[U_\star] = \mathrm{span}\{q_1, \ldots, q_r\} \tag{21}$$

is a global minimizer of (19)*. Equivalently, if*

$$\widetilde{U} = \left[\sqrt{p_1}U_1,\ \sqrt{p_2}U_2,\ \ldots,\ \sqrt{p_K}U_K\right], \tag{22}$$

then $[U_\star]$ *is spanned by the top-*r *left singular vectors of* $\widetilde{U}$*. If* $\lambda_r(\bar{P}) > \lambda_{r+1}(\bar{P})$*, the minimizer is unique as a point on* $\mathrm{Gr}(r, d)$*.*

Proof. First, note that the squared projection distance is well-defined on the Grassmann manifold. Indeed, replacing U_k by $U_k R_k$ or U by UR, where $R_k, R \in \mathbb{R}^{r\times r}$ are orthogonal matrices, does not change $\|U_k^\top U\|_F^2$. Using (18) and $\sum_{k=1}^K p_k = 1$, we have

$$\begin{aligned}\mathcal{F}([U]) &= \sum_{k=1}^K p_k \big(r - \|U_k^\top U\|_F^2\big) \\ &= r - \sum_{k=1}^K p_k \|U_k^\top U\|_F^2. \end{aligned} \tag{23}$$

Algorithm 1: ALIGN

Input: Rounds T, warm-up round T_w, clients K, datasets $\{D_k\}$ with weights $\{p_k\}$, global LoRA $(\mathbf{A}, \mathbf{B})$, global subspace $\mathbf{U}_\star$, local steps E, learning rate η, regularization coefficient λ.

for $t = 1, 2, \ldots, T$ **do**

Client Side;

for $k = 1, \ldots, K$ **do**

$(\mathbf{A}_k, \mathbf{B}_k) \leftarrow$ **LocalUpdate**$(\mathbf{A_g}, \mathbf{B_g}, \mathbf{U}_\star, t)$;

Upload $(\mathbf{A}_k, \mathbf{B}_k)$ to server;

end

Server Side;

Update global subspace $\mathbf{U}_\star$ by Eq. (5);

`/* Geometric Decomposition */`

for $k = 1, \ldots, K$ **do**

$\mathbf{B}_{k,\|} \leftarrow \mathbf{U}_\star \mathbf{U}_\star^\top \mathbf{B}_k,\ \mathbf{B}_{k,\perp} \leftarrow \mathbf{B}_k - \mathbf{B}_{k,\|}$;

end

`/* Aligned Components Aggregation */`

$\bar{\mathbf{A}} \leftarrow \sum_{k=1}^K p_k \mathbf{A}_k,\ \bar{\mathbf{B}}_\| \leftarrow \sum_{k=1}^K p_k \mathbf{B}_{k,\|}$;

`/* Orthogonal Residuals Fusion */`

Compute weights $\{\tilde{\omega}_k\}$ by Eqs. (12) and (13);

$\bar{\mathbf{B}}_\perp \leftarrow \sum_{k=1}^K \tilde{\omega}_k \mathbf{B}_{k,\perp}$;

`/* Final Aggregated LoRA */`

$\mathbf{A_g} \leftarrow \bar{\mathbf{A}},\ \mathbf{B_g} \leftarrow \bar{\mathbf{B}}_\| + \bar{\mathbf{B}}_\perp$;

Broadcast $(\mathbf{A_g}, \mathbf{B_g}, \mathbf{U}_\star)$ to all clients $k = 1, \ldots, K$;

end

LocalUpdate$(\mathbf{A_g}, \mathbf{B_g}, \mathbf{U}_\star, t)$::

Initialize $(\mathbf{A}_k, \mathbf{B}_k) \leftarrow (\mathbf{A_g}, \mathbf{B_g})$;

for $e = 1, \ldots, E$ **do**

for $(x, y) \in D_k$ **do**

if $t > T_w$ **then**

Compute direction loss $\mathcal{L}_{\text{dir}}$ by Eq. (16);

$\mathcal{L} \leftarrow \mathcal{L}_k + \lambda \mathcal{L}_{\text{dir}}$;

else

$\mathcal{L} \leftarrow \mathcal{L}_k$;

end

Update $(\mathbf{A}_k, \mathbf{B}_k)$ by AdamW with step size η;

end

end

return $(\mathbf{A}_k, \mathbf{B}_k)$;

Therefore, minimizing $\mathcal{F}([U])$ is equivalent to maximizing

$$\sum_{k=1}^K p_k \|U_k^\top U\|_F^2. \tag{24}$$

For each k,

$$\|U_k^\top U\|_F^2 = \mathrm{Tr}(U^\top U_k U_k^\top U). \tag{25}$$

Substituting (25) into (24) yields

$$\begin{aligned}\sum_{k=1}^K p_k \|U_k^\top U\|_F^2 &= \sum_{k=1}^K p_k \mathrm{Tr}(U^\top U_k U_k^\top U) \\ &= \mathrm{Tr}\left(U^\top \left(\sum_{k=1}^K p_k U_k U_k^\top\right) U\right) \\ &= \mathrm{Tr}(U^\top \bar{P} U).\end{aligned} \tag{26}$$

Thus the Fréchet mean problem is equivalent to the constrained trace maximization

$$\max_{U^\top U = I_r} \operatorname{Tr}(U^\top \bar{P} U). \tag{27}$$

The matrix $\bar{P}$ is symmetric positive semidefinite because it is a nonnegative weighted sum of orthogonal projection matrices. By the Ky Fan maximum principle for subspaces,

$$\max_{U^\top U = I_r} \operatorname{Tr}(U^\top \bar{P} U) = \sum_{j=1}^{r} \lambda_j(\bar{P}), \tag{28}$$

where $\lambda_1(\bar{P}) \geq \lambda_2(\bar{P}) \geq \cdots \geq \lambda_d(\bar{P})$ are the eigenvalues of $\bar{P}$. The maximum is attained when the columns of U span the principal r-dimensional eigenspace of $\bar{P}$. Hence any orthonormal basis of $\operatorname{span}\{q_1, \ldots, q_r\}$ is a global maximizer of (27), and therefore the corresponding Grassmann point $[U_\star]$ is a global minimizer of (19). It remains to show the equivalent SVD computation. By the definition of $\widetilde{U}$ in (22),

$$\widetilde{U}\widetilde{U}^\top = \sum_{k=1}^{K} p_k U_k U_k^\top = \bar{P}. \tag{29}$$

Therefore, the top-r left singular vectors of $\widetilde{U}$ are precisely the eigenvectors associated with the largest r eigenvalues of $\bar{P}$. Thus the closed-form Fréchet mean is equivalently obtained by a single SVD of the weighted concatenation matrix $\widetilde{U}$. □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the contributions and scope of this paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Sec. 6.4.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: For each theoretical result, the paper provide the full set of assumptions and a complete (and correct) proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We fully disclose all the information needed to reproduce the main experimental results in this paper and our code. We are convinced that the obtained results can be reproduced.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is accessible in this paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all necessary training and evaluation details in Sec. 6.1

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following the setup of FedDPA [40], we adopt a fixed random seed for all reported experiments. We do not report error bars or statistical significance tests, partly also due to the large number of experiments and the substantial computational cost of training LLMs. We plan to include repeated trials in future work for a more comprehensive statistical analysis.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information on the computer resources in Sec. 6.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper strictly adheres to the NeurIPS Code of Ethics, ensuring that all aspects of the work are in compliance with the guidelines provided.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Although federated learning can have positive societal impacts on privacy preservation, this work focuses on improving federated learning for foundation models. Our evaluation only uses publicly available NLP datasets. Therefore, the societal impacts of this work are not directly applicable, and we mark this item as N/A.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets used in this paper, including LLaMA2-7B and Qwen2.5-7B models, as well as the FLAN datasets, are properly cited with corresponding references. Their licenses are respected, and no unauthorized or scraped content was used.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our research does not involve human subjects or crowdsourcing, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Our research does not involve the use of LLMs in any important, original,or non-standard way. LLMs were not used for data generation, analysis, or methodology design.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.