DIFFERENTIALLY PRIVATE FINE-TUNING IN LEARNED SUBSPACES OF LORA

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter-efficient fine-tuning (PEFT) has emerged as an effective paradigm for adapting large language models to downstream tasks involving privacy-sensitive data. Differential privacy (DP) mitigates privacy leakage during fine-tuning but often introduces substantial utility degradation. Focusing on low-rank adaptation (LoRA), a representative PEFT method, we observe that although LoRA constrains the rank of weight updates, its parameterization remains highly redundant, which can result in large noise and impaired fine-tuning utility. To address this limitation, we propose a two-stage framework for differentially private fine-tuning within a learned low-dimensional subspace of LoRA parameters. Our method first identifies task-relevant update subspaces by extracting dominant directions from LoRA training trajectories, and then reparameterizes LoRA factors to perform private fine-tuning directly in this compact subspace. By restricting noise-perturbed updates to a low-dimensional, task-aligned space, the proposed method reduces the effective impact of DP noise while preserving the same privacy guarantees. We theoretically characterize how subspace-based updates reduce the effective magnitude of injected DP noise compared to high-dimensional parameter updates. Extensive experiments on multiple benchmark datasets demonstrate that our method consistently outperforms representative differentially private fine-tuning baselines across a broad range of privacy settings. Our code is available at https://anonymous.4open.science/r/8BA2.

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable performance across a wide range of downstream tasks Brown et al. (2020); Touvron et al. (2023); OpenAI et al. (2024). However, fine-tuning all model parameters is often computationally expensive and incurs substantial memory overhead Ding et al. (2023). This motivates the development of parameter-efficient fine-tuning (PEFT) methods Houlsby et al. (2019); Ben Zaken et al. (2022); Dettmers et al. (2023), which update only a small subset of parameters while keeping the backbone model frozen. Among these, Low-Rank Adaptation (LoRA) Hu et al. (2021) has emerged as a representative and widely adopted method. By introducing low-rank structures into weight updates, LoRA significantly reduces the number of trainable parameters while achieving performance close to full fine-tuning.

Fine-tuning LLMs typically requires access to task-specific data, which are often privacy-sensitive in many domains Fu et al. (2024), such as user-generated content, medical records, and financial documents. This raises critical privacy concerns during the fine-tuning process. Differential privacy (DP) Dwork et al. (2006) provides rigorous theoretical guarantees and has become a standard method for privacy-preserving model training and fine-tuning. By injecting random noise into parameter updates, DP limits the influence of individual training samples on the learned model, thereby reducing the risk of privacy leakage.

Despite its strong privacy guarantees, differential privacy inevitably introduces noise into gradient updates, which can distort the optimization signal and degrade model utility. Consequently, achieving a favorable privacy–utility trade-off remains a fundamental challenge in differentially private fine-tuning. Combining DP with LoRA provides a natural approach to this problem: instead of privately optimizing the entire pretrained model, DP-SGD is applied only to the trainable low-rank factors while the backbone remains frozen. This substantially reduces the number of trainable parameters

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and the computational overhead of private fine-tuning. Recent work has explored differentially private LoRA fine-tuning on financial and medical datasets, demonstrating its potential as an efficient privacy-preserving adaptation strategy Liu et al. (2025).

However, while LoRA enforces low-rank constraints on weight updates, its updates are parameterized by a pair of factor matrices that still reside in a high-dimensional space. Recent studies, such as LoRI Zhang et al. (2025), show that training only a small fraction of parameters, up to 95% fewer than standard LoRA, can achieve comparable performance, suggesting potential redundancy in LoRA's parameterization. Under differential privacy constraints, noise is injected directly into this high-dimensional parameter space. As the effective impact of DP noise scales with the dimensionality of the update space, such redundancy amplifies the influence of privacy noise, making it easier to obscure task-relevant update signals and resulting in severe performance degradation. Therefore, although the low-rank structure of LoRA alleviates the impact of DP noise to some extent, directly applying DP to LoRA factor parameters remains insufficient to fundamentally address the utility loss in differentially private fine-tuning.

Importantly, existing methods that exploit redundancy in LoRA parameters primarily focus on post-hoc compression or efficiency improvements under standard fine-tuning settings and do not consider how parameter redundancy interacts with privacy noise injection. This observation motivates a key question: **Can we further constrain the effective space of differentially private updates such that noise perturbations act only on task-relevant directions, thereby mitigating the impact of DP noise on fine-tuning performance?**

To answer this question, we propose a two-stage framework for differentially private fine-tuning within a learned low-dimensional subspace of LoRA parameters. In the first stage, we analyze the training dynamics of LoRA parameters to identify a compact subspace that captures the dominant, task-relevant update directions while filtering out redundant ones. In the second stage, we reparameterize LoRA updates with respect to the learned subspace and perform fine-tuning by restricting all parameter updates to this subspace under DP constraints. By confining privacy-preserving updates to a low-dimensional space, our framework substantially reduces the effective impact of DP noise while preserving identical privacy guarantees. We further provide a theoretical characterization of how subspace-constrained private updates reduce the effective magnitude of injected DP noise compared to high-dimensional parameter updates. Extensive experiments on multiple benchmark tasks across a wide range of privacy settings demonstrate that our method consistently outperforms representative differentially private fine-tuning baselines while maintaining identical privacy guarantees.

2 Related Work

Differentially Private Fine-Tuning and Parameter-Efficient Adaptation. Differentially private fine-tuning of large pretrained models is challenging due to both the computational cost of per-example gradient processing and the utility degradation introduced by gradient clipping and Gaussian noise. Prior work has demonstrated effective differentially private fine-tuning of language models and developed efficient clipping and optimization techniques to reduce the substantial memory and computational overhead of DP-SGD Li et al. (2022); He et al. (2023); Yu et al. (2022). Another line of work combines differential privacy with parameter-efficient fine-tuning (PEFT), updating only a small subset of model parameters through methods such as Adapters, LoRA, and BitFit Houlsby et al. (2019); Hu et al. (2021); Ben Zaken et al. (2022). Under differential privacy, parameter-efficient adaptation can substantially reduce the computational and memory costs of private training while maintaining competitive utility Yu et al. (2022); Bu et al. (2024); Liu et al. (2024). However, reducing the number of trainable parameters alone does not explicitly restrict private optimization to task-relevant directions.

Low-Dimensional Subspaces for Fine-Tuning and Private Optimization. A broad line of work has shown that neural network adaptation often exhibits substantial low-dimensional structure. Intrinsic-dimension studies demonstrate that neural networks and pretrained language models can be optimized within much lower-dimensional spaces while retaining much of the performance of full-parameter fine-tuning Li et al. (2018); Aghajanyan et al. (2021). Zhang et al. Zhang et al. (2023) further show that task-specific intrinsic subspaces can be identified from fine-tuning trajectories, while recent analyses of LoRA reveal substantial redundancy in its parameterization Cheng et al. (2025).

Low-dimensional structure has also been exploited in differentially private optimization. GEP Yu et al. (2021a) uses a non-sensitive gradient subspace to decompose private gradients into low-dimensional and residual components, while RGP Yu et al. (2021b) employs low-rank reparameterization to enable large-scale private optimization. Publicly estimated gradient subspaces can reduce the ambient-dimension dependence of private learning Kairouz et al. (2021), and subsequent work studies how the choice of public data affects the resulting gradient subspace Gu et al. (2022). More recently, PE-SGD Zou et al. (2026) studies evolving public-data-derived gradient subspaces for private text learning, and public-subspace-assisted private learning has also been explored in visual domains Sha et al. (2025). DP-SFT Zheng et al. (2026) constructs task-specific subspaces from optimization trajectories and performs private fine-tuning within the resulting low-dimensional space.

Our framework differs from existing subspace-based private optimization methods in both how the subspaces are constructed and how they are used during private fine-tuning. Specifically, we learn fixed, task-relevant subspaces from the public optimization trajectories of LoRA factors and directly optimize the corresponding low-dimensional subspace parameters on the private dataset. Consequently, all private updates are intrinsically confined to the learned LoRA subspaces, without requiring explicit projection of full-dimensional private gradients at each step.

3 PRELIMINARIES

3.1 DIFFERENTIAL PRIVACY

Definition 1 ((ε, δ)-Differential Privacy)**.** *A randomized mechanism $\mathcal{A}$ satisfies $(\varepsilon, \delta)-$differential privacy if for any neighboring datasets D and D' differing in one record and for all sets $\mathcal{S} \subseteq \mathrm{Range}(\mathcal{A})$,*

$$\Pr[\mathcal{A}(D) \in \mathcal{S}] \leq e^{\varepsilon} \Pr[\mathcal{A}(D') \in \mathcal{S}] + \delta, \tag{1}$$

where ε denotes the privacy budget, which quantifies the strength of the privacy guarantee, and δ is a failure probability that allows a small probability of violating pure differential privacy.

Differentially private stochastic gradient descent (DP-SGD) Abadi et al. (2016) is the most commonly used method for differentially private optimization in practice. At each iteration, it computes per-example gradients, clips each gradient to a fixed ℓ_2 norm bound to control sensitivity, and then adds Gaussian noise to the aggregated minibatch update. Specifically, given a minibatch of size B, DP-SGD clips per-example gradients $\{g_i\}_{i=1}^{B}$ according to $\mathrm{clip}(g, C) = g \cdot \min\{1,\, C/\|g\|_2\}$, and forms a noisy averaged gradient update as

$$\bar{g} \leftarrow \frac{1}{B}\Big(\sum_{i=1}^{B} \mathrm{clip}(g_i, C) \;+\; \mathcal{N}(0, \sigma^2 C^2 I)\Big), \tag{2}$$

where σ denotes the noise multiplier. The cumulative privacy loss over multiple iterations is typically tracked using standard privacy accounting methods, such as Rényi-differential privacy (RDP) Mironov (2017), optionally benefiting from privacy amplification under random subsampling Balle et al. (2018).

Theorem 1 (Sequential Composition)**.** *If mechanisms $\mathcal{A}_i$ satisfy $(\varepsilon_i, \delta_i)$-DP for $i = 1, \ldots, K$, then the joint mechanism $(\mathcal{A}_1, \ldots, \mathcal{A}_K)$ satisfies $\big(\sum_{i=1}^{K} \varepsilon_i, \sum_{i=1}^{K} \delta_i\big)$-DP.*

Theorem 2 (Post-processing)**.** *Let $\mathcal{A}$ be a randomized algorithm that is (ε, δ)-differentially private. Then for an arbitrary randomized mapping f, the composition $f \circ \mathcal{A}$ is also (ε, δ)-differentially private.*

3.2 SUBSPACE REPARAMETERIZATION

We use a linear subspace reparameterization to restrict optimization to a low-dimensional subspace. Let $P \in \mathbb{R}^{d \times r}$ be a basis matrix whose columns span a subspace, and let $\theta \in \mathbb{R}^r$ denote the corresponding parameters. The original parameters are represented as $W = P\theta$. For a differentiable loss function $L(W, x)$, the gradient with respect to θ is given by

$$\nabla_\theta L(P\theta, x) = P^\top \nabla_W L(W, x)\big|_{W = P\theta}. \tag{3}$$

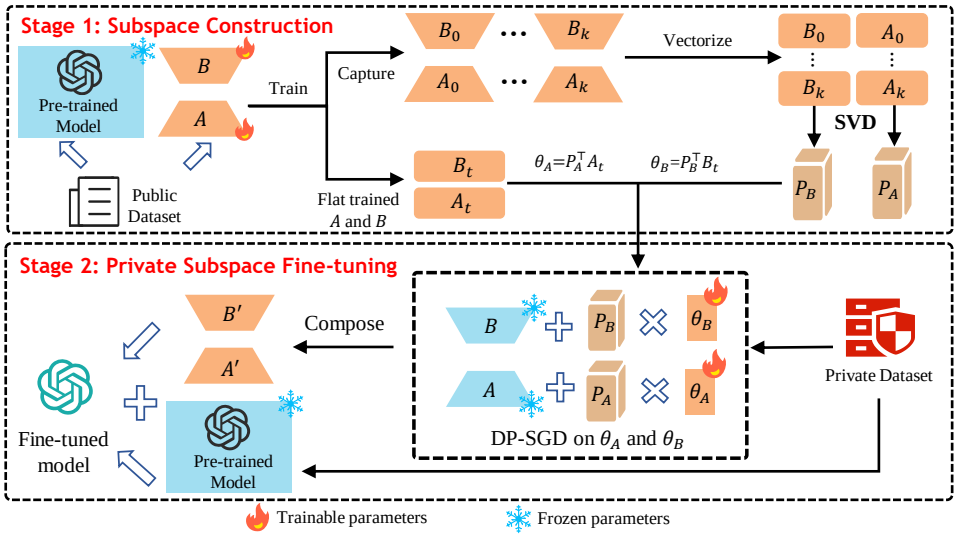


Figure 1: Overview of our framework. In Stage 1, orthonormal bases (P_A, P_B) are derived by applying SVD to trajectory matrices of the LoRA factors (A, B). In Stage 2, the subspace parameters (θ_A, θ_B) are optimized using the DP-SGD clipping-and-noise mechanism and the updated LoRA factors (A', B') are reconstructed to adapt the frozen backbone.

When $P^\top P = I_r$, gradient updates in θ induce updates in W that lie in the subspace $\mathrm{span}(P)$, equivalently corresponding to projected gradient updates in the original parameter space. This reparameterization will be used to describe subspace-constrained fine-tuning and to support subsequent privacy–utility analysis.

4 Methods

We propose a two-stage framework for differentially private fine-tuning of LoRA within learned low-dimensional subspaces, as shown in Fig. 1. The core idea is to reduce the impact of differential privacy noise by restricting private updates to a compact, task-relevant subspace, while preserving formal differential privacy guarantees. In the first stage, we identify low-dimensional subspaces that capture the dominant update directions of LoRA parameters based on their training trajectories. In the second stage, we perform differentially private fine-tuning by constraining optimization to the learned subspaces and updating only the corresponding low-dimensional parameters. This design enables private optimization to focus on task-relevant directions, thereby alleviating noise-induced performance degradation under differential privacy.

4.1 Subspace Construction

The goal of Stage 1 is to identify low-dimensional subspaces that capture the dominant update directions of LoRA parameters. Rather than constraining the parameter space a priori, we learn these subspaces directly from the optimization trajectories of the LoRA factors during training. The underlying intuition is that, although LoRA parameters are high-dimensional, their effective updates often concentrate along a small number of task-relevant directions. By extracting these directions from the observed training process, we obtain compact subspaces that preserve the most informative components of LoRA updates and can be used to constrain subsequent private fine-tuning.

Let A and B denote the trainable LoRA factors, and let A_0 and B_0 be their initial values. During Stage 1 training, we periodically record the evolution of these factors and collect snapshots of their parameter updates. Specifically, at predefined intervals, we compute the parameter displacements $\Delta A_t = A_t - A_0$ and $\Delta B_t = B_t - B_0$, which capture how the LoRA parameters evolve from their initialization over the course of optimization.

$$S_M = [\text{flat}(\Delta M_1), \text{flat}(\Delta M_2), \ldots, \text{flat}(\Delta M_T)]^\top \in \mathbb{R}^{T \times d_M}, \quad M \in \{A, B\}. \tag{4}$$

where $\text{flat}(\cdot) \in \mathbb{R}^{d_M}$ denotes vectorization into a column vector, and d_M is the dimensionality of the corresponding LoRA factor. These trajectory matrices summarize the evolution of the LoRA parameters during training and provide the basis for extracting their dominant update directions.

After collecting T snapshots of the LoRA parameter updates, we perform Singular Value Decomposition (SVD) on the trajectory matrices:

$$S_M = U_M \Sigma_M V_M^\top, \quad \text{for } M \in \{A, B\}. \tag{5}$$

Here, $\Sigma_M \in \mathbb{R}^{T \times d_M}$ is the diagonal matrix containing the singular values, $U_M \in \mathbb{R}^{T \times T}$ and $V_M \in \mathbb{R}^{d_M \times d_M}$ are the left and right singular matrix whose columns are left and right singular vectors. We select the top-r right singular vectors from V_M corresponding to the largest singular values in Σ_M, which form orthonormal bases and capture the principal update directions of the model. Therefore, projection matrices, $P_M \in \mathbb{R}^{d_M \times r}$, for $M \in \{A, B\}$, are given by:

$$P_M = [v_1^M, \ldots, v_r^M], \quad P_M^\top P_M = I_r. \tag{6}$$

To evaluate the effectiveness of a dimension-r subspace in capturing the full-space model update directions, we measure the fraction of update energy explained by the selected singular directions:

$$\rho_M(r) = \frac{\sum_{j=1}^{r} (\sigma_j^M)^2}{\sum_{j=1}^{\min(T, d_M)} (\sigma_j^M)^2}, \quad \text{for } M \in \{A, B\}, \tag{7}$$

where $\{\sigma_j^A\}$ and $\{\sigma_j^B\}$ are the singular values of the trajectory matrices S_A and S_B, respectively. When $\rho_A(r)$ and $\rho_B(r)$ approach 1, the selected bases capture most of the trajectory energy and provide strong constraints for the subsequent optimization in Stage 2.

In Stage 1, we optimize the LoRA factors using only the public auxiliary dataset D_{aux} to identify the task-relevant subspaces. Since this stage does not access the private training subset D_{priv}, it incurs no privacy cost. Differential privacy is enforced only in Stage 2, where DP-SGD is applied to optimize the model within the learned subspaces using D_{priv}. The pseudocode and detailed procedure for subspace construction are provided in Appendix B.1.

4.2 Private Subspace Fine-tuning

In the second stage, we perform differentially private fine-tuning within the low-dimensional subspaces identified in Stage 1, restricting updates to the most informative directions. We reparameterize the trainable parameters as

$$M = M_0 + P_M \theta_M, \quad \text{for } M \in \{A, B\}, \tag{8}$$

where M_0 denotes the initial value of the corresponding LoRA factor, and $\theta_A, \theta_B \in \mathbb{R}^r$ are the only trainable variables. By restricting optimization to the learned subspaces, we reduce the effective dimensionality of private updates, thereby mitigating the impact of differential privacy noise while maintaining strong fine-tuning performance.

To perform private optimization, we apply the DP-SGD mechanism to the gradients of the trainable variables θ_A and θ_B. Specifically, for each training example x_i, we compute the gradients $g_i^{\theta_A} = \nabla_{\theta_A} \mathcal{L}(P_A \theta_A, P_B \theta_B; x_i)$ and $g_i^{\theta_B} = \nabla_{\theta_B} \mathcal{L}(P_A \theta_A, P_B \theta_B; x_i)$. We then form the joint gradient $g_i^\theta = [g_i^{\theta_A}, g_i^{\theta_B}]$ and clip it to the ℓ_2 norm bound C:

$$\widehat{g}_i^\theta = g_i^\theta \cdot \min\left(1, \frac{C}{\|g_i^\theta\|_2}\right). \tag{9}$$

After clipping, we split the joint gradient $\widehat{g}_i^\theta$ into $\widehat{g}_i^{\theta_A}$ and $\widehat{g}_i^{\theta_B}$ for aggregation and noise injection.

Given a minibatch of size B, we aggregate the clipped gradients and add Gaussian noise to ensure differential privacy. Specifically, for each minibatch, we compute the noisy minibatch-averaged gradient as

$$\widetilde{g}^{\theta_M} = \frac{1}{B}\left(\sum_{i=1}^{B} \widehat{g}_i^{\theta_M} + \mathcal{N}(0, \sigma^2 C^2 I_r)\right), \quad \text{for } M \in \{A, B\}. \tag{10}$$

We then update the parameters θ_A and θ_B using the AdamW optimizer with the corresponding noisy gradients $\widetilde{g}^{\theta_A}$ and $\widetilde{g}^{\theta_B}$, respectively. The updated LoRA factors are reconstructed as $A = P_A\theta_A$ and $B = P_B\theta_B$, and applied to the frozen backbone. The pseudocode and detailed procedure of the private subspace fine-tuning are provided in Appendix B.2.

4.3 Transferability of Subspaces Across Tasks

In practice, auxiliary data drawn from exactly the same distribution as the private target dataset may be difficult to obtain, whereas public data from related tasks or similar distributions is often more accessible. We therefore consider a transfer setting in which the subspace is constructed from a related public task and subsequently reused for private fine-tuning on the target task.

Specifically, we construct the subspace from the parameter trajectories of the public task following the same procedure as in Stage 1. The resulting projection matrices are then fixed and used to constrain differentially private optimization on the private target dataset. This transfer is motivated by the observation that related tasks may share dominant parameter update directions, allowing a subspace learned from public data to retain useful optimization structure for the target task.

4.4 Theoretical Analysis

Implicit Gradient Projection. In our framework, we perform optimization on low-dimensional subspace parameters θ_A and θ_B, where the LoRA factors are reparameterized as $M = M_0 + P_M\theta_M$ for $M \in \{A, B\}$. The matrices P_A and P_B are orthonormal bases learned from the training trajectories, satisfying $P_A^\top P_A = I_r$ and $P_B^\top P_B = I_r$. This reparameterization ensures that the LoRA factors A and B always lie within the subspaces spanned by the columns of P_A and P_B.

Under this reparameterization, optimizing θ_A and θ_B induces projected gradient updates on the original LoRA factors:

$$M \leftarrow M - \eta\, P_M P_M^\top \nabla_M L(M, x), \quad \text{for } M \in \{A, B\}. \tag{11}$$

Thus, only the gradient component within $\mathrm{span}(P_A)$ (resp. $\mathrm{span}(P_B)$) contributes to the update. This implicit projection is enforced through the reparameterization and optimization process, without requiring explicit projection operations during training. The full derivation is provided in Appendix A.1.

Privacy Guarantee. The proposed framework satisfies (ε, δ)-differential privacy with respect to each data point $x_i \in \mathcal{D}_{\mathrm{priv}}$. Specifically, Stage 1 performs subspace construction using only an auxiliary dataset $\mathcal{D}_{\mathrm{aux}}$, which is assumed to be publicly available and is disjoint from the private dataset $\mathcal{D}_{\mathrm{priv}}$. Therefore, Stage 1 incurs no privacy cost with respect to $\mathcal{D}_{\mathrm{priv}}$. In Stage 2, the learned subspace is fixed and differentially private optimization is performed exclusively on $\mathcal{D}_{\mathrm{priv}}$ with privacy budget (ε, δ). Since the subspace is constructed without accessing $\mathcal{D}_{\mathrm{priv}}$ and remains fixed throughout private fine-tuning, the subsequent mapping through the fixed subspace does not incur any additional privacy loss. The complete proof is provided in Appendix A.2.

Noise Error Reduction. We compare the magnitude of the differential privacy noise injected by DP-SGD in the full d-dimensional space with the noise injected by our method in a k-dimensional subspace, where $k \ll d$. Fixing the privacy mechanism parameters, the clipping norm C and the noise multiplier σ, the injected Gaussian noise is isotropic in the update space, with $\eta \sim \mathcal{N}(0, \sigma^2 C^2 I_m)$ for an m-dimensional update. The expected ℓ_2 norm of the noise scales as $\mathbb{E}[\|\eta\|_2] \approx \sigma C\sqrt{m}$.

Thus, when DP-SGD perturbs a d-dimensional update and our method perturbs a k-dimensional update, the ratio of the expected noise magnitudes is:

$$\frac{\mathbb{E}[\|\eta_{\text{ours}}\|_2]}{\mathbb{E}[\|\eta_{\text{DP-SGD}}\|_2]} \approx \sqrt{\frac{k}{d}}. \tag{12}$$

This reduction in the noise magnitude explains our improvements in utility, as the effective noise energy is significantly reduced in the lower-dimensional subspace. A detailed derivation is deferred to Appendix A.3.

Table 1: Results (%) on four GLUE tasks using RoBERTa-base under privacy budgets $\varepsilon \in \{0.5, 1.0\}$. MNLI reports matched (m) and mismatched (mm) accuracy, and QQP reports accuracy (Acc.) and F1. Bold and underline indicate the best and second-best DP results under the same privacy budget.

ε	Method	SST-2	QNLI	MNLI		QQP	
				m	mm	Acc.	F1
–	Full-FT	94.07 ± 0.25	92.21 ± 0.11	86.45 ± 0.18	86.95 ± 0.22	91.43 ± 0.09	88.32 ± 0.14
0.5	DP-Full	90.64 ± 0.31	$\underline{86.59 \pm 0.15}$	81.35 ± 0.19	80.09 ± 0.31	82.36 ± 0.17	$\underline{79.36 \pm 0.12}$
	DP-LoRA	$\underline{90.98 \pm 0.29}$	85.66 ± 0.38	$\underline{81.38 \pm 0.21}$	$\underline{80.49 \pm 0.14}$	$\underline{83.61 \pm 0.37}$	79.18 ± 0.24
	DP-Adapter	89.93 ± 0.44	85.15 ± 0.29	79.23 ± 0.15	79.79 ± 0.25	81.05 ± 0.09	77.45 ± 0.10
	DP-BitFit	88.92 ± 0.68	84.28 ± 1.07	77.96 ± 0.29	78.82 ± 0.21	80.71 ± 0.14	77.98 ± 0.19
	Ours	$\mathbf{91.76 \pm 0.34}$	$\mathbf{88.19 \pm 0.18}$	$\mathbf{83.19 \pm 0.09}$	$\mathbf{83.56 \pm 0.18}$	$\mathbf{84.87 \pm 0.14}$	$\mathbf{81.56 \pm 0.12}$
1.0	DP-Full	91.09 ± 0.48	$\underline{86.92 \pm 0.39}$	$\underline{82.76 \pm 0.41}$	$\underline{81.08 \pm 0.19}$	83.15 ± 0.27	80.06 ± 0.18
	DP-LoRA	$\underline{91.15 \pm 0.17}$	86.34 ± 0.46	81.98 ± 0.13	81.02 ± 0.13	$\underline{84.24 \pm 0.41}$	$\underline{80.16 \pm 0.22}$
	DP-Adapter	90.52 ± 0.35	85.98 ± 0.27	80.31 ± 0.16	80.19 ± 0.28	81.87 ± 0.35	78.65 ± 0.16
	DP-BitFit	89.01 ± 0.32	84.64 ± 0.55	78.33 ± 0.20	79.28 ± 0.19	81.18 ± 0.16	78.28 ± 0.15
	Ours	$\mathbf{92.08 \pm 0.29}$	$\mathbf{88.46 \pm 0.21}$	$\mathbf{83.36 \pm 0.08}$	$\mathbf{83.75 \pm 0.17}$	$\mathbf{85.12 \pm 0.27}$	$\mathbf{81.84 \pm 0.11}$

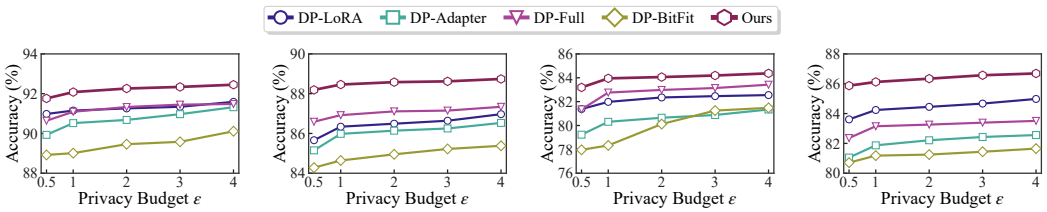
Table 2: ROUGE scores on SAMSum using Qwen2.5-1.5B. R-Ls denotes ROUGE-Lsum. Bold and underlined values indicate the best and second-best DP results, respectively.

Method	$\varepsilon = 0.5$				$\varepsilon = 4.0$			
	R-1	R-2	R-L	R-Ls	R-1	R-2	R-L	R-Ls
Full-FT	0.5044	0.2731	0.4311	0.4319	0.5044	0.2731	0.4311	0.4319
DP-Full	0.4431	0.2076	0.3667	0.3685	0.4547	0.2156	0.3763	<u>0.3793</u>
DP-LoRA (rank=32)	<u>0.4513</u>	<u>0.2177</u>	0.3704	0.3712	0.4565	0.2147	0.3757	0.3655
DP-Adapter($r = 48$)	0.4437	0.2115	<u>0.3735</u>	<u>0.3717</u>	<u>0.4573</u>	<u>0.2213</u>	<u>0.3839</u>	0.3709
DP-BitFit	0.4475	0.2033	0.3685	0.3715	0.4535	0.2197	0.3758	0.3764
Ours (LoRA rank=32)	**0.4653**	**0.2289**	**0.3841**	**0.3852**	**0.4684**	**0.2311**	**0.3917**	**0.3879**

5 EXPERIMENTS

Our experiments are conducted on 6 NVIDIA RTX 5090 GPUs. We evaluate the proposed method across natural language understanding and summarization tasks using **RoBERTa-base**, **RoBERTa-large** Liu et al. (2019), and the base, non-instruction-tuned **Qwen2.5-1.5B** Yang et al. (2025) as backbone models. The evaluation covers **SST-2**, **QNLI**, **MNLI**, **QQP** Wang et al. (2018), **IMDB** Maas et al. (2011), and **SAMSum** Gliwa et al. (2019), with accuracy reported for classification tasks and ROUGE scores for summarization. We compare against the non-private **Full-FT** upper bound and representative DP fine-tuning baselines, including **DP-Full**, **DP-LoRA**, and **DP-Adapter** Yu et al. (2022), as well as **DP-BitFit** Bu et al. (2024). Detailed descriptions of datasets and backbone models, task-specific hyperparameters, baseline configurations, privacy parameters, and implementation details are provided in Appendices C, D.1, and D.3.

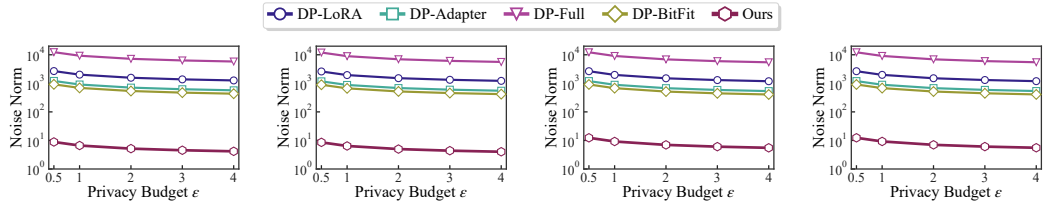
Main Results. Table 1 presents the results on four GLUE tasks under two privacy budgets ($\varepsilon \in 0.5, 1.0$). Across all tasks and both privacy budgets, our method consistently achieves the best performance among the DP baselines across all reported metrics, demonstrating improved utility under the same privacy guarantees. The gains are generally more pronounced under the stricter privacy constraint ($\varepsilon = 0.5$). In particular, on MNLI, where the competing DP methods exhibit substantial performance degradation, our method considerably narrows the gap to non-private full fine-tuning. To evaluate its applicability beyond NLU tasks, Table 2 reports results on the SAMSum generation task using Qwen2.5-1.5B, where our method also consistently outperforms the DP fine-tuning baselines under both privacy budgets. Additional results on RoBERTa-large are provided in Appendix E.1.



(a) SST-2 (b) QNLI (c) MNLI (d) QQP

Figure 2: Task performance under different privacy budgets across four GLUE tasks.

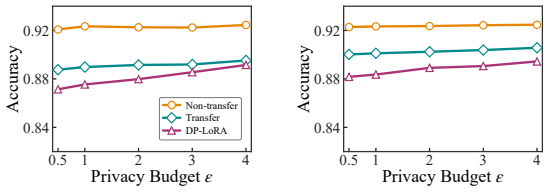
Impact of Privacy Budgets. To verify the effectiveness of our method across different privacy budgets, Fig. 2 plots the accuracy as ε increases from 0.5 to 4.0. Across all four tasks and throughout the entire range of privacy budgets, our method consistently achieves the highest accuracy among the DP baselines. Under stronger privacy constraints, baseline methods exhibit substantial performance degradation, while our method remains notably more stable, demonstrating the stability of our method in maintaining high accuracy even under stricter privacy budgets.



(a) SST-2 (b) QNLI (c) MNLI (d) QQP

Figure 3: Expected ℓ_2 norm of injected DP noise under different privacy budgets across four GLUE tasks. The y-axis is shown on a logarithmic scale.

Noise Norm Analysis. To explain the trends shown in Fig. 3, we analyze the expected magnitude of the differential privacy perturbations injected during training. Fig. 3 plots the expected noise norm as a function of the privacy budget ε, with the y-axis in log scale. For all methods, the expected noise norm decreases as ε increases, which is consistent with the intuition that weaker privacy constraints require less perturbation. Across all four tasks, our method consistently yields smaller noise norms than the baselines. This gap is most pronounced under stricter privacy budgets (smaller ε), indicating that our updates are less dominated by differential privacy noise. Taken together with the results in Table 1 and Fig. 2, these findings are consistent with the dimensional analysis in Section 4.4, suggesting that the reduced effective magnitude of injected DP noise contribute to the improved utility of our method under strong privacy constraints.



(a) From SST-2 to IMDB (b) From IMDB to SST-2

Figure 4: Accuracy under bidirectional cross-task subspace transfer between SST-2 and IMDB.

Table 3: Ablation on subspace dimension r_{sub}. Bold values denote the best results.

ε	r_{sub}	SST-2	QNLI	MNLI		QQP	
				m	mm	Acc.	F1
0.5	16	91.45	88.11	83.02	83.11	84.31	81.24
	32	91.76	88.19	83.19	**83.56**	84.87	81.56
	64	**92.06**	**88.50**	**83.36**	83.47	**85.09**	**81.68**
1.0	16	91.97	88.23	83.45	83.29	85.04	81.67
	32	92.08	88.46	83.36	83.75	85.12	81.84
	64	**92.35**	**88.58**	**84.17**	**83.92**	**85.21**	**82.09**

Ablation Study. Table 3 investigates the effect of the subspace dimension r_{sub}. Overall, performance improves as the subspace dimension increases, with $r_{\text{sub}} = 64$ achieving the strongest results

Table 4: Ablation study on subspace construction using RoBERTa-base under $\varepsilon = 1.0$. From-Init (Ours) constructs the rank-32 subspace from parameter displacements relative to initialization, Step-Delta uses parameter differences between consecutive snapshots, and Random uses a randomly generated rank-32 subspace. Results are averaged over three random seeds. Bold and underline indicate the best and second-best results, respectively.

Method	SST-2	QNLI	MNLI		QQP	
			m	mm	Acc.	F1
Random	49.69 ± 0.86	49.82 ± 0.51	32.33 ± 0.15	34.48 ± 0.24	64.52 ± 0.36	2.24 ± 0.59
Step-Delta	$\underline{91.51 \pm 0.32}$	$\underline{87.71 \pm 0.13}$	$\underline{83.25 \pm 0.11}$	$\underline{82.89 \pm 0.06}$	$\underline{84.68 \pm 0.22}$	$\underline{81.15 \pm 0.14}$
From-Init (Ours)	$\mathbf{92.08 \pm 0.29}$	$\mathbf{88.46 \pm 0.21}$	$\mathbf{83.36 \pm 0.08}$	$\mathbf{83.75 \pm 0.17}$	$\mathbf{85.12 \pm 0.27}$	$\mathbf{81.84 \pm 0.11}$

on nearly all metrics under both privacy budgets. The improvement is especially clear relative to $r_{\text{sub}} = 16$, indicating that overly restrictive subspaces may discard task-relevant update directions and limit the effectiveness of private optimization. While increasing r_{sub} also enlarges the space in which DP optimization is performed, the observed gains suggest that, within the evaluated range, retaining a richer set of informative directions provides a more favorable trade-off between subspace capacity and DP perturbation.

Table 4 further examines how the subspace itself is constructed. From-Init consistently achieves the best performance across all reported metrics, while Step-Delta provides the second-best results and substantially outperforms Random. The severe degradation under Random shows that dimensionality reduction alone is insufficient: the learned subspace must capture directions that are informative for downstream adaptation. Moreover, the consistent advantage of From-Init over Step-Delta indicates that displacements accumulated relative to initialization capture the global structure of the Stage-1 optimization trajectory more effectively than local differences between consecutive snapshots. Together, these results highlight that both the capacity and the construction of the learned subspace are important for effective private fine-tuning. Additional sensitivity results on the LoRA rank are provided in Appendix E.2.

Cross-Task Subspace Transfer. We evaluate whether a subspace learned from a similar source task can be transferred to private fine-tuning on a related target task using bidirectional transfer between SST-2 and IMDB. Since the two datasets differ substantially in size, we subsample the same number of examples from each dataset to ensure a comparable training protocol. For the transferred-subspace setting, the subspace is constructed solely from the non-private warm-up trajectory on the source task and is then fixed throughout private fine-tuning on the target task. For a fair comparison, DP-LoRA follows the same cross-task warm-up protocol before private training, but does not impose any subspace constraint. Fig. 4 compares DP-LoRA, our method using a subspace constructed from the target task itself, and our method using a subspace transferred from the other task.

Across all privacy budgets, the transferred subspace consistently outperforms DP-LoRA, while remaining slightly below the target-task subspace. These results show that task-related datasets can provide reusable subspace information that improves private fine-tuning even when the subspace is learned from a different task.

6 Conclusion

In this paper, we propose a two-stage framework for differentially private fine-tuning in learned subspaces of LoRA parameters. By restricting private optimization to low-dimensional subspaces, our method reduces the effective impact of differential privacy noise while preserving formal privacy guarantees. We provide a theoretical interpretation showing that subspace optimization is equivalent to gradient projection–based optimization and leads to reduced noise magnitude. Extensive experiments on multiple benchmarks demonstrate consistent improvements over existing DP baselines, especially under strict privacy budgets. We further show that learned subspaces can transfer across datasets from related task, enabling practical private fine-tuning when target trajectories are unavailable.

AI USE STATEMENT

We used large language models (LLMs) during the preparation of this manuscript for writing and editorial assistance, such as polishing sentences, checking grammar, improving presentation and readability, and reviewing the consistency of theoretical reasoning and mathematical notation. LLMs were not used to generate the original research ideas or to determine the methodological or experimental design. All research, experiments, and analyses were carried out by the authors. Any AI-assisted content was subsequently reviewed and revised by the authors for accuracy and alignment with the actual research. The authors assume full responsibility for the final content of this work, including its text, claims, analyses, and AI-assisted artifacts.

REPRODUCIBILITY STATEMENT

We provide sufficient details to facilitate reproduction of our experiments. Descriptions of all datasets and backbone models are given in Appendix C, while Appendix D reports the task-specific hyperparameters, privacy accounting procedure, baseline configurations, and evaluation settings. The training procedure is specified in the main text and accompanying algorithms. The implementation code has been released through an anonymous repository, with the corresponding link provided in the abstract.

ETHICS STATEMENT

We conduct this research in accordance with the ICLR Code of Ethics. Our experiments rely exclusively on existing public NLP benchmarks and do not require the collection of new personal data or interaction with human participants. The datasets used in this work, including SST-2, QNLI, MNLI, QQP, IMDB, and SAMSum, are employed only to evaluate differentially private fine-tuning under controlled experimental settings. The privacy objective of our method is to bound the influence of individual examples in the designated private training subset through formal differential privacy guarantees. This work does not conduct privacy attacks, attempt to infer the identity of dataset contributors, or seek to extract information associated with individual records.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pp. 308–318, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145/2976749.2978318.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7319–7328, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.568. URL https://aclanthology.org/2021.acl-long.568/.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/3b5020bb891119b9f5130f1fea9bd773-Paper.pdf.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, Dublin, Ireland, May

2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.1. URL https://aclanthology.org/2022.acl-short.1/.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Differentially private bias-term fine-tuning of foundation models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4730–4751. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/bu24c.html.

Jiashun Cheng, Aochuan Chen, Nuo Chen, Ziqi Gao, Yuhan Li, Jia Li, and Fugee Tsung. Revisiting LoRA through the lens of parameter redundancy: Spectral encoding helps. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 2701–2718, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.139. URL https://aclanthology.org/2025.findings-acl.139/.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023. URL https://arxiv.org/abs/2305.14314.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5 (3):220–235, 2023. doi: 10.1038/s42256-023-00626-4. URL https://doi.org/10.1038/s42256-023-00626-4.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin (eds.), *Theory of Cryptography*, pp. 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. Practical membership inference attacks against fine-tuned large language models via self-prompt calibration, 2024. URL https://arxiv.org/abs/2311.06062.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu (eds.), *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5409. URL https://aclanthology.org/D19-5409/.

Xin Gu, Gautam Kamath, and Steven Wu. Choosing public datasets for private machine learning via gradient subspace distance. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022. URL https://openreview.net/forum?id=zr6AZ8ARan.

Jiyan He, Xuechen Li, Da Yu, Huishuai Zhang, Janardhan Kulkarni, Yin Tat Lee, Arturs Backurs, Nenghai Yu, and Jiang Bian. Exploring the limits of differentially private deep learning with group-wise clipping. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=oze0clVGPeX.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

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, 2021. URL https://arxiv.org/abs/2106.09685.

Peter Kairouz, Monica Ribero Diaz, Keith Rush, and Abhradeep Thakurta. (nearly) dimension independent private erm with adagrad rates via publicly estimated subspaces. In Mikhail Belkin and Samory Kpotufe (eds.), *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pp. 2717–2746. PMLR, 15–19 Aug 2021. URL https://proceedings.mlr.press/v134/kairouz21a.html.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes, 2018. URL https://arxiv.org/abs/1804.08838.

Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners, 2022. URL https://arxiv.org/abs/2110.05679.

Hongbin Liu, Lun Wang, Om Thakkar, Abhradeep Thakurta, and Arun Narayanan. Differentially private parameter-efficient fine-tuning for large asr models, 2024. URL https://arxiv.org/abs/2410.01948.

Xiao-Yang Liu, Rongyi Zhu, Daochen Zha, Jiechao Gao, Shan Zhong, Matt White, and Meikang Qiu. Differentially private low-rank adaptation of large language model using federated learning. *ACM Trans. Manage. Inf. Syst.*, 16(2), March 2025. ISSN 2158-656X. doi: 10.1145/3682068. URL https://doi.org/10.1145/3682068.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (eds.), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015/.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, 2017. doi: 10.1109/CSF.2017.11.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne,

Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael (Rai) Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Haichao Sha, Yuncheng Wu, Ruixuan Liu, Yang Cao, and Hong Chen. Differentially private visual learning with public subspace augmented by synthetic data. In Cathal Gurrin, Klaus Schoeffmann, Min Zhang, Luca Rossetto, Stevan Rudinac, Duc-Tien Dang-Nguyen, Wen-Huang Cheng, Phoebe Chen, and Jenny Benois-Pineau (eds.), *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pp. 11357–11366. ACM, 2025. doi: 10.1145/3746027.3755037. URL https://doi.org/10.1145/3746027.3755037.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170/.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Tal Linzen, Grzegorz Chrupała, and Afra Alishahi (eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446/.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101/.

Qwen : An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi

Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=7aogOj_VYO0.

Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Large scale private learning via low-rank reparametrization, 2021b. URL https://arxiv.org/abs/2106.09352.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Q42f0dfjECO.

Juzheng Zhang, Jiacheng You, Ashwinee Panda, and Tom Goldstein. Lori: Reducing cross-task interference in multi-task low-rank adaptation, 2025. URL https://arxiv.org/abs/2504.07448.

Zhong Zhang, Bang Liu, and Junming Shao. Fine-tuning happens in tiny subspaces: Exploring intrinsic task-specific subspaces of pre-trained language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1701–1713, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.95. URL https://aclanthology.org/2023.acl-long.95/.

Lele Zheng, Xiang Wang, Tao Zhang, Yang Cao, Ke Cheng, and Yulong Shen. Differentially private subspace fine-tuning for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28830–28838, 2026. doi: 10.1609/aaai.v40i34.40117.

Tianyuan Zou, Zinan Lin, Sivakanth Gopi, Yang Liu, Ya-Qin Zhang, Robert Sim, Xin Deng, and Sergey Yekhanin. PE-SGD: Differentially private deep learning via evolution of gradient subspace for text. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=713ywmTZHv.

A Theoretical Analysis

A.1 Derivation of Implicit Gradient Projection

Let $W^{(t)} = P\theta^{(t)}$ denote the value of W at iteration t. The gradient descent update on the subspace parameter θ is

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta L(P\theta^{(t)}, x). \tag{13}$$

By the chain rule and the reparameterization $W = P\theta$, we have

$$\nabla_\theta L(P\theta, x) = P^\top \nabla_W L(W, x)\big|_{W=P\theta}. \tag{14}$$

Substituting this relation into the update of θ gives

$$\theta^{(t+1)} = \theta^{(t)} - \eta P^\top \nabla_W L(W^{(t)}, x). \tag{15}$$

Multiplying both sides by P and using $W^{(t)} = P\theta^{(t)}$, we obtain

$$\begin{aligned} W^{(t+1)} &= P\theta^{(t+1)} \\ &= W^{(t)} - \eta P P^\top \nabla_W L(W^{(t)}, x). \end{aligned} \tag{16}$$

Since P has orthonormal columns, $PP^\top$ is the orthogonal projection onto $\mathrm{span}(P)$. Therefore, optimizing the low-dimensional parameter θ implicitly projects the gradient update of W onto $\mathrm{span}(P)$, without requiring an explicit projection operation during training.

A.2 Privacy Guarantee

We consider the public-data-assisted setting adopted throughout this work. Under our threat model, the auxiliary dataset $\mathcal{D}_{\text{aux}}$ is assumed to be publicly available or otherwise non-sensitive, while $\mathcal{D}_{\text{priv}}$ contains the records to be protected. The two datasets are disjoint, and Stage 1 accesses only $\mathcal{D}_{\text{aux}}$. Therefore, Stage 1 does not access any record in $\mathcal{D}_{\text{priv}}$ and incurs no privacy cost with respect to $\mathcal{D}_{\text{priv}}$. The projection bases obtained in Stage 1 are fixed before private fine-tuning begins.

In Stage 2, differentially private optimization is performed exclusively on $\mathcal{D}_{\text{priv}}$. For each sampled training example, the gradients of all trainable subspace parameters are concatenated and jointly clipped to have ℓ_2 norm at most C. The clipped per-example gradients are then aggregated over a minibatch and perturbed with Gaussian noise according to the standard DP-SGD mechanism Abadi et al. (2016). Given the sampling rate q, noise multiplier σ, number of optimization steps T, and target failure probability δ, a standard privacy accountant is used to determine the resulting privacy loss ε. Hence, the complete Stage 2 optimization satisfies (ε, δ)-differential privacy with respect to $\mathcal{D}_{\text{priv}}$.

Because the projection bases are fixed independently of $\mathcal{D}_{\text{priv}}$, the reparameterization used during Stage 2 does not introduce any additional access to private data beyond the DP-SGD mechanism. Moreover, reconstructing the LoRA factors from the optimized subspace parameters, e.g., $A = P_A\theta_A$ and $B = P_B\theta_B$, is a deterministic transformation of the differentially private output. By Theorem 2, this reconstruction incurs no additional privacy loss.

Therefore, the complete framework satisfies (ε, δ)-differential privacy with respect to $\mathcal{D}_{\text{priv}}$.

A.3 Analysis of Noise Error

Noise magnitude under isotropic Gaussian perturbations. Let $\eta \sim \mathcal{N}(0, \sigma^2 C^2 I_m)$ denote the DP noise added to an m-dimensional update, where C is the clipping norm and σ is the noise multiplier. Then $\|\eta\|_2/(\sigma C)$ follows a χ distribution with m degrees of freedom, and

$$\mathbb{E}[\|\eta\|_2] = \sigma C \cdot \mathbb{E}[\chi_m] = \sigma C \cdot \sqrt{2}\,\frac{\Gamma\left(\frac{m+1}{2}\right)}{\Gamma\left(\frac{m}{2}\right)} \approx \sigma C\sqrt{m}, \tag{17}$$

where the approximation uses the standard asymptotic of $\mathbb{E}[\chi_m]$ for large m.

Derivation of Eq. 12. In standard DP-SGD, $m = d$, while in our subspace optimization stage, $m = k$. Assuming the same noise multiplier σ, clipping norm C, and sampling scheme, we obtain

$$\frac{\mathbb{E}[\|\eta_{\text{ours}}\|_2]}{\mathbb{E}[\|\eta_{\text{DP-SGD}}\|_2]} \approx \frac{\sigma C\sqrt{k}}{\sigma C\sqrt{d}} = \sqrt{\frac{k}{d}}. \tag{18}$$

Discussion. The above comparison isolates the dimensional effect of injecting isotropic Gaussian noise in the update space. Under the same privacy accounting configuration, the noise multiplier is determined independently of the update dimension, so the difference in the expected noise norm is governed by the dimensionality of the optimization space.

B The Algorithm of the Proposed Method

B.1 Stage 1: Subspace Construction

Algorithm 1 outlines the procedure for constructing the subspaces during Stage 1 using the optimization trajectories of the LoRA factors on the public auxiliary dataset. The algorithm takes as inputs a frozen backbone model Θ, trainable LoRA factors (A, B) with dimensions d_A and d_B, subspace dimension r, number of snapshots T, public auxiliary dataset $\mathcal{D}_{\text{aux}}$, learning rate η, loss function $\mathcal{L}$, number of Stage 1 training steps T_1, and batch size $|\mathcal{B}_t|$. During Stage 1, the backbone parameters remain frozen while the LoRA factors (A, B) are optimized on $\mathcal{D}_{\text{aux}}$ using standard gradient-based optimization. Since this stage accesses only public auxiliary data and does not use the private training subset $\mathcal{D}_{\text{priv}}$, it incurs no privacy cost.

Algorithm 1 Subspace Construction.

Require: Frozen backbone model Θ, trainable LoRA factors (A, B) with dimensions d_A and d_B, subspace dimension r, number of snapshots T, public auxiliary dataset $\mathcal{D}_{\text{aux}}$, learning rate η, loss $\mathcal{L}$, Stage 1 steps T_1, and batch size $|\mathcal{B}_t|$.
Ensure: Trained LoRA factors A_f and B_f, projection matrices P_A and P_B.

Let $\mathcal{M} \leftarrow \{A, B\}$;
Initialize LoRA on Θ and freeze the backbone; only the low-rank factors A and B are trainable;
Store the initial LoRA factors A_0 and B_0;
Initialize empty trajectory matrices S_A and S_B;
Set the recording period $y \leftarrow \lfloor T_1/T \rfloor$;
for $t = 1$ to T_1 **do**
  Sample a minibatch $\mathcal{B}_t \subset \mathcal{D}_{\text{aux}}$ of size $|\mathcal{B}_t|$;
  $g_t \leftarrow \nabla_{A,B} \frac{1}{|\mathcal{B}_t|} \sum_{x_i \in \mathcal{B}_t} \mathcal{L}(\Theta, A_t, B_t, x_i)$;
  $A_{t+1}, B_{t+1} \leftarrow Adam(A_t, B_t, g_t, \eta)$;
  if $t \bmod y = 0$ **then**
    for $M \in \mathcal{M}$ **do**
      $\bar{m} \leftarrow \text{flat}(M_t - M_0)$;
      Append $\bar{m}$ to S_M;
    end for
  end if
end for
$P_A, P_B \leftarrow \text{SVD}(S_A, S_B, r)$;
$A_f \leftarrow A_{T_1}, \quad B_f \leftarrow B_{T_1}$;
return A_f, B_f, P_A, P_B;

During optimization, the algorithm records T snapshots at regular intervals, specifically every $y = \lfloor T_1/T \rfloor$ steps. At each snapshot, it computes the displacements of (A, B) from their initial values, flattens these displacements into vectors, and appends them to the trajectory matrices S_A and S_B. After Stage 1 training is complete, Singular Value Decomposition (SVD) is applied to the trajectory matrices (S_A, S_B), and the top-r right singular vectors are extracted to form the orthonormal projection bases (P_A, P_B). The final Stage 1 LoRA factors (A_f, B_f) are also returned for subsequent initialization.

B.2 Stage 2: Private Subspace Fine-tuning

Algorithm 2 outlines the procedure for the second stage of fine-tuning, which performs differential privacy (DP) optimization within the learned low-dimensional subspaces. The algorithm accepts as inputs Model parameters Θ, trained LoRA factors A_f, B_f, projection matrices P_A, P_B with orthonormal columns, subspace dimension r, training set $\mathcal{D}$, learning rate η, loss function $\mathcal{L}$, training steps T_2, batch size $|\mathcal{B}_t|$, clipping threshold C, and privacy parameters (ε, δ). The algorithm first initializes the subspace parameters by projecting the flattened LoRA factors onto the respective bases:

$$\theta_A \leftarrow P_A^\top \cdot \text{flat}(A_f), \quad \theta_B \leftarrow P_B^\top \cdot \text{flat}(B_f). \tag{19}$$

For each iteration, the algorithm computes per-example gradients with respect to the subspace parameters (θ_A, θ_B). These per-example gradients are then concatenated into a single joint vector $g_t^\theta(x_i) = [g_t^{\theta_A}(x_i), g_t^{\theta_B}(x_i)]$, which is clipped using an ℓ_2 norm with a threshold C. This clipping step ensures that the total gradient contribution across both subspaces is bounded by C, preventing potential inflation that could occur if each subspace were clipped independently.

After clipping, the gradients are split back into their respective components, averaged over the minibatch, and then perturbed with Gaussian noise. The noise is scaled by σC, where σ is the noise multiplier computed by the RDP accountant. The perturbed gradients are then used to update the subspace parameters (θ_A, θ_B) using the Adam optimizer.

Finally, the LoRA factors are reconstructed by reshaping the subspace parameters back into their original dimensions by $A^* \leftarrow \text{reshape}(P_A \theta_A), \quad B^* \leftarrow \text{reshape}(P_B \theta_B)$. These updated LoRA factors (A^*, B^*) are returned as the output.

Algorithm 2 Private Subspace Fine-tuning.

Require: Model parameters Θ, trained LoRA factors A_f, B_f, projection matrices P_A, P_B with orthonormal columns, subspace dimension r, training set $\mathcal{D}$, learning rate η, loss function $\mathcal{L}$, training steps T_2, batch size $|\mathcal{B}_t|$, clipping threshold C, and privacy parameters (ε, δ).

Ensure: Updated LoRA factors (A^*, B^*).

Let $\mathcal{M} \leftarrow \{A, B\}$;
$\sigma \leftarrow rdp\left(\varepsilon, \delta, \frac{|\mathcal{B}_t|}{|\mathcal{D}|}, T_2\right)$;
Initialize $\theta_A = P_A^\top \cdot \text{flat}(A_f)$, $\theta_B = P_B^\top \cdot \text{flat}(B_f)$;
for $t = 1$ to T_2 **do**
  Sample a minibatch $\mathcal{B}_t \subset \mathcal{D}$ of size $|\mathcal{B}_t|$;
  for x_i in $\mathcal{B}_t$ **do**
    for $M \in \mathcal{M}$ **do**
      $g_t^{\theta_M}(x_i) = \nabla_{\theta_M} \mathcal{L}(P_A\theta_A, P_B\theta_B, x_i)$;
    end for
    $g_t^{\theta}(x_i) = \left[g_t^{\theta_A}(x_i),\ g_t^{\theta_B}(x_i)\right]$;
    $\overline{g_t^{\theta}}(x_i) = g_t^{\theta}(x_i) \cdot \min\left(1, \frac{C}{\|g_t^{\theta}(x_i)\|_2}\right)$
    $\left[\overline{g_t^{\theta_A}}(x_i),\ \overline{g_t^{\theta_B}}(x_i)\right] \leftarrow \overline{g_t^{\theta}}(x_i)$
  end for
  for $M \in \mathcal{M}$ **do**
    $\bar{g}_t^{\theta_M} \leftarrow \frac{1}{|\mathcal{B}_t|} \sum_{x_i \in \mathcal{B}_t} \overline{g_t^{\theta_M}}(x_i)$;
    $\widetilde{g}_t^{\theta_M} \leftarrow \bar{g}_t^{\theta_M} + \frac{1}{|\mathcal{B}_t|}\mathcal{N}(0, \sigma^2 C^2 I_r)$;
  end for
  $\theta_A^{t+1}, \theta_B^{t+1} \leftarrow Adam(\theta_A^t, \theta_B^t, \widetilde{g}_t^{\theta_A}, \widetilde{g}_t^{\theta_B}, \eta)$;
end for
$A^* \leftarrow \text{reshape}(P_A\theta_A), \quad B^* \leftarrow \text{reshape}(P_B\theta_B)$;
return A^*, B^*;

C Details of Datasets and Backbone Models

C.1 Backbone Models

RoBERTa-base and RoBERTa-large. RoBERTa Liu et al. (2019) is a Transformer-based masked language model pretrained with an optimized BERT-style training procedure. In our natural language understanding experiments, we use **RoBERTa-base** and **RoBERTa-large** as encoder-only backbone models for classification tasks, including SST-2, QNLI, MNLI, QQP, and IMDB. These two model scales allow us to evaluate whether the proposed method remains effective across different backbone capacities. For RoBERTa-based experiments, the task-specific classification head is frozen, while the remaining pretrained parameters are treated according to the corresponding method configuration.

Qwen2.5-1.5B. Qwen2.5-1.5B Yang et al. (2025) is a decoder-only causal language model from the Qwen2.5 model family. We use the base, non-instruction-tuned version of **Qwen2.5-1.5B** as the backbone model for the SAMSum dialogue summarization experiments. This setting evaluates whether the proposed private fine-tuning framework extends beyond encoder-only classification tasks to generative language modeling. For SAMSum, we report ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum, using the same generation configuration across all compared methods.

C.2 Datasets

SST-2. The Stanford Sentiment Treebank (SST-2) dataset Socher et al. (2013) is a binary sentiment classification dataset consisting of movie reviews. It has 67,349 training samples and 872 validation samples. Each review is labeled as either "positive" or "negative." The dataset is commonly used for evaluating sentiment analysis models. The maximum sequence length is set to 128 tokens.

QNLI. The Question Natural Language Inference (QNLI) dataset is a part of the GLUE benchmark Wang et al. (2018), where the task is to classify whether a given pair of sentences contains a natural language inference. It has 104,743 training examples and 5,463 validation examples. Each example consists of a question and a sentence, with labels indicating whether the sentence is an answer to the question. The maximum sequence length is set to 128 tokens.

MNLI. The Multi-Genre Natural Language Inference (MNLI) dataset Williams et al. (2018) is designed for the task of recognizing textual entailment across different genres. The dataset contains 392,702 training examples and 9,815 validation examples (matched). Each example consists of a premise and a hypothesis sentence, with labels indicating whether the hypothesis is entailed, contradicted, or neutral with respect to the premise. The maximum sequence length is set to 256 tokens.

QQP. The Quora Question Pairs (QQP) dataset Wang et al. (2018) is used for the task of identifying duplicate questions. The dataset consists of question pairs, with 363,846 for training and 40,430 for validation. Each question pair is labeled as either a duplicate or non-duplicate. The maximum sequence length is set to 128 tokens.

IMDB. The IMDB sentiment dataset Maas et al. (2011) consists of 50,000 movie reviews for sentiment classification. The dataset is balanced, with 25,000 positive and 25,000 negative reviews, and is commonly used for evaluating sentiment analysis models. The dataset is split into 25,000 training samples and 25,000 test samples. The maximum sequence length is set to 256 tokens.

SAMSum. The SAMSum dataset Gliwa et al. (2019) is an abstractive dialogue summarization benchmark consisting of messenger-like conversations paired with human-written summaries. It contains over 16k dialogues and is used in our experiments to evaluate Qwen2.5-1.5B on a generative task under differential privacy. We report ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum as evaluation metrics. The maximum input length is set to 512 tokens, and the maximum number of newly generated tokens during evaluation is set to 256.

D Additional Implementation Details of Experiments

D.1 Hyperparameter Settings

This section summarizes the task-specific hyperparameters and evaluation settings used in our experiments. Classification tasks and generation tasks are reported separately because they involve different evaluation protocols.

Task	Max Length	η	C	δ
SST-2	128	5×10^{-4}	0.1	10^{-5}
QNLI	128	5×10^{-4}	0.1	10^{-5}
IMDB	256	5×10^{-4}	0.1	10^{-5}
QQP	128	5×10^{-4}	0.5	10^{-6}
MNLI	256	5×10^{-4}	0.5	10^{-6}

Table 5: Task-wise hyperparameter settings for classification tasks.

Task	Model	Max Input	Max New Tokens	η	C	δ
SAMSum	Qwen2.5-1.5B	512	256	5×10^{-4}	1.0	10^{-5}

Table 6: Hyperparameter settings for the SAMSum summarization task. Max New Tokens denotes the maximum number of newly generated tokens during evaluation.

Tables 5 and 6 show the evaluation split for each task along with the corresponding hyperparameters. The learning rate η controls the step size in optimization, C is the gradient norm bound, and δ is the additive slack term in (ε, δ)-DP, bounding the probability of privacy failure.

D.2 Privacy Accounting

We follow the standard DP-SGD pipeline provided by Opacus, including per-sample gradient clipping with norm C and Gaussian noise calibrated by the RDP accountant.

D.3 Baseline Configurations

DP-Full. We fine-tune all model parameters under DP-SGD with the same privacy accountant, clipping norm, batch size, optimizer, and number of training epochs as the other DP methods.

DP-LoRA. We insert LoRA modules into `query`, `key`, `value`, and the attention output projection (`dense`) in each Transformer layer, and optimize only the LoRA parameters under DP-SGD while keeping the backbone frozen.

DP-Adapter. We use Houlsby-style adapters and train only the adapter parameters under DP-SGD. Concretely, we adopt `AdapterConfig.load("houlsby")` with ReLU non-linearity, pre-adapter layer normalization (`ln_before=True`), and no post-adapter layer normalization (`ln_after=False`).

DP-BitFit. Following Bu et al. (2024), we train only the bias terms under DP-SGD while keeping all non-bias backbone parameters frozen. For classification tasks, the task-specific classification head is also trained under DP-SGD. The privacy accountant, clipping norm, batch size, optimizer, and number of training epochs are kept the same as in the other DP methods.

E Additional Results

E.1 Results on RoBERTa-Large

Table 7 evaluates scalability on the larger RoBERTa-large backbone. Under various privacy budgets ($\varepsilon \in \{0.5, 4.0\}$), our method consistently outperforms DP-LoRA and DP-Adapter across all four tasks, indicating that the proposed subspace parameterization remains effective.

Table 7: Results (%) on RoBERTa-large under $\varepsilon \in \{0.5, 4.0\}$. MNLI reports matched (m) and mismatched (mm) accuracy, and QQP reports accuracy (Acc.) and F1. Bold and underline indicate the best and second-best DP results under the same privacy budget.

ε	Method	SST-2	QNLI	MNLI		QQP	
				m	mm	Acc.	F1
0.5	DP-Full	91.18 ± 0.16	87.61 ± 0.68	$\underline{84.59 \pm 0.12}$	84.58 ± 0.09	$\underline{84.65 \pm 0.33}$	$\underline{82.13 \pm 0.06}$
	DP-LoRA	92.21 ± 0.19	$\underline{87.95 \pm 0.46}$	84.14 ± 0.05	$\underline{84.76 \pm 0.13}$	84.42 ± 0.17	81.74 ± 0.16
	DP-Adapter	91.26 ± 0.54	86.52 ± 0.72	83.70 ± 0.23	84.33 ± 0.29	83.81 ± 0.28	80.56 ± 0.32
	DP-BitFit	$\underline{92.28 \pm 0.26}$	86.34 ± 0.86	82.63 ± 0.51	81.77 ± 0.49	82.88 ± 0.47	80.28 ± 0.96
	Ours	$\mathbf{93.46 \pm 0.57}$	$\mathbf{90.69 \pm 0.68}$	$\mathbf{87.34 \pm 0.36}$	$\mathbf{87.34 \pm 0.21}$	$\mathbf{86.39 \pm 0.14}$	$\mathbf{83.41 \pm 0.23}$
4.0	DP-Full	92.44 ± 0.14	$\underline{88.52 \pm 0.38}$	$\underline{85.63 \pm 0.33}$	84.61 ± 0.38	$\underline{85.06 \pm 0.29}$	$\underline{82.65 \pm 0.08}$
	DP-LoRA	$\underline{93.06 \pm 0.32}$	88.37 ± 1.11	85.48 ± 0.09	$\underline{85.73 \pm 0.31}$	84.81 ± 0.12	82.09 ± 0.35
	DP-Adapter	92.59 ± 0.21	86.94 ± 0.51	84.51 ± 0.16	84.27 ± 0.14	84.74 ± 0.26	81.18 ± 0.11
	DP-BitFit	92.35 ± 0.42	87.65 ± 0.43	83.54 ± 0.62	82.94 ± 0.19	83.67 ± 0.31	81.95 ± 0.46
	Ours	$\mathbf{93.89 \pm 0.15}$	$\mathbf{91.55 \pm 0.31}$	$\mathbf{88.04 \pm 0.13}$	$\mathbf{87.52 \pm 0.20}$	$\mathbf{87.26 \pm 0.13}$	$\mathbf{83.49 \pm 0.16}$

E.2 Sensitivity to LoRA Rank

Table 8 evaluates the sensitivity of our method to the LoRA rank. The main experiments use rank 32, while here we consider rank 16 and rank 64 under the same training protocol. Across both privacy budgets and all evaluation metrics, rank 64 consistently outperforms rank 16, indicating that increasing the LoRA capacity can provide additional task-relevant flexibility during private fine-tuning. However, the improvements are relatively modest, ranging from 0.31 to 0.91 percentage

Table 8: Sensitivity to the LoRA rank using RoBERTa-base. The main experiments use rank 32, while rank 16 and rank 64 are evaluated here under the same training protocol. MNLI reports matched (m) and mismatched (mm) accuracy, and QQP reports accuracy (Acc.) and F1. Results are averaged over three random seeds and reported as mean $\pm$ standard deviation.

ε	Rank	SST-2	QNLI	MNLI		QQP	
				m	mm	Acc.	F1
0.5	16	91.38 ± 0.52	87.94 ± 0.27	83.13 ± 0.16	83.16 ± 0.22	84.46 ± 0.08	81.25 ± 0.18
	64	91.96 ± 0.42	88.28 ± 0.23	83.54 ± 0.13	83.69 ± 0.21	85.16 ± 0.09	82.07 ± 0.08
1.0	16	92.06 ± 0.38	88.03 ± 0.25	83.56 ± 0.17	83.83 ± 0.26	84.74 ± 0.10	82.33 ± 0.14
	64	92.54 ± 0.33	88.79 ± 0.42	84.15 ± 0.06	84.22 ± 0.09	85.65 ± 0.39	82.64 ± 0.17

points across the reported metrics. These results suggest that our method remains effective to the choice of LoRA rank within the evaluated range, while moderately larger ranks can yield further performance gains.