BASE: BILATERAL SUBSPACE PROJECTION FOR DIFFERENTIALLY PRIVATE LLM FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are commonly adapted to downstream tasks through fine-tuning, where task-specific data often contain sensitive information. Differential privacy (DP) provides rigorous privacy guarantees, but noise injected into high-dimensional update spaces can severely degrade model utility. Existing approaches typically mitigate this issue by restricting optimization to low-dimensional spaces. However, these spaces are typically predefined and may not align well with task-relevant update directions, limiting their effectiveness in private fine-tuning. To address this challenge, we propose BASE, a differentially private bilateral subspace fine-tuning framework that constructs task-aligned low-dimensional update spaces for private fine-tuning. BASE first learns a structured bilateral subspace from public auxiliary data by capturing the principal variation patterns of model updates. This construction is motivated by a gradient energy preservation objective under a Kronecker-structured approximation, leading to a principled construction of input-side and output-side subspaces from update statistics. Private fine-tuning is then constrained within the resulting subspace, preserving informative update directions under DP noise while maintaining computational efficiency. Extensive experiments across diverse models, privacy budgets, and benchmark tasks demonstrate that BASE consistently outperforms strong DP fine-tuning baselines, achieving superior privacy–utility trade-offs while substantially reducing training time and memory overhead. Our code is available at https://anonymous.4open.science/r/ksf-anonymous-2ee2/.

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable performance across a wide range of natural language processing tasks, including text classification, question answering, summarization, and instruction following (Raffel et al., 2020; Brown et al., 2020; Ouyang et al., 2022; Chung et al., 2024). To adapt pretrained models to downstream applications, task-specific fine-tuning is commonly employed (Devlin et al., 2019; Raffel et al., 2020). Fine-tuning allows the models to capture domain-specific patterns and improve performance on both understanding and generation tasks, enabling them to leverage the general knowledge encoded during pretraining while addressing the unique requirements of specific tasks.

However, task-specific fine-tuning often involves sensitive user data, such as medical records, financial transactions, or personal communications, raising significant privacy concerns. To address these risks, differential privacy (DP) has been widely adopted, providing formal guarantees against information leakage during model training. DP is typically enforced through mechanisms such as differentially private stochastic gradient descent (DP-SGD), which clips per-sample gradients and perturbs the aggregated update with Gaussian noise (Abadi et al., 2016). Despite its rigorous privacy guarantees, applying DP to deep model fine-tuning remains highly challenging. When the update space is high-dimensional, as in full-parameter fine-tuning of large language models, the total energy of Gaussian perturbations grows with the dimensionality of the update space. This dimensional dependence can make the noisy update dominate useful gradient information and degrade optimization utility (Yu et al., 2021b). As a result, the informative gradient signal may be severely obscured, causing pronounced utility degradation (Kerrigan et al., 2020; Li et al., 2024).

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.

To mitigate the impact of DP noise, recent approaches constrain optimization to low-dimensional spaces. DP-PEFT methods such as DP-LoRA, DP-Adapter, and DP-BiTFiT reduce the dimension of private optimization by restricting updates to predefined parameter spaces, while such spaces may not align well with task-specific update structure required by different downstream tasks (Yu et al., 2022; Bu et al., 2024). Another line of work, represented by DP-SFT (Zheng et al., 2026), learns task-specific subspaces from optimization trajectories, reducing private optimization dimensionality while constraining updates to task-relevant directions. Meanwhile, DP-SFT relies on global SVD and projection operations in the vectorized full-parameter space, incurring substantial overhead and limiting scalability to large models. These limitations motivate a task-adaptive, module-wise bilateral subspace that captures structured update directions while avoiding costly full-space projections.

To address this challenge, we propose BASE, a framework that constructs task-aligned, low-dimensional update spaces for differentially private fine-tuning. The key idea is to exploit the intrinsic matrix structure of low-rank updates and capture their dominant variation patterns through a bilateral subspace, rather than constructing a subspace over the fully vectorized parameter space. Concretely, BASE decomposes the original high-dimensional update modeling problem into two low-dimensional subspaces along the input and output dimensions, yielding a structured and computationally efficient approximation. These subspaces are estimated using a public auxiliary dataset, so their construction incurs no privacy cost with respect to the private training data (Dwork et al., 2006). During subsequent private optimization, model updates are strictly restricted to the learned bilateral subspace, enabling efficient low-dimensional optimization under differential privacy constraints. In this way, BASE retains the computational advantages of parameter-efficient adaptation while tailoring the optimization space to task-specific update structure. Our contributions are summarized as follows:

- We propose BASE, a two-stage differentially private bilateral subspace fine-tuning framework for large language models. The framework first constructs input-side and output-side subspaces to capture the main directions of task-relevant updates, and then performs private optimization constrained within this low-dimensional subspace, enabling effective fine-tuning under DP constraints.
- We motivate the bilateral construction through a Kronecker-factored gradient-energy objective and derive a separable trajectory-energy surrogate whose optimum is given by the leading eigenspaces of the two side statistics. This establishes a principled connection between the bilateral structure and the public warm-up trajectory used in BASE.
- We conduct extensive experiments across multiple models, tasks, and privacy budgets, demonstrating that BASE consistently outperforms strong DP fine-tuning baselines. Our evaluation highlights its superior privacy–utility trade-offs and substantial reductions in training time and memory overhead, showcasing the practical effectiveness and performance advantages of our method.

2 Related Work

Differentially Private Fine-Tuning. Recent studies on differentially private fine-tuning of language models have increasingly explored low-dimensional update strategies as alternatives to full-parameter DP-SGD. One line of work studies differentially private parameter-efficient fine-tuning (DP-PEFT), which freezes most pretrained parameters and updates only a small set of trainable components. Representative methods include the DP fine-tuning framework of Yu et al. (2022), which adapts parameter-efficient fine-tuning methods such as adapters and LoRA to the private setting, and DP-BiTFiT (Bu et al., 2024), which shows that bias-only private fine-tuning can reduce the time and memory overhead of DP training. These methods inherit the general PEFT principle of restricting adaptation to predefined parameterizations, such as adapters (Houlsby et al., 2019), low-rank adapters (Hu et al., 2022), and bias terms (Ben Zaken et al., 2022).

Another line of work seeks low-dimensional structure directly from gradients or updates. GEP projects private gradients into a non-sensitive anchor subspace and perturbs the low-dimensional embedding and residual separately (Yu et al., 2021a). Large-scale private learning via low-rank reparameterization estimates a principal subspace from historical updates and performs private optimization within the resulting low-rank structure (Yu et al., 2021b). Theoretical results further show that, under suitable subspace assumptions, publicly estimated gradient subspaces can reduce the

ambient-dimension dependence of private empirical risk minimization (Kairouz et al., 2021). Related work also studies how the alignment between public and private gradient subspaces affects the usefulness of public data for private learning (Gu et al., 2022). More recently, DP-SFT constructs task-specific subspaces from optimization trajectories and performs private fine-tuning within the resulting low-dimensional space (Zheng et al., 2026), while PE-SGD dynamically evolves public-data-derived gradient subspaces for differentially private text learning (Zou et al., 2026). Public-subspace-assisted private learning has also been explored in visual domains (Sha et al., 2025). Together, these studies demonstrate the importance of task-relevant low-dimensional structure in differentially private optimization.

Kronecker-Factored Structured Optimization. Kronecker-factored approximations are widely used to make otherwise intractable second-order statistics practical in large-scale deep learning. Representative methods include K-FAC (Martens & Grosse, 2015), its convolutional extension (Grosse & Martens, 2016), and other structured second-order approximations such as block-structured Gauss–Newton methods (Botev et al., 2017). These methods approximate large curvature matrices using structured layerwise factors, improving efficiency while retaining local statistical structure. They are primarily designed for curvature approximation, natural-gradient preconditioning, and optimization acceleration in non-private training.

Existing DP-PEFT methods improve private fine-tuning efficiency through predefined architectural parameterizations, while prior subspace-based private optimization methods learn low-dimensional structures from gradients or updates and typically operate in vectorized parameter or gradient spaces. Kronecker-factored optimization methods exploit structured matrix factorizations, but mainly for curvature approximation and non-private optimization. BASE connects these directions by constructing data-adaptive bilateral subspaces over target linear modules, combining low-dimensional private optimization with the intrinsic matrix structure of linear transformations.

3 PRELIMINARIES

3.1 DIFFERENTIAL PRIVACY

A randomized mechanism $\mathcal{A}$ with input space $\mathcal{X}$ and output space $\mathcal{R}$ satisfies (ε, δ)-differential privacy if, for any two adjacent datasets $D, D' \in \mathcal{X}$ differing in one record and any measurable output event $S \subseteq \mathcal{R}$,

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

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

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

3.2 GAUSSIAN MECHANISM AND PRIVACY ACCOUNTING

In deep learning, differential privacy is commonly achieved through DP-SGD, which clips per-example gradients and adds calibrated Gaussian noise (Liu, 2019). Given a per-example gradient g_i, clipping with threshold C gives

$$\bar{g}_i = g_i \cdot \min\left(1, \frac{C}{\|g_i\|_2}\right), \tag{2}$$

where C is the clipping threshold. For a minibatch $\mathcal{B}$, DP-SGD releases the noisy averaged gradient

$$\tilde{g} = \frac{1}{|\mathcal{B}|}\left(\sum_{i \in \mathcal{B}} \bar{g}_i + \xi\right), \qquad \xi \sim \mathcal{N}(0, \sigma^2 C^2 I_d), \tag{3}$$

where σ is the noise multiplier and I_d is the d-dimensional identity matrix. The noise multiplier is calibrated to ensure that the full training procedure satisfies a target privacy budget. Rényi Differential Privacy (RDP) (Mironov, 2017) is commonly used to compose privacy loss across training steps. Given (ε, δ), sampling probability q, and total steps T, an RDP accountant determines σ.

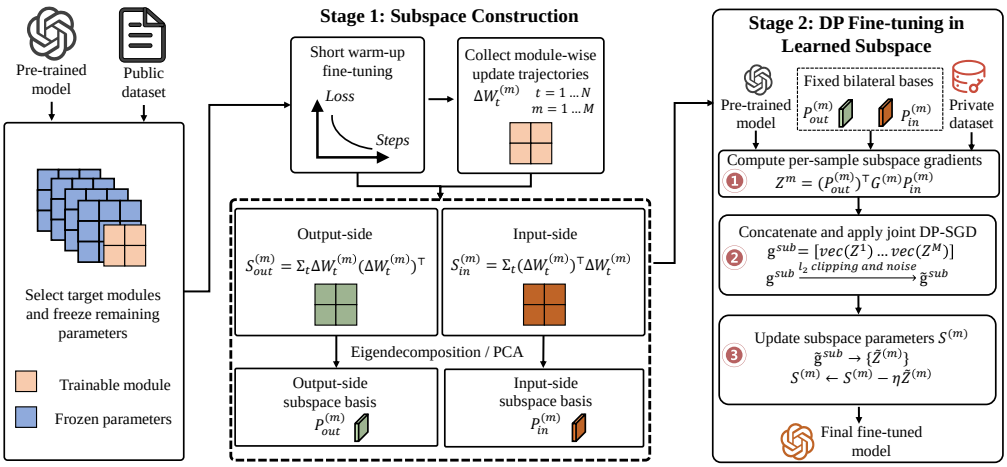


Figure 1: Overview of the proposed BASE framework. Stage 1 uses only public data to construct module-wise subspace bases. Stage 2 reloads the original pre-trained model, keeps the learned bases fixed, and performs differentially private fine-tuning on private data within the learned subspace.

3.3 Kronecker Product and Vectorization

For matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$, the Kronecker product $A \otimes B \in \mathbb{R}^{mp \times nq}$ is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}, \tag{4}$$

where a_{ij} is the (i, j)-th entry of A.

The vectorization operator $\mathrm{vec}(\cdot)$ stacks the columns of a matrix from left to right into a single vector, and we use this convention throughout the paper. Under this column-major convention, for $A \in \mathbb{R}^{m \times n}$, $X \in \mathbb{R}^{n \times p}$, and $B \in \mathbb{R}^{p \times q}$, the standard vectorization identity is given by

$$\mathrm{vec}(AXB) = (B^\top \otimes A)\,\mathrm{vec}(X). \tag{5}$$

Since $AXB \in \mathbb{R}^{m \times q}$, both sides of the identity are vectors in $\mathbb{R}^{mq}$.

4 The BASE Framework

This section introduces BASE, a two-stage framework for differentially private bilateral subspace fine-tuning. Section 4.1 presents the algorithmic procedure, while Section 4.2 establishes the theoretical foundations. We first motivate the bilateral subspace construction from a Kronecker-factored energy-maximization objective. We then establish the relationship between core gradients and bilateral projection of ambient-space gradients, and show that core optimization restricts all effective parameter updates to the corresponding bilateral subspaces. Finally, we prove the privacy guarantee when the projection bases are fixed before private optimization and estimated only from public data.

4.1 The BASE Algorithm

Algorithm 1 summarizes the two-stage procedure of BASE. Stage 1 performs a privacy-free warm-up on the public auxiliary dataset, updating only target linear modules such as attention projections and recording their weight-update trajectories as sequences of parameter displacements from initialization across warm-up steps. For each target module, BASE forms input-side and output-side empirical covariance matrices from these trajectories and extracts their leading principal components to construct fixed bilateral bases P_{in} and P_{out}. Because the bases use only non-sensitive

public data, Stage 1 incurs no privacy cost; the warm-up model is discarded after subspace construction. Stage 2 reloads the pretrained model, reparameterizes each target module in the learned bilateral subspace, and privately optimizes only its low-dimensional core. At each private step, BASE vectorizes and concatenates the per-example subspace gradients from all target modules into a joint subspace gradient. These joint gradients are globally clipped in ℓ_2 norm, averaged over the minibatch, and perturbed with calibrated isotropic Gaussian noise. The noisy joint gradient is partitioned into module-specific core gradients to update the low-dimensional cores, which parameterize the target-module weights through the fixed bilateral bases. Appendix A provides additional implementation details.

4.2 Kronecker-Factored Subspace Objective

The bilateral subspace construction is motivated by the matrix structure of gradients in linear modules. Consider a target weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. For an example z_i, let $x_i \in \mathbb{R}^{d_{\text{in}}}$ denote the input activation and let $\delta_i \in \mathbb{R}^{d_{\text{out}}}$ denote the backpropagated error signal. The corresponding per-example gradient has the outer-product form

$$G_i = \nabla_W \ell(\theta; z_i) = \delta_i x_i^\top, \qquad \text{vec}(G_i) = x_i \otimes \delta_i. \tag{6}$$

Although $\text{vec}(G_i)$ lies in a $d_{\text{out}} d_{\text{in}}$-dimensional space, its second-order structure is naturally associated with the input and output dimensions of the linear module. Directly constructing the full gradient second-moment matrix is prohibitively expensive. We therefore adopt the Kronecker-factored surrogate

$$M = \mathbb{E}\left[\text{vec}(G_i)\,\text{vec}(G_i)^\top\right] \approx C_{\text{in}} \otimes C_{\text{out}}, \tag{7}$$

where $C_{\text{in}} = \mathbb{E}[x_i x_i^\top]$ and $C_{\text{out}} = \mathbb{E}[\delta_i \delta_i^\top]$.

Given orthonormal bilateral bases $P_{\text{out}} \in \mathbb{R}^{d_{\text{out}} \times k_{\text{out}}}$ and $P_{\text{in}} \in \mathbb{R}^{d_{\text{in}} \times k_{\text{in}}}$, we measure the gradient information retained by the induced bilateral subspace through $\|P_{\text{out}}^\top G_i P_{\text{in}}\|_F^2$. This leads to the following expected energy-preservation objective:

$$\max_{P_{\text{out}}, P_{\text{in}}} \quad \mathbb{E}\left[\left\|P_{\text{out}}^\top G_i P_{\text{in}}\right\|_F^2\right] \quad \text{s.t.} \quad P_{\text{out}}^\top P_{\text{out}} = I, \qquad P_{\text{in}}^\top P_{\text{in}} = I. \tag{8}$$

Under the Kronecker-factored approximation in Eq. (7), this objective decomposes as

$$\mathbb{E}\left[\left\|P_{\text{out}}^\top G_i P_{\text{in}}\right\|_F^2\right] \approx \text{tr}\left(P_{\text{in}}^\top C_{\text{in}} P_{\text{in}}\right) \text{tr}\left(P_{\text{out}}^\top C_{\text{out}} P_{\text{out}}\right). \tag{9}$$

For fixed bilateral ranks, the two factors are independently maximized by the leading eigenspaces of C_{in} and C_{out}, respectively.

Rather than explicitly estimating these Kronecker factors, BASE uses public warm-up weight displacements as trajectory-based surrogates for the dominant bilateral update directions. For each target module, let $\{\Delta W_t^{(m)}\}_{t=1}^N$ denote the displacements from the original pretrained weight across the collected warm-up snapshots. We construct the empirical output-side and input-side statistics

$$S_{\text{out}}^{(m)} = \sum_{t=1}^{N} \Delta W_t^{(m)} \left(\Delta W_t^{(m)}\right)^\top, \qquad S_{\text{in}}^{(m)} = \sum_{t=1}^{N} \left(\Delta W_t^{(m)}\right)^\top \Delta W_t^{(m)}. \tag{10}$$

Their leading eigenvectors define the fixed bilateral bases $P_{\text{out}}^{(m)}$ and $P_{\text{in}}^{(m)}$, maximizing the corresponding side-wise trajectory energies. The derivation is provided in Appendix B.1.

4.3 Equivalence to Bilateral Gradient Projection

For fixed orthonormal bases P_{out} and P_{in}, BASE parameterizes the effective weight of a target linear module as

$$W(S) = W_0 + P_{\text{out}} S P_{\text{in}}^\top, \tag{11}$$

where S is the trainable core matrix. Let $G = \nabla_W \ell(W(S))$ denote the gradient with respect to the effective weight matrix. By the chain rule, the gradient with respect to S is

$$\nabla_S \ell = P_{\text{out}}^\top G P_{\text{in}}. \tag{12}$$

Algorithm 1 BASE Training Procedure

Require: Public auxiliary dataset $\mathcal{D}_{\text{aux}}$, private dataset $\mathcal{D}_{\text{priv}}$, pretrained model θ_0, learning rate η, clipping threshold C, noise multiplier σ, bilateral ranks $\{k_{\text{out}}^{(m)}, k_{\text{in}}^{(m)}\}_{m=1}^{M}$.
Ensure: Privately fine-tuned model θ_{final}.

1: **Stage 1: Public bilateral subspace construction.** Starting from θ_0, freeze all non-target parameters and train only the target modules on $\mathcal{D}_{\text{aux}}$. Construct bilateral projection bases $\{P_{\text{out}}^{(m)}, P_{\text{in}}^{(m)}\}_{m=1}^{M}$ and projected warm-up core initializations $\{S_0^{(m)}\}_{m=1}^{M}$.
2: Discard the public warm-up model.
3: **Stage 2: Private subspace optimization.** Reload the original pretrained model θ_0 and initialize $S^{(m)} \leftarrow S_0^{(m)}$ for each target module. For each target module m, parameterize the private update as $\Delta W^{(m)} = P_{\text{out}}^{(m)} S^{(m)} (P_{\text{in}}^{(m)})^\top$. Freeze pretrained parameters and optimize only the core matrices $\{S^{(m)}\}_{m=1}^{M}$ on $\mathcal{D}_{\text{priv}}$ with DP-SGD.
4: **return** θ_{final}.

Mapping this gradient back to the original parameter space gives

$$P_{\text{out}} \nabla_S \ell P_{\text{in}}^\top = P_{\text{out}} P_{\text{out}}^\top G P_{\text{in}} P_{\text{in}}^\top. \tag{13}$$

The right-hand side is the orthogonal projection of G onto the bilateral subspace $\mathcal{S} = \{P_{\text{out}} A P_{\text{in}}^\top : A \in \mathbb{R}^{k_{\text{out}} \times k_{\text{in}}}\}$. Therefore, mapping the gradient with respect to the core matrix back to the original parameter space yields exactly the bilaterally projected gradient. Moreover, any update ΔS to the core matrix induces a parameter update of the form $P_{\text{out}} \Delta S P_{\text{in}}^\top$, which remains within the learned bilateral subspace. The complete derivation is provided in Appendix B.2.

Privacy Guarantee. BASE satisfies (ε, δ)-differential privacy with respect to $\mathcal{D}_{\text{priv}}$. Stage 1 accesses only the public auxiliary dataset $\mathcal{D}_{\text{aux}}$ and incurs no privacy cost, whereas Stage 2 applies standard DP-SGD with global ℓ_2 clipping and Gaussian noise in the fixed bilateral subspaces; mapping the privatized updates back through the public bases is post-processing. Given δ, the resulting ε is computed using a standard privacy accountant from the sampling rate q, noise multiplier σ, and number of steps T, with the formal proof deferred to Appendix B.3.

5 Experiments

We conduct experiments on 5 NVIDIA RTX 5090 GPUs. BASE is evaluated across natural language understanding, dialogue summarization, and instruction-following tasks using three backbone models: **RoBERTa-base**, **RoBERTa-large** (Liu et al., 2019), and **Qwen2.5-1.5B** (Yang et al., 2025). The evaluation covers seven datasets and benchmarks, including four GLUE tasks (Wang et al., 2019): **SST-2**, **QNLI**, **MNLI**, and **QQP**; the dialogue summarization dataset **SAMSum** (Gliwa et al., 2019); the instruction-tuning dataset **Dolly-15k** (Conover et al., 2023); and the instruction-following benchmark **IFEval** (Zhou et al., 2023). We compare BASE with the non-private empirical upper bound **Full FT** and five private fine-tuning baselines: **DP-Full**, **DP-LoRA**, and **DP-Adapter**, following Yu et al. (2022); **DP-BiTFiT**, following Bu et al. (2024); and **DP-SFT**, following Zheng et al. (2026).

Unless otherwise specified, all results are averaged over three random seeds. For all DP methods, we adopt a public-auxiliary-data protocol consistent with prior public-data-assisted private optimization (Yu et al., 2021a; Sha et al., 2025). To simulate the availability of non-sensitive in-distribution auxiliary data on standard benchmarks, we randomly shuffle the original training split and partition it into two disjoint subsets: a public auxiliary subset $\mathcal{D}_{\text{aux}}$ and a private subset $\mathcal{D}_{\text{priv}}$. The former is treated as public and accessed only during Stage 1, whereas all differentially private optimization is performed exclusively on $\mathcal{D}_{\text{priv}}$ in Stage 2. For BASE, $\mathcal{D}_{\text{aux}}$ is used to learn the bilateral subspaces from public warm-up trajectories. Accordingly, all reported privacy guarantees are with respect to examples in $\mathcal{D}_{\text{priv}}$. Detailed optimization settings, hyperparameters, and experimental configurations are provided in Appendix D.

Main results. Table 1 reports the GLUE results with RoBERTa-base and RoBERTa-large. BASE consistently achieves the best DP performance across all reported metrics under both privacy bud-

Table 1: Main GLUE results on RoBERTa-base and RoBERTa-large under $\varepsilon \in \{0.5, 4.0\}$. Results are reported as mean $\pm$ standard deviation over three random seeds. Bold and underline indicate the best and second-best DP results under the same model and privacy budget, respectively.

Method	ε	SST-2	QNLI	MNLI		QQP	
				Matched	Mismatched	Acc.	F1
RoBERTa-base							
Full FT	–	94.30 ± 0.29	92.89 ± 0.09	87.76 ± 0.20	87.51 ± 0.15	91.45 ± 0.05	88.66 ± 0.09
DP-Full	0.5	90.13 ± 0.46	85.18 ± 0.59	81.87 ± 0.18	80.02 ± 0.21	82.65 ± 0.14	78.22 ± 0.15
DP-LoRA	0.5	90.44 ± 0.25	83.51 ± 0.40	80.47 ± 0.16	81.21 ± 0.12	83.60 ± 0.25	79.09 ± 0.23
DP-Adapter	0.5	89.53 ± 0.35	81.39 ± 0.26	77.98 ± 0.13	78.71 ± 0.22	82.44 ± 0.04	77.70 ± 0.08
DP-BiTFiT	0.5	88.76 ± 0.74	78.25 ± 1.13	78.05 ± 0.24	79.24 ± 0.14	81.42 ± 0.05	76.94 ± 0.16
DP-SFT	0.5	<u>91.49 ± 0.65</u>	<u>87.25 ± 0.31</u>	<u>82.33 ± 0.11</u>	<u>82.46 ± 0.25</u>	<u>85.11 ± 0.09</u>	<u>79.94 ± 0.26</u>
BASE	0.5	**92.62 ± 0.43**	**88.57 ± 0.16**	**83.80 ± 0.09**	**83.49 ± 0.19**	**85.99 ± 0.11**	**81.71 ± 0.16**
DP-Full	4.0	91.40 ± 0.49	86.27 ± 0.66	82.52 ± 0.38	82.72 ± 0.13	84.95 ± 0.17	80.79 ± 0.22
DP-LoRA	4.0	91.02 ± 0.19	84.31 ± 0.49	80.96 ± 0.15	81.70 ± 0.26	84.07 ± 0.14	79.58 ± 0.15
DP-Adapter	4.0	90.29 ± 0.35	83.13 ± 0.34	78.89 ± 0.15	79.57 ± 0.27	83.11 ± 0.06	78.39 ± 0.08
DP-BiTFiT	4.0	90.14 ± 0.25	81.15 ± 0.47	78.81 ± 0.24	79.64 ± 0.05	82.66 ± 0.11	77.40 ± 0.26
DP-SFT	4.0	<u>91.97 ± 0.31</u>	<u>87.42 ± 0.16</u>	<u>82.57 ± 0.20</u>	<u>83.16 ± 0.06</u>	<u>85.83 ± 0.07</u>	<u>81.92 ± 0.14</u>
BASE	4.0	**92.78 ± 0.44**	**88.80 ± 0.08**	**83.92 ± 0.17**	**83.99 ± 0.12**	**86.97 ± 0.04**	**82.80 ± 0.17**
RoBERTa-large							
Full FT	–	96.14 ± 0.14	94.62 ± 0.14	90.26 ± 0.04	90.27 ± 0.02	92.07 ± 0.09	89.52 ± 0.15
DP-Full	0.5	91.81 ± 0.19	86.41 ± 0.76	83.19 ± 0.20	83.85 ± 0.34	84.29 ± 0.27	82.08 ± 0.19
DP-LoRA	0.5	92.12 ± 0.28	86.15 ± 1.43	84.64 ± 0.08	84.80 ± 0.31	84.22 ± 0.25	80.85 ± 0.26
DP-Adapter	0.5	91.93 ± 0.60	84.08 ± 0.81	83.64 ± 0.32	83.43 ± 0.20	83.18 ± 0.16	79.49 ± 0.15
DP-BiTFiT	0.5	91.21 ± 0.53	85.24 ± 0.93	82.39 ± 0.58	81.17 ± 0.41	82.28 ± 0.57	79.22 ± 1.16
DP-SFT	0.5	<u>92.89 ± 0.67</u>	<u>89.33 ± 0.62</u>	<u>86.96 ± 0.31</u>	<u>86.74 ± 0.19</u>	<u>85.21 ± 0.24</u>	<u>82.25 ± 0.29</u>
BASE	0.5	**93.69 ± 0.67**	**91.68 ± 0.58**	**88.50 ± 0.16**	**88.16 ± 0.10**	**86.96 ± 0.15**	**83.24 ± 0.13**
DP-Full	4.0	92.87 ± 0.11	87.80 ± 0.40	84.81 ± 0.26	84.69 ± 0.15	85.64 ± 0.23	82.48 ± 0.15
DP-LoRA	4.0	93.39 ± 0.52	86.88 ± 1.32	85.02 ± 0.06	85.16 ± 0.20	85.68 ± 0.20	82.34 ± 0.26
DP-Adapter	4.0	93.08 ± 0.44	86.95 ± 0.09	84.77 ± 0.14	84.12 ± 0.08	84.70 ± 0.15	81.17 ± 0.12
DP-BiTFiT	4.0	92.33 ± 0.27	86.03 ± 0.55	83.45 ± 0.49	83.88 ± 0.31	83.87 ± 0.43	81.76 ± 0.66
DP-SFT	4.0	<u>93.58 ± 0.36</u>	<u>90.11 ± 0.47</u>	<u>87.04 ± 0.12</u>	<u>86.93 ± 0.17</u>	<u>86.12 ± 0.37</u>	<u>82.56 ± 0.19</u>
BASE	4.0	**93.73 ± 0.44**	**91.73 ± 0.52**	**88.55 ± 0.16**	**88.20 ± 0.09**	**87.12 ± 0.15**	**83.38 ± 0.14**

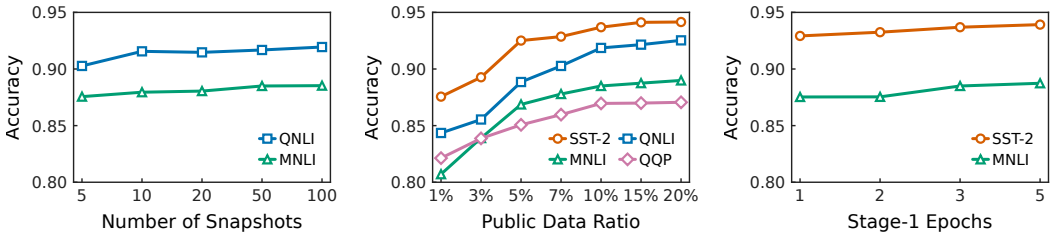
gets and model scales. Averaging over the six reported GLUE metrics, BASE outperforms the strong subspace-based baseline DP-SFT by 1.27/1.48 percentage points (pp) under $\varepsilon = 0.5$ and by 1.07/1.06 pp under $\varepsilon = 4.0$ on RoBERTa-base/RoBERTa-large, respectively. Compared with the strongest remaining DP baseline by the six-metric average, the corresponding improvements are 2.98/3.24 pp under $\varepsilon = 0.5$ and 1.77/2.37 pp under $\varepsilon = 4.0$. Notably, the gains over DP-SFT become particularly pronounced on QNLI and MNLI with RoBERTa-large, indicating that the structured bilateral subspace remains effective as the model scale increases. These utility gains are achieved with a substantially more scalable module-wise parameterization, whose computational and memory efficiency is further analyzed in Table 4.

Table 2 further evaluates BASE on Qwen2.5-1.5B for SAMSum summarization and Dolly-15k instruction tuning evaluated with IFEval. BASE achieves the best performance among all evaluated DP methods on all three ROUGE metrics and both IFEval accuracy metrics under $\varepsilon \in \{0.5, 4.0\}$. DP-SFT is not evaluated at this scale because its global projection matrix incurs prohibitive memory costs, as detailed in Appendix D.1. The consistent improvements over DP-Full, DP-LoRA, DP-Adapter, and DP-BiTFiT demonstrate that the benefit of BASE extends beyond classification to generative and instruction-following tasks at the billion-parameter scale. Together, these results support the effectiveness of learning structured, task-relevant bilateral subspaces for reducing the dimensionality of private optimization while preserving downstream utility.

Table 2: Main results with Qwen2.5-1.5B on SAMSum and Dolly-15k instruction tuning under $\varepsilon \in \{0.5, 4.0\}$. Results are averaged over three random seeds and reported as mean $\pm$ standard deviation. Bold and underline indicate the best and second-best results among the evaluated DP methods under the same privacy budget, respectively.

Method	ε	SAMSum			Dolly-15k + IFEval	
		ROUGE-1	ROUGE-2	ROUGE-L	Prompt Acc.	Inst Acc.
Full FT	–	0.5078 ± 0.0012	0.2692 ± 0.0008	0.4239 ± 0.0015	31.69 ± 0.21%	43.36 ± 0.24%
DP-Full	0.5	0.4402 ± 0.0034	0.2208 ± 0.0016	0.3787 ± 0.0018	26.58 ± 0.32%	36.21 ± 0.27%
DP-LoRA	0.5	0.4574 ± 0.0026	0.2220 ± 0.0015	0.3818 ± 0.0024	26.03 ± 0.18%	36.69 ± 0.16%
DP-Adapter	0.5	0.4528 ± 0.0039	0.2158 ± 0.0028	0.3805 ± 0.0022	25.32 ± 0.11%	35.12 ± 0.14%
DP-BiTFiT	0.5	0.4457 ± 0.0025	0.2105 ± 0.0021	0.3799 ± 0.0023	26.92 ± 0.24%	37.04 ± 0.22%
DP-SFT	0.5			–†		
BASE	0.5	**0.4793 ± 0.0019**	**0.2438 ± 0.0022**	**0.3968 ± 0.0020**	**28.87 ± 0.17%**	**38.62 ± 0.13%**
DP-Full	4.0	0.4547 ± 0.0025	0.2292 ± 0.0022	0.3828 ± 0.0026	27.12 ± 0.21%	37.53 ± 0.23%
DP-LoRA	4.0	0.4611 ± 0.0031	0.2330 ± 0.0026	0.3896 ± 0.0028	26.77 ± 0.25%	36.93 ± 0.28%
DP-Adapter	4.0	0.4649 ± 0.0028	0.2347 ± 0.0024	0.3904 ± 0.0029	26.65 ± 0.22%	36.85 ± 0.24%
DP-BiTFiT	4.0	0.4554 ± 0.0032	0.2249 ± 0.0019	0.3827 ± 0.0035	27.67 ± 0.19%	37.89 ± 0.26%
DP-SFT	4.0			–†		
BASE	4.0	**0.4821 ± 0.0023**	**0.2529 ± 0.0012**	**0.4133 ± 0.0015**	**29.65 ± 0.14%**	**39.44 ± 0.18%**

†DP-SFT is not evaluated at the 1.5B scale due to the prohibitive memory cost of its global projection matrix; see Appendix D.1 for details.



(a) Snapshots (b) Public data ratio (c) Warm-up epochs

Figure 2: Stage-1 subspace construction ablations on RoBERTa-large under $\varepsilon = 0.5$. We vary the number of collected snapshots, the ratio of public data used for warm-up, and the number of Stage-1 warm-up epochs, while keeping the remaining training configuration fixed.

Sensitivity of Stage-1 construction. Figure 2 examines the sensitivity of BASE to Stage-1 subspace construction. BASE remains stable as the number of collected snapshots increases, once sufficient update directions have been observed, indicating that it does not require a long public trajectory. The public data ratio has a more pronounced effect when the ratio is very small, but the gains saturate after a moderate amount of public data is used. Similarly, increasing the number of warm-up epochs provides limited additional improvement after the early epochs. These results suggest that Stage-1 construction is lightweight and can produce effective projection bases without extensive public computation.

Mechanism-level ablations. Table 3 examines Stage-2 core initialization and bilateral subspace construction. With the bilateral bases fixed, core initialization has only a modest effect on performance. Warm-up initialization performs best across four tasks, while Gaussian and Kaiming initialization remain competitive. Importantly, warm-up initialization carries over only the low-dimensional core parameters; the backbone is reloaded from the pretrained checkpoint before Stage 2. Thus, the gains of BASE cannot be primarily attributed to inheriting the Stage-1 model state.

In contrast, the source of the bilateral bases has a substantially larger effect. Here, R and L denote random orthonormal bases and bases learned from public warm-up trajectories, respectively, with the first and second letters referring to P_{out} and P_{in}. Replacing either learned basis with a random one leads to a pronounced performance degradation across all tasks, while using learned bases on both sides consistently performs best. Since all variants retain the same low-dimensional

Table 3: Design ablations on RoBERTa-large under $\varepsilon = 0.5$: (a) Stage-2 core initialization and (b) bilateral basis source. MNLI reports matched accuracy, while QQP reports accuracy. Bold and underline indicate the best and second-best results in each column, respectively.

(a) Core initialization

Initialization	SST-2	QNLI	MNLI	QQP	Avg.
Zero	93.11	90.73	<u>88.26</u>	86.21	89.58
Kaiming	93.35	90.88	87.64	86.06	89.48
Gaussian	<u>93.46</u>	<u>91.27</u>	88.12	<u>86.59</u>	<u>89.86</u>
Warm-up	**93.69**	**91.68**	**88.50**	**86.96**	**90.21**

(b) Basis source

Basis	SST-2	QNLI	MNLI	QQP	Avg.
R–R	50.92	49.46	35.19	63.18	49.69
R–L	51.50	<u>53.89</u>	35.83	<u>70.11</u>	52.83
L–R	<u>55.71</u>	52.00	<u>44.85</u>	64.32	<u>54.22</u>
L–L	**93.69**	**91.68**	**88.50**	**86.96**	**90.21**

Table 4: Efficiency comparison of different DP fine-tuning methods on RoBERTa-large with QNLI under $\varepsilon = 4.0$. All DP methods use the same deterministic 10%/90% public/private data split, with 3 public warm-up epochs followed by 3 private optimization epochs. The logical/physical batch sizes are 32/8 for all methods. Training time and peak memory are averaged over three runs.

Method	Stage-1 / Stage-2 Params	DP Dim.	Peak Mem. (GiB)	Total Time (min)
DP-Full	355.362M / 355.362M	355.362M	18.82	126.19
DP-LoRA	8.130M / 8.130M	8.130M	3.66	106.41
DP-Adapter	3.166M / 3.166M	3.166M	2.67	76.20
DP-BiTFiT	1.323M / 1.323M	1.323M	3.03	74.54
DP-SFT	355.362M / 355.362M	32	91.68	488.13
BASE	100.761M / 24576	24576	5.72	65.17

bilateral parameterization, this ablation isolates the quality of the learned subspace from the effect of dimensionality reduction itself. Together, these results support that the effectiveness of BASE primarily stems from identifying task-relevant update directions through the public warm-up trajectories, rather than from low-dimensional parameterization or Stage-2 initialization alone.

Efficiency analysis. Table 4 compares the computational efficiency of different DP fine-tuning methods on RoBERTa-large with QNLI. Under the same experimental settings, BASE achieves the lowest end-to-end training time among all compared methods, requiring only 65.17 minutes, respectively. Its efficiency mainly comes from the substantial reduction in the private optimization space: while 100.761M parameters are trainable during Stage 1, only 24,576 parameters remain trainable in Stage 2. This reduction is particularly pronounced compared with DP-SFT, for which BASE decreases peak memory from 91.68 to 5.72 GiB and training time from 488.13 to 65.17 minutes, corresponding to a $16.0\times$ reduction in memory usage and a $7.5\times$ speedup. Overall, these efficiency gains stem from performing private optimization within compact module-wise subspaces, thereby avoiding both high-dimensional DP updates and costly full-space subspace operations.

6 Conclusion

In this work, we present BASE, a differentially private fine-tuning framework that constrains private fine-tuning to a low-dimensional bilateral subspace derived from public auxiliary data. By decomposing updates into input-side and output-side components, BASE reduces the effective optimization dimension and performs subspace-restricted updates efficiently, without materializing dense projection matrices. We theoretically motivate the bilateral construction through a Kronecker-factored gradient-energy objective and show that core optimization corresponds to gradient projection onto the learned bilateral subspace. Extensive experiments demonstrate that BASE consistently outperforms strong DP baselines in both privacy–utility trade-offs and computational efficiency. These results demonstrate that bilateral subspaces provide an effective and efficient way to improve differentially private fine-tuning, highlighting their potential for future research in privacy-preserving adaptation of large language models.

AI USE STATEMENT

Large language models (LLMs) were used to support manuscript preparation and language refinement, including grammar correction, sentence rewriting, improving clarity and readability, and verifying the consistency of mathematical notation and theoretical arguments. The research ideas, methodological design, experiments, and analyses were developed and conducted by the authors without relying on LLMs for their origination. All AI-assisted material was carefully examined and revised by the authors to ensure its correctness and consistency with the underlying research. The authors take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

We provide comprehensive experimental details to support reproducibility. The experimental setup, including datasets, backbone models, baselines, data-partition protocol, privacy setting, hardware, and random-seed configuration, is described in Section 5. Detailed task-specific hyperparameters and additional experimental configurations are provided in Appendix D. The implementation code for reproducing the experiments have been released in an anonymous repository, with the link provided in the abstract.

ETHICS STATEMENT

This work follows the ICLR Code of Ethics. Our study focuses on differentially private fine-tuning of large language models and does not involve new data collection, human-subject experiments, or animal studies. All experiments are conducted on established publicly available datasets and benchmarks, including GLUE, SAMSum, Dolly-15k, and IFEval, following their intended research usage. The proposed method is designed to improve the utility and efficiency of privacy-preserving fine-tuning while providing formal differential privacy guarantees for the private training data. We do not attempt to identify individuals or recover sensitive information from the datasets.

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.

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/.

Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical Gauss-Newton optimisation for deep learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 557–565. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/botev17a.html.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc.,

2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

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.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. URL http://jmlr.org/papers/v25/23-0870.html.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Third Theory of Cryptography Conference (TCC 2006)*, volume 3876 of *Lecture Notes in Computer Science*, pp. 265–284. Springer, March 2006. ISBN 3-540-32731-2. URL https://www.microsoft.com/en-us/research/publication/calibrating-noise-to-sensitivity-in-private-data-analysis/.

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/.

Roger Grosse and James Martens. A kronecker-factored approximate fisher matrix for convolution layers. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 573–582, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/grosse16.html.

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.

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 Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

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 *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

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.

Gavin Kerrigan, Dylan Slack, and Jens Tuyls. Differentially private language models benefit from public pre-training. In Oluwaseyi Feyisetan, Sepideh Ghanavati, Shervin Malmasi, and Patricia Thaine (eds.), *Proceedings of the Second Workshop on Privacy in NLP*, pp. 39–45, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.privatenlp-1.5. URL https://aclanthology.org/2020.privatenlp-1.5/.

Xianzhi Li, Ran Zmigrod, Zhiqiang Ma, Xiaomo Liu, and Xiaodan Zhu. Fine-tuning language models with differential privacy through adaptive noise allocation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8368–8375, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.491. URL https://aclanthology.org/2024.findings-emnlp.491/.

Fang Liu. Generalized gaussian mechanism for differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 31(4):747–756, 2019. doi: 10.1109/TKDE.2018.2845388.

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.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2408–2417, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/martens15.html.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Haichao Sha, Yuncheng Wu, Ruixuan Liu, Yang Cao, and Hong Chen. Differentially private visual learning with public subspace augmented by synthetic data. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 11357–11366, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3755037. URL https://doi.org/10.1145/3746027.3755037.

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, 2019. URL https://arxiv.org/abs/1804.07461.

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. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12208–12218. PMLR, 18–24 Jul 2021b. URL https://proceedings.mlr.press/v139/yu21f.html.

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.

Lele Zheng, Xiang Wang, Tao Zhang, Yang Cao, Ke Cheng, and Yulong Shen. Differentially private subspace fine-tuning for large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(34):28830–28838, Mar. 2026. doi: 10.1609/aaai.v40i34.40117. URL https://ojs.aaai.org/index.php/AAAI/article/view/40117.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

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 Detailed Description of the BASE Training Procedure

This section provides a detailed description of the two training stages of BASE. Algorithm 2 describes the public bilateral subspace construction stage, and Algorithm 3 describes the private subspace optimization stage. The key design is that the public warm-up model in Stage 1 is used to construct the bilateral projection bases and the low-dimensional core initialization, and is discarded afterward. Stage 2 always reloads the original pretrained model θ_0 and performs differentially private optimization only within the fixed low-dimensional subspaces.

Stage 1: Public bilateral subspace construction. Stage 1 constructs module-wise bilateral subspaces from the public auxiliary dataset $\mathcal{D}_{\text{aux}}$. The target modules are selected linear layers in the pretrained model, such as the query, key, value, and output projections in the attention blocks or the linear layers in the feed-forward networks. Starting from the pretrained model θ_0, BASE freezes all non-target parameters and performs a short warm-up fine-tuning process in which only the selected target modules are updated. During this process, a sequence of weight snapshots is collected for each target module. For module m, the displacement at snapshot t is defined as $\Delta W_t^{(m)} = W_t^{(m)} - W_0^{(m)}$, where $W_0^{(m)}$ is the corresponding pretrained weight. Measuring all snapshots relative to the same pretrained weight allows the collected trajectory to reflect the cumulative task-induced variation observed during public warm-up.

Rather than vectorizing and jointly decomposing the full update trajectory, BASE summarizes its matrix structure through two module-wise trajectory statistics. The output-side statistic $S_{\text{out}}^{(m)}$ captures variation along the output dimension, while the input-side statistic $S_{\text{in}}^{(m)}$ captures variation

Algorithm 2 Public Bilateral Subspace Construction

Require: Public auxiliary dataset $\mathcal{D}_{\text{aux}}$, pretrained model θ_0, target weights $\{W_0^{(m)}\}_{m=1}^M$, bilateral ranks $\{k_{\text{out}}^{(m)}, k_{\text{in}}^{(m)}\}_{m=1}^M$.

Ensure: Projection bases $\{P_{\text{out}}^{(m)}, P_{\text{in}}^{(m)}\}_{m=1}^M$ and initial cores $\{S_0^{(m)}\}_{m=1}^M$.

1: Initialize a warm-up model $\theta_{\text{aux}} \leftarrow \theta_0$.
2: Freeze all non-target parameters and train only the target modules on $\mathcal{D}_{\text{aux}}$.
3: During warm-up, collect N target-module snapshots and form $\{\Delta W_t^{(m)} = W_t^{(m)} - W_0^{(m)}\}_{t=1}^N$.
4: After warm-up, compute the final displacement $\Delta W_{\text{aux}}^{(m)} = W_{\text{aux}}^{(m)} - W_0^{(m)}$ for each target module.
5: **for** each target module $m = 1$ **to** M **do**
6: Compute $S_{\text{out}}^{(m)} = \sum_{t=1}^N \Delta W_t^{(m)} \left(\Delta W_t^{(m)}\right)^\top$.
7: Compute $S_{\text{in}}^{(m)} = \sum_{t=1}^N \left(\Delta W_t^{(m)}\right)^\top \Delta W_t^{(m)}$.
8: Let $P_{\text{out}}^{(m)}$ and $P_{\text{in}}^{(m)}$ be the top-$k_{\text{out}}^{(m)}$ and top-$k_{\text{in}}^{(m)}$ eigenvectors of $S_{\text{out}}^{(m)}$ and $S_{\text{in}}^{(m)}$, respectively.
9: Compute the warm-up core initialization

$$S_0^{(m)} = \left(P_{\text{out}}^{(m)}\right)^\top \Delta W_{\text{aux}}^{(m)} P_{\text{in}}^{(m)}.$$

10: **end for**
11: Discard the warm-up model θ_{aux}.
12: **return** $\{P_{\text{out}}^{(m)}, P_{\text{in}}^{(m)}, S_0^{(m)}\}_{m=1}^M$.

along the input dimension. Their leading eigenvectors identify the dominant output-side and input-side directions, respectively. BASE retains the top $k_{\text{out}}^{(m)}$ eigenvectors of $S_{\text{out}}^{(m)}$ as $P_{\text{out}}^{(m)}$ and the top $k_{\text{in}}^{(m)}$ eigenvectors of $S_{\text{in}}^{(m)}$ as $P_{\text{in}}^{(m)}$. The resulting bases are fixed after Stage 1 and are subsequently used to restrict private optimization to the learned bilateral subspaces.

Because all warm-up updates and trajectory statistics are computed solely from $\mathcal{D}_{\text{aux}}$, Stage 1 incurs no privacy cost with respect to $\mathcal{D}_{\text{priv}}$. After constructing the projection bases and initial cores, BASE discards the warm-up model θ_{aux} and retains only $\{P_{\text{out}}^{(m)}, P_{\text{in}}^{(m)}, S_0^{(m)}\}_{m=1}^M$. Stage 2 then reloads the original pretrained model θ_0, ensuring that private fine-tuning does not continue from the public warm-up model. Algorithm 2 summarizes the complete construction procedure.

Stage 2: Private subspace optimization. Stage 2 performs differentially private fine-tuning over the low-dimensional core matrices. This stage does not continue training from the public warm-up model θ_{aux}. Instead, BASE reloads the original pretrained model θ_0, freezes all pretrained parameters, and keeps the bilateral bases $\{P_{\text{out}}^{(m)}, P_{\text{in}}^{(m)}\}_{m=1}^M$ fixed throughout private training.

For each target module m, let $W_{\text{aux}}^{(m)}$ denote its final weight after public warm-up and define the corresponding displacement from the pretrained weight as

$$\Delta W_{\text{aux}}^{(m)} = W_{\text{aux}}^{(m)} - W_0^{(m)}. \tag{14}$$

The Stage 2 core is initialized by projecting this public warm-up displacement onto the learned bilateral subspace:

$$S_0^{(m)} = \left(P_{\text{out}}^{(m)}\right)^\top \Delta W_{\text{aux}}^{(m)} P_{\text{in}}^{(m)}. \tag{15}$$

Thus, warm-up initialization transfers only the low-dimensional component of the public update captured by the learned bases, rather than continuing from the full warm-up model. Unless otherwise specified, this initialization is used in the main experiments.

For each target module m, the trainable core matrix $S^{(m)} \in \mathbb{R}^{k_{\text{out}}^{(m)} \times k_{\text{in}}^{(m)}}$ is then optimized during Stage 2, and the effective weight at private step t is

$$W_t^{(m)} = W_0^{(m)} + P_{\text{out}}^{(m)} S_t^{(m)} \left(P_{\text{in}}^{(m)}\right)^\top. \tag{16}$$

Thus, only the core matrices are optimized, while every resulting update to a target weight remains within the bilateral subspace determined by the public bases.

Algorithm 3 Private Subspace Optimization

Require: Private dataset $\mathcal{D}_{\text{priv}}$, pretrained model θ_0, target weights $\{W_0^{(m)}\}_{m=1}^M$, bases $\{P_{\text{out}}^{(m)}, P_{\text{in}}^{(m)}\}_{m=1}^M$, initial cores $\{S_0^{(m)}\}_{m=1}^M$, learning rate η, clipping threshold C, noise multiplier σ, sampling rate q.

Ensure: Privately fine-tuned model θ_{final}.

Reload a fresh model $\theta_{\text{priv}} \leftarrow \theta_0$.

Freeze all pretrained parameters and initialize $S^{(m)} \leftarrow S_0^{(m)}$ for each target module m.

Let

$$R = \sum_{m=1}^{M} k_{\text{out}}^{(m)} k_{\text{in}}^{(m)}.$$

for each private step $t = 0$ **to** $T - 1$ **do**

  Poisson-sample a minibatch $\mathcal{B}_t \subseteq \mathcal{D}_{\text{priv}}$ with sampling rate q.

  for each sample $z_i \in \mathcal{B}_t$ **do**

    for each target module $m = 1$ **to** M **do**

      Use effective weight

$$W_t^{(m)} = W_0^{(m)} + P_{\text{out}}^{(m)} S_t^{(m)} (P_{\text{in}}^{(m)})^\top.$$

      Compute the module-wise subspace gradient

$$Z_{i,t}^{(m)} = \left(P_{\text{out}}^{(m)}\right)^\top \nabla_{W_t^{(m)}} \mathcal{L}(\theta_t; z_i) P_{\text{in}}^{(m)}.$$

    end for

    Concatenate

$$g_{i,t}^{\text{sub}} = \text{Concat}\left(\left\{\text{vec}\left(Z_{i,t}^{(m)}\right)\right\}_{m=1}^{M}\right) \in \mathbb{R}^R.$$

    Clip

$$\bar{g}_{i,t}^{\text{sub}} = g_{i,t}^{\text{sub}} \cdot \min\left(1, \frac{C}{\left\|g_{i,t}^{\text{sub}}\right\|_2}\right).$$

  end for

  Sample $\xi_t \sim \mathcal{N}(0, \sigma^2 C^2 I_R)$ and set

$$\widetilde{g}_t^{\text{sub}} = \frac{1}{|\mathcal{B}_t|}\left(\sum_{z_i \in \mathcal{B}_t} \bar{g}_{i,t}^{\text{sub}} + \xi_t\right).$$

  Unflatten $\widetilde{g}_t^{\text{sub}}$ into $\{\widetilde{Z}_t^{(m)}\}_{m=1}^M$ and update

$$S_{t+1}^{(m)} = \text{Adam}\left(S_t^{(m)}, \widetilde{Z}_t^{(m)}, \eta\right)$$

  for all m.

end for

Compose

$$W_{\text{final}}^{(m)} = W_0^{(m)} + P_{\text{out}}^{(m)} S_T^{(m)} (P_{\text{in}}^{(m)})^\top$$

for all target modules.

return θ_{final}.

B PROOF

B.1 DERIVATION OF THE KRONECKER-FACTORED SUBSPACE OBJECTIVE

This section first derives the Kronecker-factored gradient-energy objective that motivates the bilateral structure, and then formalizes the separable trajectory-energy surrogate optimized by the construction used in BASE. For clarity, we omit the module superscript (m) in the theoretical derivation and restore it for the trajectory-based construction.

Consider a target linear module with weight $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. For an example z_i, let $x_i \in \mathbb{R}^{d_{\text{in}}}$ denote the input activation and $\delta_i \in \mathbb{R}^{d_{\text{out}}}$ the backpropagated error signal. The corresponding per-example gradient and its column-major vectorization are

$$G_i = \nabla_W \ell(\theta; z_i) = \delta_i x_i^\top, \qquad \operatorname{vec}(G_i) = x_i \otimes \delta_i. \tag{17}$$

Let

$$M = \mathbb{E}\left[\operatorname{vec}(G_i)\operatorname{vec}(G_i)^\top\right] \tag{18}$$

denote the full gradient second-moment matrix. Using the mixed-product property of the Kronecker product,

$$\begin{aligned} M &= \mathbb{E}\left[(x_i \otimes \delta_i)(x_i \otimes \delta_i)^\top\right] \\ &= \mathbb{E}\left[(x_i x_i^\top) \otimes (\delta_i \delta_i^\top)\right]. \end{aligned} \tag{19}$$

Constructing M directly requires memory quadratic in the number of entries of W. We therefore adopt the Kronecker-factored approximation

$$M \approx C_{\text{in}} \otimes C_{\text{out}}, \qquad C_{\text{in}} = \mathbb{E}[x_i x_i^\top], \qquad C_{\text{out}} = \mathbb{E}[\delta_i \delta_i^\top]. \tag{20}$$

We use this as a structured approximation of the full gradient second moment; the approximation need not hold exactly and is not invoked as a claim of sample-wise independence between x_i and δ_i.

Let $P_{\text{out}} \in \mathbb{R}^{d_{\text{out}} \times k_{\text{out}}}$ and $P_{\text{in}} \in \mathbb{R}^{d_{\text{in}} \times k_{\text{in}}}$ have orthonormal columns. The gradient information retained by the corresponding bilateral subspace is measured by

$$\mathcal{J}(P_{\text{out}}, P_{\text{in}}) = \mathbb{E}\left[\left\|P_{\text{out}}^\top G_i P_{\text{in}}\right\|_F^2\right]. \tag{21}$$

Using $\operatorname{vec}(AXB) = (B^\top \otimes A)\operatorname{vec}(X)$, we obtain

$$\begin{aligned} \mathcal{J}(P_{\text{out}}, P_{\text{in}}) &= \operatorname{tr}\left[\left(P_{\text{in}} P_{\text{in}}^\top \otimes P_{\text{out}} P_{\text{out}}^\top\right) M\right] \\ &\approx \operatorname{tr}\left[\left(P_{\text{in}} P_{\text{in}}^\top \otimes P_{\text{out}} P_{\text{out}}^\top\right)(C_{\text{in}} \otimes C_{\text{out}})\right] \\ &= \operatorname{tr}\left(P_{\text{in}}^\top C_{\text{in}} P_{\text{in}}\right) \operatorname{tr}\left(P_{\text{out}}^\top C_{\text{out}} P_{\text{out}}\right). \end{aligned} \tag{22}$$

The last equality follows from the mixed-product property of the Kronecker product, the cyclic property of the trace, and $\operatorname{tr}(A \otimes B) = \operatorname{tr}(A)\operatorname{tr}(B)$.

Because C_{in} and C_{out} are positive semidefinite, the two factors in Eq. (22) are nonnegative and depend separately on P_{in} and P_{out}. For fixed bilateral ranks, a maximizer of the factorized objective can therefore be obtained by maximizing the two factors independently. By the Ky Fan maximum principle,

$$\begin{aligned} \max_{P_{\text{in}}^\top P_{\text{in}} = I} \operatorname{tr}\left(P_{\text{in}}^\top C_{\text{in}} P_{\text{in}}\right) &= \sum_{j=1}^{k_{\text{in}}} \lambda_j(C_{\text{in}}), \\ \max_{P_{\text{out}}^\top P_{\text{out}} = I} \operatorname{tr}\left(P_{\text{out}}^\top C_{\text{out}} P_{\text{out}}\right) &= \sum_{j=1}^{k_{\text{out}}} \lambda_j(C_{\text{out}}), \end{aligned} \tag{23}$$

where the eigenvalues are ordered in descending order. Hence, the factorized gradient objective is maximized when P_{in} and P_{out} span the leading eigenspaces of C_{in} and C_{out}, respectively.

Trajectory-based realization. BASE does not explicitly estimate C_{in} and C_{out} from per-example activations and backpropagated error signals. Instead, the Kronecker analysis motivates a separate input- and output-side subspace construction, which BASE realizes empirically using the public warm-up trajectory.

For each target module m, let

$$\Delta W_t^{(m)} = W_t^{(m)} - W_0^{(m)}, \qquad t = 1, \dots, N, \tag{24}$$

denote the weight displacements collected at the N warm-up snapshots. BASE constructs the trajectory side statistics

$$S_{\text{out}}^{(m)} = \sum_{t=1}^{N} \Delta W_t^{(m)} \left(\Delta W_t^{(m)}\right)^\top, \qquad S_{\text{in}}^{(m)} = \sum_{t=1}^{N} \left(\Delta W_t^{(m)}\right)^\top \Delta W_t^{(m)}. \tag{25}$$

A normalization factor $1/N$ is omitted because it does not affect the eigenvectors.

For any orthonormal $P_{\text{out}}^{(m)}$ and $P_{\text{in}}^{(m)}$, define the retained output-side and input-side trajectory energies as

$$\begin{aligned} \mathcal{E}_{\text{out}}^{(m)}\left(P_{\text{out}}^{(m)}\right) &= \sum_{t=1}^{N} \left\| \left(P_{\text{out}}^{(m)}\right)^\top \Delta W_t^{(m)} \right\|_F^2, \\ \mathcal{E}_{\text{in}}^{(m)}\left(P_{\text{in}}^{(m)}\right) &= \sum_{t=1}^{N} \left\| \Delta W_t^{(m)} P_{\text{in}}^{(m)} \right\|_F^2. \end{aligned} \tag{26}$$

Expanding the Frobenius norms gives

$$\begin{aligned} \mathcal{E}_{\text{out}}^{(m)}\left(P_{\text{out}}^{(m)}\right) &= \operatorname{tr}\left[\left(P_{\text{out}}^{(m)}\right)^\top S_{\text{out}}^{(m)} P_{\text{out}}^{(m)} \right], \\ \mathcal{E}_{\text{in}}^{(m)}\left(P_{\text{in}}^{(m)}\right) &= \operatorname{tr}\left[\left(P_{\text{in}}^{(m)}\right)^\top S_{\text{in}}^{(m)} P_{\text{in}}^{(m)} \right]. \end{aligned} \tag{27}$$

Motivated by the factorized gradient objective in Eq. (22), we define the corresponding factorized trajectory surrogate

$$\begin{aligned} \mathcal{J}_{\text{traj}}^{(m)} &= \mathcal{E}_{\text{out}}^{(m)}\left(P_{\text{out}}^{(m)}\right) \mathcal{E}_{\text{in}}^{(m)}\left(P_{\text{in}}^{(m)}\right) \\ &= \operatorname{tr}\left[\left(P_{\text{out}}^{(m)}\right)^\top S_{\text{out}}^{(m)} P_{\text{out}}^{(m)} \right] \operatorname{tr}\left[\left(P_{\text{in}}^{(m)}\right)^\top S_{\text{in}}^{(m)} P_{\text{in}}^{(m)} \right]. \end{aligned} \tag{28}$$

We emphasize that $J_{\text{traj}}^{(m)}$ is a separable surrogate and is generally distinct from the joint bilateral trajectory energy $\sum_{t=1}^{N} \|(P_{\text{out}}^{(m)})^\top \Delta W_t^{(m)} P_{\text{in}}^{(m)}\|_F^2$. The basis construction in BASE maximizes the former rather than the latter.

The two factors in Eq. (28) are nonnegative and depend separately on the two bilateral bases. Applying the Ky Fan maximum principle again, the leading $k_{\text{out}}^{(m)}$ eigenvectors of $S_{\text{out}}^{(m)}$ maximize the retained output-side trajectory energy, while the leading $k_{\text{in}}^{(m)}$ eigenvectors of $S_{\text{in}}^{(m)}$ maximize the retained input-side trajectory energy. Therefore, choosing these leading eigenspaces jointly maximizes the factorized trajectory surrogate and yields the bilateral bases $P_{\text{out}}^{(m)}$ and $P_{\text{in}}^{(m)}$ used by BASE.

Importantly, $S_{\text{out}}^{(m)}$ and $S_{\text{in}}^{(m)}$ are not assumed to estimate C_{out} and C_{in} themselves. The role of the Kronecker analysis is to motivate the factorized output- and input-side subspace structure, whereas the public trajectory provides a task-adaptive empirical surrogate with the same separable optimization structure. Consequently, the trajectory construction selects the bilateral bases that optimally retain the observed side-wise public-update energy, without requiring construction of the full gradient second-moment matrix or explicit estimation of the activation and error second moments.

B.2 Core-Matrix Gradients and Bilateral Gradient Projection

This section establishes the relationship between optimization over the low-dimensional core matrix and bilateral gradient projection in the original parameter space. We show that the gradient with respect to the core matrix is obtained by bilaterally projecting the gradient with respect to the weight matrix. Mapping the resulting subspace gradient back to the original parameter space yields the orthogonal projection onto the induced bilateral subspace. We further show that this mapping preserves the Frobenius norm and that global clipping of the concatenated subspace gradient applies the same shared clipping factor to the corresponding bilaterally projected module-wise gradients.

Consider one target linear module with pretrained weight $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. Let $P_{\text{out}} \in \mathbb{R}^{d_{\text{out}} \times k_{\text{out}}}$ and $P_{\text{in}} \in \mathbb{R}^{d_{\text{in}} \times k_{\text{in}}}$ be fixed bilateral bases with orthonormal columns, satisfying $P_{\text{out}}^\top P_{\text{out}} = I$ and $P_{\text{in}}^\top P_{\text{in}} = I$. BASE introduces a trainable core matrix $S \in \mathbb{R}^{k_{\text{out}} \times k_{\text{in}}}$ and parameterizes the target linear module as

$$W(S) = W_0 + P_{\text{out}} S P_{\text{in}}^\top. \tag{29}$$

The resulting updates are restricted to the bilateral subspace $\mathcal{S} = \{P_{\text{out}} A P_{\text{in}}^\top : A \in \mathbb{R}^{k_{\text{out}} \times k_{\text{in}}}\}$.

Let $\phi(S) = \mathcal{L}(W(S))$ denote the reparameterized objective, and let $G = \nabla_W \mathcal{L}(W(S))$ denote the gradient with respect to the weight matrix. For an infinitesimal perturbation dS, the corresponding perturbation of the weight is $dW = P_{\text{out}} dS P_{\text{in}}^\top$. Using the Frobenius inner product, we obtain

$$\begin{aligned} d\phi &= \langle G, dW \rangle_F \\ &= \operatorname{tr}\left(G^\top P_{\text{out}} dS P_{\text{in}}^\top\right) \\ &= \left\langle P_{\text{out}}^\top G P_{\text{in}}, dS \right\rangle_F. \end{aligned} \tag{30}$$

Therefore, the gradient with respect to the core matrix is

$$\nabla_S \phi(S) = P_{\text{out}}^\top G P_{\text{in}}. \tag{31}$$

Mapping this core gradient back to the ambient parameter space gives

$$\begin{aligned} \mathcal{T}\left(\nabla_S \phi(S)\right) &= P_{\text{out}} \left(P_{\text{out}}^\top G P_{\text{in}}\right) P_{\text{in}}^\top \\ &= P_{\text{out}} P_{\text{out}}^\top G P_{\text{in}} P_{\text{in}}^\top. \end{aligned} \tag{32}$$

Define

$$\Pi_{\mathcal{S}}(G) = P_{\text{out}} P_{\text{out}}^\top G P_{\text{in}} P_{\text{in}}^\top. \tag{33}$$

The matrix $\Pi_{\mathcal{S}}(G)$ belongs to $\mathcal{S}$. To show that it is the orthogonal projection of G onto $\mathcal{S}$, consider an arbitrary matrix

$$H = P_{\text{out}} A P_{\text{in}}^\top \in \mathcal{S}. \tag{34}$$

We have

$$\begin{aligned} \langle G - \Pi_{\mathcal{S}}(G), H \rangle_F &= \operatorname{tr}\left[\left(G - P_{\text{out}} P_{\text{out}}^\top G P_{\text{in}} P_{\text{in}}^\top\right)^\top P_{\text{out}} A P_{\text{in}}^\top\right] \\ &= \operatorname{tr}\left(G^\top P_{\text{out}} A P_{\text{in}}^\top\right) \\ &\quad - \operatorname{tr}\left[P_{\text{in}} P_{\text{in}}^\top G^\top P_{\text{out}} P_{\text{out}}^\top P_{\text{out}} A P_{\text{in}}^\top\right] \\ &= \operatorname{tr}\left(G^\top P_{\text{out}} A P_{\text{in}}^\top\right) - \operatorname{tr}\left(G^\top P_{\text{out}} A P_{\text{in}}^\top\right) \\ &= 0, \end{aligned} \tag{35}$$

where the third equality follows from $P_{\text{out}}^\top P_{\text{out}} = I$, $P_{\text{in}}^\top P_{\text{in}} = I$, and the cyclic property of the trace. Hence, $\Pi_{\mathcal{S}}(G)$ is the orthogonal projection of G onto the bilateral subspace $\mathcal{S}$. Combining Eq. (31) and Eq. (33) yields

$$\mathcal{T}\left(\nabla_S \phi(S)\right) = \Pi_{\mathcal{S}}(G). \tag{36}$$

Thus, the core gradient and the bilaterally projected ambient-space gradient represent the same gradient information under the fixed bilateral mapping.

The mapping $\mathcal{T}$ also preserves the Frobenius inner product. For any two core matrices A and B,

$$\begin{aligned}\langle \mathcal{T}(A), \mathcal{T}(B) \rangle_F &= \operatorname{tr}\left[\left(P_{\text{out}} A P_{\text{in}}^\top \right)^\top P_{\text{out}} B P_{\text{in}}^\top \right] \\ &= \operatorname{tr}\left[P_{\text{in}} A^\top P_{\text{out}}^\top P_{\text{out}} B P_{\text{in}}^\top \right] \\ &= \operatorname{tr}\left[P_{\text{in}} A^\top B P_{\text{in}}^\top \right] \\ &= \operatorname{tr}\left[A^\top B P_{\text{in}}^\top P_{\text{in}} \right] \\ &= \operatorname{tr}\left(A^\top B \right) \\ &= \langle A, B \rangle_F . \end{aligned} \tag{37}$$

Setting $A = B$ gives

$$\left\| P_{\text{out}} A P_{\text{in}}^\top \right\|_F = \|A\|_F . \tag{38}$$

For a per-example ambient-space gradient G_i, define the corresponding subspace gradient as

$$Z_i = P_{\text{out}}^\top G_i P_{\text{in}} . \tag{39}$$

Its mapped ambient-space form is

$$P_{\text{out}} Z_i P_{\text{in}}^\top = \Pi_{\mathcal{S}}(G_i) . \tag{40}$$

By Eq. (38),

$$\|Z_i\|_F = \left\| \Pi_{\mathcal{S}}(G_i) \right\|_F . \tag{41}$$

Thus, the norm of a module-wise subspace gradient is exactly equal to the norm of its bilaterally projected counterpart in the ambient parameter space.

For multiple target modules, let $\{Z_i^{(m)}\}_{m=1}^M$ denote the module-wise subspace gradients. BASE vectorizes and concatenates these blocks to form the joint per-example subspace gradient

$$g_i^{\text{sub}} = \operatorname{Concat}\left(\left\{ \operatorname{vec}\left(Z_i^{(m)} \right) \right\}_{m=1}^M \right) \in \mathbb{R}^R, \qquad R = \sum_{m=1}^M k_{\text{out}}^{(m)} k_{\text{in}}^{(m)} . \tag{42}$$

Its squared Euclidean norm satisfies

$$\begin{aligned}\left\| g_i^{\text{sub}} \right\|_2^2 &= \sum_{m=1}^M \left\| \operatorname{vec}\left(Z_i^{(m)} \right) \right\|_2^2 \\ &= \sum_{m=1}^M \left\| Z_i^{(m)} \right\|_F^2 \\ &= \sum_{m=1}^M \left\| P_{\text{out}}^{(m)} Z_i^{(m)} \left(P_{\text{in}}^{(m)} \right)^\top \right\|_F^2 . \end{aligned} \tag{43}$$

BASE applies global ℓ_2 clipping to this concatenated gradient:

$$\bar{g}_i^{\text{sub}} = g_i^{\text{sub}} \cdot \min\left(1, \frac{C}{\left\| g_i^{\text{sub}} \right\|_2} \right) . \tag{44}$$

Since the same scalar clipping factor is applied to every block of g_i^{sub}, unflattening $\bar{g}_i^{\text{sub}}$ yields

$$\bar{Z}_i^{(m)} = Z_i^{(m)} \cdot \min\left(1, \frac{C}{\left\| g_i^{\text{sub}} \right\|_2} \right) \tag{45}$$

for every target module m. Mapping each clipped block back to the ambient parameter space gives

$$P_{\text{out}}^{(m)} \bar{Z}_i^{(m)} \left(P_{\text{in}}^{(m)} \right)^\top = P_{\text{out}}^{(m)} \left(P_{\text{out}}^{(m)} \right)^\top G_i^{(m)} P_{\text{in}}^{(m)} \left(P_{\text{in}}^{(m)} \right)^\top \cdot \min\left(1, \frac{C}{\|g_i^{\text{sub}}\|_2} \right) . \tag{46}$$

Therefore, global ℓ_2 clipping of the joint subspace gradient g_i^{sub} applies exactly the same shared clipping factor to the collection of bilaterally projected module-wise gradients in the ambient parameter space.

For ordinary gradient descent, the core matrix is updated as

$$S_{t+1} = S_t - \eta \nabla_S \phi(S_t). \tag{47}$$

The resulting change in the effective ambient-space weight is

$$\begin{aligned} W(S_{t+1}) - W(S_t) &= P_{\text{out}} \left(S_{t+1} - S_t\right) P_{\text{in}}^\top \\ &= -\eta P_{\text{out}} \nabla_S \phi(S_t) P_{\text{in}}^\top \\ &= -\eta \Pi_{\mathcal{S}}(G_t). \end{aligned} \tag{48}$$

Therefore, ordinary gradient descent on the core matrix is exactly equivalent to applying a bilaterally projected gradient-descent update in the ambient parameter space.

When a general optimizer is applied to the core matrix, let U_t denote the update produced from the current and previous privatized subspace gradients, such that

$$S_{t+1} = S_t + U_t. \tag{49}$$

The corresponding ambient-space change is

$$W(S_{t+1}) - W(S_t) = P_{\text{out}} U_t P_{\text{in}}^\top \in \mathcal{S}. \tag{50}$$

Hence, optimization of the core matrix always restricts the effective parameter update to the learned bilateral subspace. For adaptive optimizers such as Adam, however, the update U_t includes element-wise scaling based on historical subspace gradients. The resulting update therefore does not generally coincide with applying the same adaptive optimizer directly to the projected ambient-space gradient. The exact update equivalence in Eq. (48) applies to ordinary gradient descent, whereas the gradient relationship, orthogonal projection property, norm preservation, joint clipping equivalence, and subspace restriction remain valid for the general core-matrix optimization used by BASE.

B.3 Privacy Guarantee

This section provides the privacy proof for BASE. The bilateral projection bases and initial core matrices are determined exclusively from the public auxiliary dataset before private training, while the bilateral bases remain fixed throughout Stage 2. Therefore, the privacy guarantee is determined entirely by the clipping and perturbation of the Stage-2 joint subspace gradients.

Theorem 3 (Privacy Guarantee of BASE)**.** *Assume that the bilateral bases $\{P_{\text{out}}^{(m)}, P_{\text{in}}^{(m)}\}_{m=1}^M$ and initial cores $\{S_0^{(m)}\}_{m=1}^M$ are determined exclusively from $\mathcal{D}_{\text{aux}}$ before Stage 2, and that the bilateral bases remain fixed throughout private training. At each private optimization step, suppose that the joint per-example subspace gradient $g_{i,t}^{\text{sub}}$ is clipped to have ℓ_2 norm at most C, and Gaussian noise with noise multiplier σ is applied as in standard DP-SGD. Then Stage 2 has the same privacy guarantee as standard DP-SGD under the same sampling scheme, clipping threshold, noise multiplier, and number of private steps. Consequently, for any target $\delta > 0$, BASE satisfies (ε, δ)-differential privacy with respect to $\mathcal{D}_{\text{priv}}$, where ε is computed by the adopted privacy accountant.*

Proof. Stage 1 accesses only the public auxiliary dataset $\mathcal{D}_{\text{aux}}$ and does not access any example in $\mathcal{D}_{\text{priv}}$. It therefore incurs no privacy cost with respect to the private dataset. All quantities transferred from Stage 1, including the bilateral bases and initial core matrices, depend only on public data and are fixed before private training begins.

At private step t, let $G_{i,t}^{(m)}$ denote the per-example gradient of the effective weight of target module m. Its module-wise subspace gradient is

$$Z_{i,t}^{(m)} = \left(P_{\text{out}}^{(m)}\right)^\top G_{i,t}^{(m)} P_{\text{in}}^{(m)}.$$

BASE vectorizes and concatenates the module-wise subspace gradients to form the joint per-example subspace gradient:

$$g_{i,t}^{\text{sub}} = \text{Concat}\left(\left\{\text{vec}\left(Z_{i,t}^{(m)}\right)\right\}_{m=1}^M\right) \in \mathbb{R}^R, \qquad R = \sum_{m=1}^M k_{\text{out}}^{(m)} k_{\text{in}}^{(m)}. \tag{51}$$

By the norm-preserving property of vectorization,

$$\left\|g_{i,t}^{\mathrm{sub}}\right\|_2^2 = \sum_{m=1}^{M} \left\|Z_{i,t}^{(m)}\right\|_F^2 . \tag{52}$$

Thus, globally clipping the joint subspace gradient applies one shared clipping factor to all module-wise subspace gradients:

$$\bar{g}_{i,t}^{\mathrm{sub}} = g_{i,t}^{\mathrm{sub}} \cdot \min\left(1, \frac{C}{\left\|g_{i,t}^{\mathrm{sub}}\right\|_2}\right). \tag{53}$$

Every clipped per-example joint gradient therefore has ℓ_2 norm at most C.

For a Poisson-sampled minibatch $\mathcal{B}_t$, BASE produces the privatized joint subspace gradient

$$\widetilde{g}_t^{\mathrm{sub}} = \frac{1}{|\mathcal{B}_t|}\left(\sum_{z_i \in \mathcal{B}_t} \bar{g}_{i,t}^{\mathrm{sub}} + \xi_t\right), \qquad \xi_t \sim \mathcal{N}\left(0, \sigma^2 C^2 I_R\right). \tag{54}$$

This is the standard DP-SGD mechanism applied to the joint subspace gradients. Under Poisson subsampling with sampling rate q, composing this mechanism over T private optimization steps yields the same privacy loss as standard DP-SGD with sampling rate q, clipping threshold C, noise multiplier σ, and T steps. The resulting value of ε is obtained using the adopted privacy accountant.

After $\widetilde{g}_t^{\mathrm{sub}}$ is produced, BASE only unflattens it into module-wise privatized subspace gradients, updates the optimizer states and core matrices, and maps the resulting updates through the fixed bilateral bases. These operations depend only on the privatized gradients, the fixed public quantities, and previous privatized outputs. They are therefore post-processing and incur no additional privacy loss.

Since Stage 1 incurs no privacy cost with respect to $\mathcal{D}_{\mathrm{priv}}$ and Stage 2 follows the standard DP-SGD privacy mechanism, the complete BASE procedure satisfies (ε, δ)-differential privacy with respect to $\mathcal{D}_{\mathrm{priv}}$. □

C Additional Experiments

This section provides additional experimental results that complement the main empirical analysis. These experiments examine the sensitivity of BASE to bilateral subspace ranks and target-module selection.

C.1 Rank Sensitivity

Tables 5, 6, and 7 report the rank-sensitivity analysis of BASE on RoBERTa-base, RoBERTa-large, and Qwen2.5-1.5B, respectively. Across model scales and tasks, BASE remains stable over a range of symmetric and asymmetric bilateral ranks, while increasing either k_{out} or k_{in} does not consistently improve utility.

The RoBERTa results further reveal a consistent task-dependent pattern. For both RoBERTa-base and RoBERTa-large, the compact 8–8 configuration performs best on SST-2, whereas 16–16 achieves the best results on QNLI, MNLI, and QQP among the evaluated configurations. Moreover, asymmetric rank choices do not exhibit a systematic advantage over their symmetric counterparts, and further increasing the ranks does not necessarily yield additional gains. Similar sensitivity is observed on SAMSum, indicating that the preferred bilateral rank depends on the downstream task rather than following a single universally optimal configuration.

Based on these observations, we use compact symmetric ranks selected for each task in the main experiments, with the detailed configurations reported in Appendix D.

C.2 Target-Module Selection

Table 8 reports the target-module ablation results. BASE performs consistently across different target-module choices, while the default Attention-QKVO setting achieves the best results on both

Table 5: Rank ablation results of BASE on RoBERTa-base under $\varepsilon = 4.0$. For each task, we fix the selected Stage-2 learning rate: $5\mathrm{e}{-4}$ for SST-2 and QNLI, and $1\mathrm{e}{-4}$ for MNLI and QQP. Results are averaged over three random seeds. The best result in each column is shown in bold.

k_{out}	k_{in}	SST-2	QNLI	MNLI		QQP	
		Acc.	Acc.	Matched	Mismatched	Acc.	F1
4	4	91.93	87.85	81.83	81.90	84.55	79.85
4	8	92.16	87.95	82.04	82.29	84.73	80.07
8	4	92.05	88.05	82.07	82.28	84.69	79.95
8	8	**92.78**	88.41	83.18	83.48	85.37	80.83
16	16	92.24	**88.80**	**83.92**	**83.99**	**86.97**	**82.80**
16	8	92.51	88.63	83.34	83.54	86.46	81.97
8	16	92.62	88.53	83.26	83.53	86.44	81.92

Table 6: Rank ablation results of BASE on RoBERTa-large under $\varepsilon = 0.5$. Results are averaged over three random seeds. The best result in each column is shown in bold.

k_{out}	k_{in}	SST-2 Acc.	QNLI Acc.	MNLI Matched	QQP Acc.
4	4	93.29	89.42	86.53	85.64
4	8	93.56	90.63	87.68	86.39
8	4	93.42	90.59	87.49	85.47
8	8	**93.69**	90.74	87.76	85.64
16	16	93.04	**91.68**	**88.50**	**86.96**
16	8	93.51	90.94	87.39	85.18
8	16	93.27	90.39	86.94	85.23
32	32	93.15	90.67	87.49	86.17
16	32	93.34	90.86	87.26	85.34
32	16	93.04	90.58	87.59	85.42

QNLI and MNLI and the highest average score. Adding FFN layers to the target set does not provide further gains and instead results in a slight performance decrease, supporting Attention-QKVO as an effective default choice.

D Experimental Details and Hyperparameter Configurations

This section reports the hyperparameter configurations used in the experiments. For the competing DP baselines, the learning rate, clipping threshold, and number of epochs are selected separately for each backbone model. For DP-LoRA, the LoRA rank is set to 16; for DP-Adapter, the reduction factor is set to 48; and for DP-SFT, the subspace rank is set to 32. For BASE, the bilateral ranks and stage-specific optimization hyperparameters are selected for each task.

Unless otherwise specified, all methods use Adam with a logical batch size of 32, a linear learning-rate scheduler with a warmup ratio of 0.1, and a weight decay of 0.01. Task- and method-specific settings are detailed in the following subsections.

Public auxiliary protocol. Unless otherwise specified, we randomly split the original training set into 10% public auxiliary data $\mathcal{D}_{\text{aux}}$ and 90% private data $\mathcal{D}_{\text{priv}}$. All DP methods use the same $\mathcal{D}_{\text{aux}}$ for non-private warm-up before private fine-tuning on $\mathcal{D}_{\text{priv}}$. For the DP baselines, Stage 2 continues from the corresponding warm-up model, whereas BASE uses the public warm-up trajectory to construct the bilateral bases and reloads the original pretrained backbone before private optimization. Thus, all methods have access to the same public and private data partitions, and privacy accounting applies only to $\mathcal{D}_{\text{priv}}$.

Evaluation and reporting. For SST-2 and QNLI, we report classification accuracy. For MNLI, we report accuracy on both the matched and mismatched validation sets, and for QQP, we report both

Table 7: Rank ablation results of BASE on Qwen2.5-1.5B with SAMSum under $\varepsilon = 4.0$. Results are averaged over three random seeds. The best result in each column is shown in bold.

k_{out}	k_{in}	ROUGE-1	ROUGE-2	ROUGE-L
4	4	0.4711	0.2379	0.3968
8	8	0.4775	0.2412	0.3969
4	8	0.4763	0.2435	0.4058
8	4	0.4759	0.2464	0.4049
8	16	0.4775	0.2476	0.4062
16	8	0.4762	0.2498	0.4054
16	16	**0.4821**	**0.2529**	**0.4133**

Table 8: Target-module ablation on RoBERTa-large under $\varepsilon = 0.5$. Attention-QKVO denotes the default setting that applies BASE to the query, key, value, and attention-output projections. Attention-QV applies BASE only to the query and value projections. FFN applies BASE to the intermediate and output feed-forward layers. All target-module settings use the same task-specific bilateral rank as the main experiments: $k_{\text{out}} = k_{\text{in}} = 16$ for QNLI and MNLI. Avg. denotes the arithmetic mean over QNLI and MNLI. Bold and underline indicate the best and second-best results in each column.

Target modules	QNLI	MNLI	Avg.
Attention-QV	90.23	87.10	88.67
FFN	91.36	88.43	89.90
Attention-QKVO (default)	**91.68**	**88.50**	**90.09**
Attention-QKVO + FFN	<u>91.48</u>	<u>88.45</u>	<u>89.97</u>

accuracy and F1 score. For SAMSum, we report ROUGE-1, ROUGE-2, and ROUGE-L scores. For Dolly-15k instruction tuning, we evaluate the fine-tuned models using IFEval and report strict prompt-level accuracy and strict instruction-level accuracy. Unless otherwise specified, results are reported as mean $\pm$ standard deviation over three random seeds. For the Avg. column in the GLUE results, the MNLI score is first computed as the arithmetic mean of matched and mismatched accuracy, while the QQP score is computed as the arithmetic mean of accuracy and F1. The final Avg. is then computed as the arithmetic mean over SST-2, QNLI, MNLI, and QQP.

Privacy settings. For all differentially private experiments, the privacy parameter δ is fixed according to the dataset. Specifically, we set $\delta = 1 \times 10^{-5}$ for SST-2, QNLI, SAMSum, and Dolly-15k, and $\delta = 1 \times 10^{-6}$ for MNLI and QQP. Private training uses Poisson subsampling with sampling rate q, and the cumulative privacy loss over all DP-SGD steps is computed using a Rényi differential privacy (RDP) accountant. The noise multiplier σ is calibrated according to the target (ε, δ), sampling rate q, and total number of optimization steps. The privacy budget ε is set to either 0.5 or 4.0 in the main experiments.

D.1 Scalability of DP-SFT on Billion-Scale Models

As discussed in the experimental section, we do not report DP-SFT results for Qwen2.5-1.5B because its global subspace representation incurs prohibitive memory costs at this model scale. DP-SFT represents the optimization subspace using a global projection matrix $\mathbf{P} \in \mathbb{R}^{d \times k}$, where d is the number of parameters in the full model and k is the subspace dimension. Consequently, storing the projection matrix alone requires $O(dk)$ memory.

For Qwen2.5-1.5B, with approximately $d = 1.54 \times 10^9$ parameters, the storage required by $\mathbf{P}$ under FP32 is

$$1.54 \times 10^9 \times 32 \times 4 \approx 197.1 \text{ GB}, \tag{55}$$

when $k = 32$, and

Table 9: Hyperparameter configurations for BASE. The table reports the output-side rank k_{out}, input-side rank k_{in}, and Stage-1 and Stage-2 optimization hyperparameters for each task.

Hyperparameter	SST-2	QNLI	MNLI	QQP	SAMSum	Dolly-15k
k_{out}	8	16	16	16	16	16
k_{in}	8	16	16	16	16	16
Stage-1 Learning Rate	5e−5	5e−5	5e−5	5e−5	2e−5	2e−5
Stage-1 Epochs	3	3	3	3	3	3
Stage-2 Learning Rate	5e−4	5e−4	1e−4	1e−4	5e−4	5e−4
Stage-2 Clip Threshold	1.0	1.0	1.0	1.0	2.0	2.0
Stage-2 Epochs	3	3	3	3	5	5

Table 10: Optimization hyperparameters for the DP baselines. Warm-up LR and epochs denote the learning rate and number of epochs used during public auxiliary-data warm-up, while DP LR, Clip C, and DP epochs denote the learning rate, clipping norm, and number of epochs used during private DP-SGD fine-tuning. Datasets grouped in the same row share the same configuration.

Method	Dataset(s)	Warm-up LR	Warm-up epochs	DP LR	Clip C	DP epochs
(a) RoBERTa-base						
DP-Full	All tasks	5×10^{-5}	3	5×10^{-5}	1.0	3
DP-LoRA	SST-2 / MNLI	5×10^{-5}	3	5×10^{-5}	1.0	3
	QNLI / QQP	5×10^{-5}	3	1×10^{-4}	1.0	3
DP-SFT	All tasks	5×10^{-5}	3	1×10^{-4}	0.5	3
DP-Adapter	SST-2 / QNLI	5×10^{-5}	3	5×10^{-4}	1.0	3
	MNLI / QQP	5×10^{-5}	3	1×10^{-4}	1.0	3
DP-BitFit	SST-2 / QNLI	5×10^{-5}	3	1×10^{-3}	0.5	3
	MNLI / QQP	5×10^{-5}	3	5×10^{-4}	0.5	3
(b) RoBERTa-large						
DP-Full	SST-2	2×10^{-5}	5	5×10^{-5}	1.0	3
	QNLI	1×10^{-5}	5	5×10^{-5}	1.0	3
	MNLI	5×10^{-5}	3	5×10^{-5}	1.0	3
	QQP	1×10^{-5}	5	5×10^{-5}	1.0	3
DP-LoRA	SST-2 / MNLI	5×10^{-5}	3	5×10^{-5}	1.0	3
	QNLI / QQP	5×10^{-5}	3	1×10^{-4}	1.0	3
DP-SFT	All tasks	5×10^{-5}	3	1×10^{-4}	0.5	3
DP-Adapter	SST-2 / QNLI	5×10^{-5}	3	5×10^{-4}	1.0	3
	MNLI / QQP	5×10^{-5}	3	1×10^{-4}	1.0	3
DP-BitFit	SST-2 / QNLI	5×10^{-5}	3	1×10^{-3}	0.5	3
	MNLI / QQP	5×10^{-5}	3	5×10^{-4}	0.5	3
(c) Qwen2.5-1.5B						
DP-Full	SAMSum / Dolly-15k	5×10^{-5}	3	5×10^{-5}	10.0	5
DP-LoRA	SAMSum / Dolly-15k	5×10^{-5}	3	5×10^{-4}	10.0	5
DP-Adapter	SAMSum / Dolly-15k	5×10^{-5}	3	5×10^{-4}	10.0	5
DP-BitFit	SAMSum / Dolly-15k	5×10^{-5}	3	5×10^{-4}	1.0	5

$$1.54 \times 10^{9} \times 64 \times 4 \approx 394.2 \text{ GB}, \tag{56}$$

when $k = 64$. These estimates account only for the projection matrix itself and therefore constitute a conservative lower bound on the memory requirement. Additional memory is required for the model parameters, optimization states, trajectory snapshots, and intermediate storage associated with subspace construction, including the SVD computation.

In contrast, BASE does not construct a global projection matrix over the vectorized parameter space. Instead, it learns compact input- and output-side bases independently for each target linear module

and optimizes only the corresponding low-dimensional core matrices during private fine-tuning. This module-wise bilateral formulation avoids the $O(dk)$ global projection matrix and enables BASE to remain practical on Qwen2.5-1.5B, for which DP-SFT becomes prohibitively memory intensive.