Dual-Priv Pruning : Efficient Differential Private Fine-Tuning in Multimodal Large Language Models

Qianshan Wei[1]∗, Jiaqi Li[1]∗, Zihan You[2]∗, Yi Zhan[3] , Kecen Li[4] , Jialin Wu[5] , Xinfeng Li[5,6]
Hengjun Liu[7] , Yi Yu[6] , Bin Cao[4] , Yiwen Xu[8] , Yang Liu[6] , Guilin Qi[1]†

[1]School of Cyber Science and Engineering, Southeast University, Nanjing, China
[2]School of Automation, Southeast University, Nanjing, China
[3]School of Computer Science, Peking University, Beijing, China
[4]Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China
[5]Zhejiang University, Hangzhou, China
[6]Nanyang Technological University, Singapore
[7]Electrical Engineering and Information Technology, Technische Universität Chemnitz, Chemnitz, Germany
[8]University of California, Los Angeles (UCLA), Los Angeles, CA, USA

Abstract

Differential Privacy (DP) is a widely adopted technique, valued for its effectiveness in protecting the privacy of task-specific datasets, making it a critical tool for large language models. However, its effectiveness in Multimodal Large Language Models (MLLMs) remains uncertain. Applying Differential Privacy (DP) inherently introduces substantial computation overhead, a concern particularly relevant for MLLMs which process extensive textual and visual data. Furthermore, a critical challenge of DP is that the injected noise, necessary for privacy, scales with parameter dimensionality, leading to pronounced model degradation; This trade-off between privacy and utility complicates the application of Differential Privacy (DP) to complex architectures like MLLMs. To address these, we propose **Dual-Priv Pruning**, a framework that employs two complementary pruning mechanisms for DP fine-tuning in MLLMs: (i) *visual token pruning* to reduce input dimensionality by removing redundant visual information, and (ii) *gradient-update pruning* during the DP optimization process. This second mechanism selectively prunes parameter updates based on the magnitude of noisy gradients, aiming to mitigate noise impact and improve utility. Experiments demonstrate that our approach achieves competitive results with minimal performance degradation. In terms of computational efficiency, our approach consistently utilizes less memory than standard DP-SGD. While requiring only 1.74% more memory than zeroth-order methods which suffer from severe performance issues on A100 GPUs, our method demonstrates leading memory efficiency on H20 GPUs. To the best of our knowledge, we are the first to explore DP fine-tuning in MLLMs. Our code is coming soon.

1 Introduction

Large Language Models (LLMs) [35, 43, 52] have showcased remarkable proficiency in natural language processing, driving their widespread adoption in downstream tasks [54], and Multimodal Large Language Models (MLLMs) [2, 27, 45]extend the power of LLMs by integrating text and visual data, opening up possibilities for applications that require understanding across different

---

∗These authors contributed equally to this work.
†Corresponding author.

modalities. However, both models are easy to risk leaking sensitive information during training [8, 32]. Differential Privacy [9] (DP) , the technology for providing privacy guarantees that limit the ability to infer whether a data point was used in the training process of a model by observing its output. This technology is typically achieved by injecting noise during training processes, limiting the discernible impact of single data point. The degree of privacy guarantee is tuned using a privacy budget (ϵ), where stronger privacy guarantee (lower ϵ) generally comes at the cost of adding more noise and degrading model performance. The inherent trade-off between privacy and utility presents a significant challenge, particularly when applying DP to large and complex models like LLMs, since the necessary noise often scales with parameter dimensionality. Prior works [11, 25, 28, 50] have shown that LLMs with hundreds of millions of parameters can be effectively and efficiently fine-tuned to yield models with high performance under modest privacy leakage.

However, it remains unclear whether such conclusions of LLMs are transferable to MLLMs. Similar to unimodal models, DP also face challenges under MLLMs. The first is **computation consumption**. This challenge is exacerbated in MLLMs, which rely on a large number of visual tokens (e.g., 197 tokens per image in CLIP-ViT [36] or hundreds in LLaVA [27]) to represent detailed visual information, significantly increasing computation demands. Recent work [25]introduced "ghost clipping" to address the computation overhead of DP-SGD in LLMs. While ghost clipping reduces computation overhead by leveraging the sequential structure of text, its reliance on this sequentiality renders it unsuitable for image features, as MLLMs process these features as non-sequential data within their multimodal components. Zeroth-order methods (e.g., DP-ZO [42]) also aim to reduce computation overhead by avoiding explicit gradient calculations. However, these methods introduce severe convergence issues. For instance, DP-ZO required more training steps (75k vs 200) than standard DP-SGD to achieve comparable performance on SQuAD [42], making this gradient-free approach prohibitively slow for practical MLLM training. Another challenge is **model degradation**. Differential privacy introduces noise to safeguard data privacy, but this noise perturbs the gradient signals during training, leading to performance degradation. In MLLMs, DP noise scales with parameter dimensionality, overwhelming gradient signals in high-dimensional layers and necessitating more iterations to stabilize optimization, as noted in foundational work on DP-SGD [1].

To tackle these challenges, we introduce **Dual-Priv Pruning**, a novel DP finetuning approach tailored for MLLMs. Our approach integrates two complementary pruning mechanisms designed to work in concert, addressing these issues from both the input representation and the optimization process. The first key pruning mechanism focuses on optimizing the visual input stream prior to training: it employs an attention-based mechanism to identify and prune redundant visual tokens, thereby substantially reducing the input dimensionality and subsequent computational demands. The less critical visual information pruned in this manner is then fused into some compact contextual representations, to which a calibrated heuristic noise is added. This step aims to preserve essential global context while further alleviating the processing load for the differential privacy mechanism. The second core pruning mechanism refines the differential private fine-tuning process itself. While adhering to the standard DP-SGD framework for rigorous noise addition to guarantee privacy, Dual-Priv Pruning introduces a *gradient-update pruning* technique. This technique analyzes the noisy gradients resulting from DP noise injection. It then selectively applies these gradients for parameter updates only to those blocks where the underlying signal is deemed sufficiently strong and reliable to overcome the obfuscating effect of the DP noise, thereby preserving model utility and stabilizing training. Dual-Priv Pruning offers a robust solution. As the first work to explore DP finetuning specifically tailored for MLLMs, our method achieves a superior privacy-utility trade-off and enhanced computational efficiency, delivering competitive performance even under stringent privacy budgets.

We summarize our main contributions as follows: **(1)** We pioneer the integration of DP into the domain of MLLMs, addressing a critical research gap in privacy-preserving multimodal learning. **(2)** We introduce a novel privacy-aware visual pruning mechanism that significantly reduces computational overhead by optimizing visual inputs, thereby creating more favorable conditions for subsequent DP fine-tuning. **(3)** We propose an DP-compatible gradient-update pruning strategy that intelligently applies noisy gradients to mitigate the adverse effects of DP noise on model performance, thereby enhancing utility while maintaining strong privacy guarantees. **(4)** Extensive experiments demonstrate that our Dual-Priv Pruning achieves robust privacy protection, substantial memory reduction, and competitive performance on diverse vision-language tasks, even under stringent privacy budgets.

2 Related Work

Differential Privacy (DP) [9] ensures privacy guarantees by limiting the ability to infer whether a data point was used in the training process of a model, making it a cornerstone for privacy-preserving learning. In the area of computer vision, [41] developed DP methods for image classification by adding noisy priors, achieving strong privacy-utility trade-offs, and [30] applied DP to video recognition, enforcing video-level differential privacy through clip-based classification models. In natural language processing, [31] trained recurrent language models with DP, reducing risks of data memorization. For LLMs, [24] demonstrated DP fine-tuning but noted challenges with utility degradation due to noise, while [18] showed that public pre-training followed by private fine-tuning can alleviate some performance losses. Memory-efficient techniques, such as "ghost-clipping" [25], optimize DP-SGD for LLMs but rely on text-specific assumptions, limiting their applicability to multimodal settings. Zeroth-order optimization [42] offers an alternative for LLMs by avoiding gradient instantiation, but it suffers from too long training times. Other efforts to improve DP include manipulating gradients, such as GIP [48] that perturbed individual gradient indices, though its privacy analysis clarity was questioned; In contrast, our gradient-update pruning operates as a post-processing step on entire noised logical parameter blocks, simplifying privacy analysis and aligning with PEFT. In multimodal learning, [15] introduced DP to CLIP training, protecting vision-language data, and [51] proposed low-rank reparametrization for scalable private learning, applicable to multimodal tasks. Additionally, [17] applied DP to medical image, emphasizing privacy in sensitive domains. Despite these advances, no prior work has explored DP fine-tuning for MLLMs, which face unique challenges due to cross-modal interactions and massive length visual tokens. Existing methods, do not address the memory demands and model degradation of MLLMs, a gap that our work to addresses.

Multimodal Large Language Models (MLLMs) integrate visual and textual modalities to solve a wide range of tasks. Flamingo [3] introduced a query-based cross-attention mechanism to enable multimodal interactions, while BLIP-2 [23] proposed the lightweight Q-Former to enhance efficiency. InstructBLIP [7] further aligned models with user intent via instruction tuning across diverse datasets. LLaVA [27] improved visual understanding using curated training data, while subsequent efforts such as Qwen-VL [4] and CogVLM [46] introduced advanced training strategies and modular visual expert systems to boost performance. A major challenge in MLLMs is the redundancy of visual tokens, which significantly increases memory and computational costs [6]. Recent work addresses this inefficiency: FastVLM [44] prunes tokens based on attention scores, and VisionZip [49] identifies contextual tokens that retain global semantics (e.g., background information). Visual token redundancy offers a promising avenue for DP in MLLMs. Pruning low-importance tokens reduces sensitive data exposure. We leverage this property to enable even source-level privacy protection and efficient DP fine-tuning.

3 Preliminaries

3.1 Differential Privacy

Differential privacy (DP) [9] provides a rigorous framework to safeguard sensitive data by ensuring that model outputs remain statistically indistinguishable for datasets differing by a single record. This guarantee inherently limits the ability of inferring individual record participation, mitigating risks such as membership inference attacks [39]. A hallmark of DP is its robustness to post-processing: if an algorithm $\mathcal{A}$ satisfies (ϵ, δ)-DP, any function $f \circ \mathcal{A}$ preserves the same (ϵ, δ)-DP guarantee.

Definition 3.1 $((\epsilon, \delta)$-Differential Privacy)**.** *A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if, for any two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$, differing by one record, and any set of outputs $S \subseteq Range(\mathcal{A})$, the following holds:*

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

where $\epsilon \geq 0$ is the privacy budget, controlling the strength of the privacy guarantee, and $\delta \in [0, 1)$ is a small failure probability.

In the context of fine-tuning MLLMs, two datasets $\mathcal{D}$ and $\mathcal{D}'$ are defined as neighboring if one can be obtained from the other by adding or removing a single image-text pair. The application of DP in iterative training (introduced in Section 3.1.1), relies on fundamental mechanisms and accounting principles. The Gaussian Mechanism (detailed in Fact A.1) is employed to add noise. To manage the overall privacy loss across multiple iterations, privacy accounting techniques like Rényi Differential Privacy (RDP) (detailed in Fact A.2) are utilized. These principles are central to the DP application.

3.1.1 Differentially Private SGD

Differentially Private Stochastic Gradient Descent (DP-SGD) [1] adapts SGD to ensure the trained model parameters $\theta \in \mathbb{R}^d$ satisfy an overall (ϵ, δ)-DP guarantee with respect to $\mathcal{D}_{\text{train}}$. In each iteration k, for a minibatch ξ_k of size m sampled with probability $q = m/N$: First, per-sample gradients $g_i = \nabla_\theta \mathcal{L}(\theta_{k-1}, (\mathcal{I}_i, \mathcal{T}_i))$ are computed for each $i \in \xi_k$. Second, to bound sensitivity, the L_2 norm of each gradient g_i is clipped using a threshold C: $\hat{g}_i = g_i / \max(1, \|g_i\|_2/C)$. This ensures $\|\hat{g}_i\|_2 \leq C$, thereby limiting the influence of any single sample and resulting in an L_2 sensitivity of $\Delta f = C/m$ for the subsequent average gradient (details in Appendix B). Third, these clipped gradients are aggregated by averaging: $\bar{g} = \frac{1}{m} \sum_{i \in \xi_k} \hat{g}_i$. Finally, calibrated Gaussian noise is added to this average gradient before updating:

$$\theta_k = \theta_{k-1} - \eta \cdot \left(\bar{g} + \mathcal{N}(0, \sigma^2 C^2 I_d / m^2) \right). \tag{2}$$

The hyperparameters C (clipping norm) and σ (noise multiplier) control the trade-off between privacy and utility. The appropriate value for σ is determined based on the overall privacy budget (ϵ, δ), total training steps, and sampling rate, typically using privacy accounting methods like RDP (Fact A.2).

3.2 Problem Definition: Differentially Private Fine-Tuning of MLLMs

Our work focuses on fine-tuning a pre-trained MLLM $\mathcal{M}_\theta$ with parameters $\theta \in \mathbb{R}^d$. The fine-tuning is performed on a private dataset $\mathcal{D}_{\text{fine}} = \{(\mathcal{I}_i, \mathcal{T}_i)\}_{i=1}^N$, where each pair consists of an image $\mathcal{I}_i$ and a text sequence $\mathcal{T}_i = \{w_1, \ldots, w_i\}$. The primary objective is to adapt $\mathcal{M}_\theta$ to downstream vision-language tasks by learning parameters θ_{fine} that exhibit **high utility**. This utility is typically achieved by minimizing an empirical risk, often the negative log-likelihood loss, over the $\mathcal{D}_{\text{fine}}$.

A crucial and defining requirement for this process is that it must adhere to a **strict (ϵ, δ)-Differential Privacy (DP) guarantee** (Definition 3.1) with respect to $\mathcal{D}_{\text{fine}}$. This requires the learning algorithm $\mathcal{A}$ to generate θ_{fine} from $\mathcal{D}_{\text{fine}}$ and θ under (ϵ, δ)-DP guarantees. The core problem can be summarized as finding parameters θ_{fine} that balance utility and privacy, as formally stated below:

Problem Formulation

Objective: Minimize the empirical risk on the private dataset $\mathcal{D}_{\text{fine}}$:

$$\mathcal{L}(\theta, \mathcal{D}_{\text{fine}}) := \frac{1}{N} \sum_{i=1}^N \left(- \sum_{t=1}^{T_i} \log P_{\mathcal{M}_\theta}(w_{i,t} | \mathcal{I}_i, w_{i,1}, \ldots, w_{i,t-1}) \right) \tag{3}$$

The learning algorithm $\mathcal{A}$ producing θ_{fine} from $\mathcal{D}_{\text{fine}}$ must be (ϵ, δ)-Differentially Private:

$$\text{Find } \theta_{\text{fine}} \approx \underset{\theta \in \mathbb{R}^d}{\arg\min} \, \mathcal{L}(\theta, \mathcal{D}_{\text{fine}}) \quad \text{s.t.} \quad \mathcal{A}(\mathcal{D}_{\text{fine}}) \text{ is } (\epsilon, \delta)\text{-DP.} \tag{4}$$

---

4 Method

We introduce **Dual-Priv Pruning**, the first framework for differential private (DP) fine-tuning of MLLMs, designed to optimize the privacy-utility trade-off. **Mechanism 1** performs attention-based token pruning and fusion to transform the visual input into a compact representation $\mathcal{V}'$. **Mechanism 2** applies (ϵ, δ)-DP to the trainable parameters θ_{train} using DP-SGD (Section 3.1.1), enhanced with a *gradient-update pruning* strategy to improve utility. This provides formal (ϵ, δ)-DP guarantees for the entire pipeline. Further details and motivations are in Appendix E and Appendix F.

4.1 Mechanism 1: Visual Token Pruning and Fusion

This initial stage reduces the computation cost associated with long visual token sequences before the differential private fine-tuning process begins. It consists of identifying and retaining the most important visual tokens based on attention, followed by merging the remaining tokens and applying noise prior. This stage is not designed to provide the formal DP guarantee.

Dominant Token Selection via CLS Attention. For an input image $\mathcal{I}_i$, the vision encoder extracts an initial set of n visual tokens $\mathcal{V} = \{v_{cls}, v_1, \ldots, v_{n-1}\}$, including a class token v_{cls} and $n-1$ patch tokens, where $v_j \in \mathbb{R}^d$. We hypothesize that tokens receiving significant attention from the class token ([CLS]) include the most critical global information.
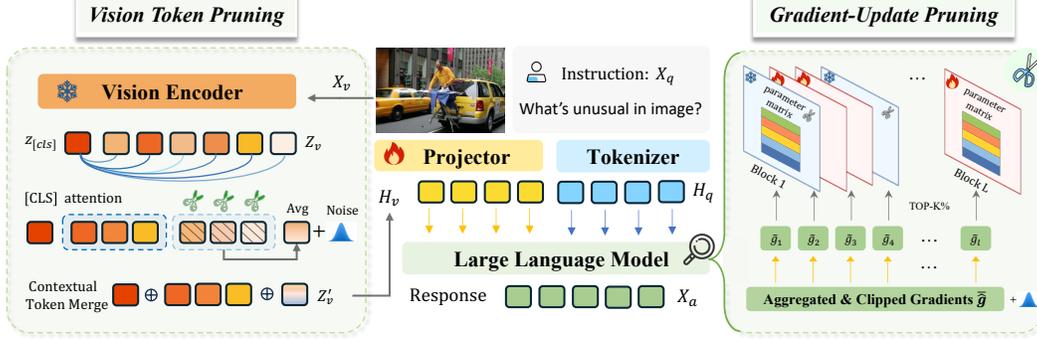
Figure 1: Overview of our Dual-Priv Pruning. **(Left)**: Visual Token Pruning and Fusion. Using [CLS] attention, dominant tokens are selected; less important ones are averaged with heuristic noise. **(Right)**: DP Fine-tuning with gradient pruning. Noise is added to gradients in LLM blocks, and updates are selectively applied based on noisy gradient magnitude. Frozen parameters remain unchanged.

To identify these dominant tokens, we first compute the multi-head self-attention maps within a selected layer of the vision encoder. The attention map for a single head h is given by:

$$S_h = \text{Softmax}\left(\frac{Q_h K_h^\top}{\sqrt{D_h}}\right) \in \mathbb{R}^{n \times n}, \tag{5}$$

where Q_h, K_h are the query and key matrices, and D_h is the head dimension. We average these maps across all H heads to get an aggregated attention map $S_{\text{avg}} \in \mathbb{R}^{n \times n}$:

$$S_{\text{avg}} = \frac{1}{H} \sum_{h=1}^{H} S_h. \tag{6}$$

The importance score s_j for each patch token v_j ($j \in \{1, \ldots, n-1\}$) is then determined by the attention receives from the [CLS] token in the aggregated map. We select the K patch tokens with the highest importance scores s_j as the dominant patch tokens $\mathcal{V}_d = \{v_j \mid s_j \text{ is among the top K scores}\}$. The class token v_{cls} is always retained. The remaining patch tokens form the non-dominant set $\mathcal{V}_{nd}$.

Contextual Token Fusion and Heuristic Noise. To preserve the visual context features from $\mathcal{V}_{nd}$ while reducing sequence length, we uniformly randomly select tokens $v_{\text{center},i}$ from $\mathcal{V}_{nd}$ as cluster centers and enhance their representation based on cosine similarity with the remaining non-dominant tokens. Subsequently, Gaussian noise scaled by σ_{fuse}^2 is heuristically applied to the enhanced v_{center}, producing the fused contextual tokens c, as defined in the following formula:

$$\mathbf{c} = \begin{bmatrix} v_{\text{center},1} + \frac{1}{|\mathcal{C}_1|} \sum_{v_j \in \mathcal{C}_1} v_j \\ v_{\text{center},2} + \frac{1}{|\mathcal{C}_2|} \sum_{v_j \in \mathcal{C}_2} v_j \\ \vdots \\ v_{\text{center},k} + \frac{1}{|\mathcal{C}_k|} \sum_{v_j \in \mathcal{C}_k} v_j \end{bmatrix} + \mathcal{N}\left(0, \sigma_{\text{fuse}}^2 I_{kd}\right), \tag{7}$$

where $\mathcal{C}_i$ is the set of non-dominant tokens assigned to the i-th cluster based on similarity:

$$\mathcal{C}_i = \left\{ v_j \in \mathcal{V}_{nd} \mid i = \arg\max_l \text{sim}(v_j, v_{\text{center},l}) \right\}, \quad i = 1, 2, \ldots, k. \tag{8}$$

The noise adding process serves as a form of regularization or stochasticity injection; A key aspect of our design is to maintain consistency with the noise introduced by the DP mechanism in the subsequent stage. Therefore, the variance of this heuristic noise, σ_{fuse}^2, is set to be equivalent to the variance of the Gaussian noise added per step in the DP optimization process (Mechanism 2, Section 4.2). It does not contribute to the formal (ϵ, δ)-DP guarantee derived in Mechanism 2. The final set of visual tokens passed to the MLLM for the DP fine-tuning stage is $\mathcal{V}' = \{v_{cls}\} \cup \mathcal{V}_d \cup \{C\}$, which has a significantly reduced size of $K + |C| + 1$ tokens.

4.2 Mechanism 2: DP Fine-tuning with gradient-update pruning

This core mechanism performs the (ϵ, δ)-differential private fine-tuning of the trainable parameters θ_{train} (e.g., LoRA matrices [14]), leveraging the pruned visual inputs $(\mathcal{V}', \mathcal{T})$ from Mechanism 2.

Our approach builds upon DP-SGD (Section 3.1.1) but introduces a **post-noise adaptive update** mechanism designed to enhance utility without compromising the privacy guarantee.

The process within each training iteration t begins with standard DP-SGD procedures. For a minibatch ξ_t of size m, we first compute per-sample gradients $g_i = \nabla_{\theta_{\text{train}}}\mathcal{L}(\theta_{t-1}; (\mathcal{V}'_i, \mathcal{T}_i))$. To bound the influence of individual samples, we clip the L_2 norm of each gradient using a threshold C: $\hat{g}_i = g_i / \max(1, \|g_i\|_2 / C)$. These clipped gradients are then averaged across the minibatch to produce $\hat{\bar{g}} = \frac{1}{m}\sum_{i \in \xi_t}\hat{g}_i$. The crucial step for ensuring differential privacy follows. Gaussian noise is added *unconditionally* to the entire aggregated gradient vector:

$$\tilde{g} = \hat{\bar{g}} + \mathcal{N}\left(0, \frac{\sigma^2 C^2}{m^2}I_{d_{\text{train}}}\right). \tag{9}$$

Here, d_{train} is the dimensionality of θ_{train}, and the noise multiplier σ is determined by the overall privacy budget (ϵ, δ), number of steps T, and sampling rate q via privacy accounting (Fact A.2). At this point, the noisy gradient $\tilde{g}$ is an (ϵ_t, δ_t)-differentially private quantity for the current step. Our mechanism diverges from standard DP-SGD hereafter. Instead of directly using $\tilde{g}$ for the update, we first analyze its structure and magnitude. We partition $\tilde{g}$ into components $\tilde{g}_j$ corresponding to logical parameter blocks within θ_{train} and compute the L_2 norm $N_j = \|\tilde{g}_j\|_2$ for each block.

Based on these norms, we generate a binary mask M, structured identically to θ_{train}, to selectively prune the parameter update. A block j is chosen for update (M_j remains 1): **only if its noisy gradient norm N_j is among the top K% of norms across all blocks**, otherwise M_j remains 0.

$$M_j = \mathbb{I}(N_j \in \text{Top-K\%}(\{N_1, N_2, \ldots, N_J\})), \tag{10}$$

where $\mathbb{I}(\cdot)$ is the indicator function, J is the total number of parameter blocks, and Top-K%$(\cdot)$ denotes the set of the $K\%$ largest norm values. The percentage for K% is a hyperparameter.

Finally, the model parameters are updated using the noisy gradient $\tilde{g}$, but applied selectively through the generated mask M via element-wise multiplication (Hadamard product $\odot$):

$$\theta_t = \theta_{t-1} - \eta_t \cdot (M \odot \tilde{g}). \tag{11}$$

This ensures that parameter updates are only applied to blocks where the noisy gradient signal was deemed sufficiently strong or reliable according to the gating criterion. The full step-by-step procedure is formally detailed in Appendix L.

4.3 Overall Privacy Guarantee

The (ϵ, δ)-DP guarantee of the Dual-Priv Pruning method is entirely derived from Mechanism 2 (Section 4.2). Mechanism 1 (Section 4.1) involves data preprocessing *before* the DP mechanism is applied and does not consume privacy budget. The adaptive update mechanism within Mechanism 2, constitutes post-processing on the private intermediate result $\tilde{g}$ and thus does not affect the formal (ϵ, δ)-DP guarantee (Appendix D).

5 Experiments

We conduct a comprehensive experimental evaluation of our proposed **Dual-Priv Pruning** method. Our experiments are designed to validate four core advantages of Dual-Priv Pruning: **(1)** Preserve utility, especially under strict privacy budgets ($\epsilon \leq 3$), compared to baseline methods; **(2)** Significant improvements in computation cost, highlighted by an approximate 14.34% reduction in peak GPU memory usage; and **(3)** Validated effectiveness on challenging, visual tasks, encompassing high-resolution real-world scenes and medical images, demonstrating the method's practical applicability in complex, privacy-sensitive domains. **(4)** Empirically shown to be effective against privacy attacks like Membership Inference Attacks (MIA).

5.1 Experimental Setup

Datasets. We evaluate performance by fine-tuning on the training sets and evaluating on the test sets of several vision-language benchmarks. These include standard datasets such as ScienceQA [29] (Scientific VQA), TextVQA [40] (VQA over text in images), and GQA [16] (Compositional VQA). To specifically assess scalability and robustness on complex inputs, we utilize MME-RealWorld [53], an MLLM benchmark designed for high-difficulty tasks involving high-resolution real-world images. Additionally, we incorporate two medical visual question answering dataset, PathVQA [13]and VQA-RAD [21], to further test generalization on specialized, challenging domains.

Table 1: Comparison of different methods on standard benchmarks (BS = 12). For reference, non-private performance ($\epsilon = \infty$) are included. Metrics reported are Accuracy (Acc) and Image-based Accuracy (IMG). The best results for each ϵ setting are shown in **bold**.

ϵ	DZPO				DP-SGD				Dual-Priv(ours)			
	ScienceQA		TextVQA	GQA	ScienceQA		TextVQA	GQA	ScienceQA		TextVQA	GQA
	Acc(%)	IMG	Acc(%)	Acc(%)	Acc(%)	IMG	Acc(%)	Acc(%)	Acc(%)	IMG	Acc(%)	Acc(%)
1	23.30	21.50	1.13	0.00	81.54	72.51	34.52	38.61	**84.20**	**78.43**	**34.74**	**39.06**
3	21.50	19.90	2.82	0.00	78.80	70.59	**35.64**	39.11	**82.80**	**75.98**	35.17	**39.65**
8	21.50	19.90	1.31	0.00	82.52	74.00	35.60	39.16	**85.10**	**76.47**	**35.71**	**39.78**
∞	22.16	0.98	0.95	0.00	81.10	73.53	34.89	38.92	**84.60**	**79.41**	**35.53**	**39.06**

Model & Training Strategy. We utilize LLAVA-7B [27] as our base MLLM. Specifically, for tasks in the medical domain (PathVQA, VQA-RAD, and MIA on ROCOV2), we employ Med-LLaVA[22], a LLaVA variant adapted for medical vision-language understanding. To isolate the impact of DP fine-tuning methods, we do not perform additional instruction tuning stages beyond the initial pre-training of LLAVA. Parameter-efficient fine-tuning is achieved using LoRA [14] (rank $r = 128$, scaling $\alpha = 256$) with batch size 12. All models are trained on the train set using the Adam optimizer [19] with a learning rate of 2e-4 for 1 epoch. We use 4 A100 40G GPUs for training.

DP Implementation. We guarantee (ϵ, δ)-DP via the Gaussian Mechanism Privacy loss is tracked using Rényi Differential Privacy(RDP) [33]. We set δ close to the inverse dataset size $(1/N)$ and evaluate across strict to mild privacy budgets: $\epsilon \in \{1, 3, 8\}$. Per-sample gradients are clipped at a maximum L_2 norm of $C = 1.0$.

Baselines. Our Dual-Priv Pruning method is compared against: DP-SGD [1]: The standard baseline for DP fine-tuning, applying Gaussian noise to the averaged clipped gradients of all trainable parameters. DPZO [42]: A representative zeroth-order DP optimization method, included to assess alternatives that avoid direct gradient computation. Detailed for baselines are in Appendix G.

Dual-Priv Pruning Configuration. Mechanism 1 (Section 4.1) retains $K = 191$ attention-selected visual tokens plus [CLS] and 30 fused token (40% of total). Mechanism 2 (Section 4.2) employs gradient-update pruning by selecting parameter blocks for update if their noisy gradient norms are among the **top 80%** of all block norms (Eq. (10)).

5.2 Performance on Standard Benchmarks

The results presented in Table 1 demonstrate the efficacy of Dual-Priv Pruning. Our method consistently achieves performance that is often superior to DP-SGD, especially under stricter privacy constraints ($\epsilon \in \{1, 3\}$). DPZO consistently underperforms across all settings, yielding significantly lower accuracy (e.g., only 23.30 on ScienceQA at $\epsilon = 1$, and 0.00 on GQA). This poor performance is largely attributed to the significant convergence difficulties encountered when applying zeroth-order optimization directly to complex MLLM training under DP constraints. On ScienceQA, our approach excels. At the budget of $\epsilon = 3$, Dual-Priv Pruning achieves **82.80** and a notable **75.98** IMG, much outperforming DP-SGD (78.80/70.59 respectively). This significant gain, particularly in the visual-dependent IMG metric, underscores our method's strength in preserving vital visual information despite DP noise. This performance likely benefits from the synergistic effect of visual token pruning and fusion and selective gradient-update. Even at the strictest budget of $\epsilon = 1$, our method maintains a clear advantage on ScienceQA (84.20 vs 81.54; 78.43 vs 72.51). And our method achieves the best non-private performance ($\epsilon = \infty$).On TextVQA and GQA, our method generally performs slightly better than DP-SGD across various privacy levels, confirming its broad applicability. For instance, on GQA, we achieve the best DP performance across all tested ϵ values. On TextVQA, performance is highly competitive. While DP-SGD leads slightly at $\epsilon = 3$ (35.64 vs 35.17), our method achieves the highest accuracy at $\epsilon = 1$ and $\epsilon = 8$ and also obtains the best non-private result. These results suggest that Dual-Priv Pruning offers a more robust privacy-utility trade-off than standard DP techniques for MLLMs. Its strengths are particularly notable under tight privacy budgets.

5.3 Performance on Medical Visual Tasks

To further assess applicability in privacy-sensitive domains, we evaluated performance on PathVQA [13] (pathology) and VQA-RAD [21] (radiology). Table 2 presents a detailed com-

Table 2: Comparison on PathVQA and VQA-RAD under different DP budgets (ours on the right).

| | DPZO | | | | DP-SGD | | | | Ours (Dual-Priv) | | | |
| | PathVQA | | | VQA-RAD | PathVQA | | | VQA-RAD | PathVQA | | | VQA-RAD |
ϵ	BLUE	EXT	F1	Acc(%)	BLUE	EXT	F1	Acc(%)	BLUE	EXT	F1	Acc(%)
1	0.6534	0.0301	0.0592	0.0	0.7222	0.3732	0.3675	47.3	**0.7385**	**0.3840**	**0.3792**	**48.6**
3	0.6534	0.0301	0.0592	0.0	0.7257	0.3712	0.3653	48.1	**0.7263**	**0.3738**	**0.3701**	**48.8**
8	0.6534	0.0301	0.0592	0.0	0.7140	0.3683	0.3635	46.8	**0.7195**	**0.3763**	**0.3713**	**49.0**

parison of performance under different privacy budget. Our method, consistently outperformed DP-SGD across all the metrics, particularly under stricter privacy budgets. For $\epsilon = 1$: on PathVQA, our approach achieved scores of 0.74(BLUE), 0.38 (EXT), and 0.38 (F1), compared to DP-SGD's 0.72, 0.37, and 0.37, respectively. On VQA-RAD, our method achieved an accuracy of 48.60%, surpassing DP-SGD (47.30%). The DPZO baseline performed poorly on both medical datasets. These consistent gains underscore the potential of Dual-Priv Pruning for tuning MLLMs on sensitive medical data while effectively balancing privacy and utility.

5.4 Performance on High Resolution Visual Tasks

To evaluate performance on tasks demanding fine-grained perception and complex reasoning crucial for real-world applicability, we utilize the MME-RealWorld benchmark [53]. Applying differential privacy in such scenarios is particularly challenging, as the noise required for privacy can significantly impair the model's ability to discern visual details and perform nuanced

Table 3: Accuracy (%) on the MME-RealWorld Benchmark (Lite version evaluation, BS=12).

Method	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 8$	$\epsilon = \infty$
DPZO	0.89	19.80	6.33	22.67
DP-SGD	35.44	44.03	42.17	44.50
Ours (Dual-Priv)	**43.98**	**45.34**	**44.40**	42.16

reasoning. Models are DP-finetuned on the main MME-RealWorld training dataset and subsequently evaluated on the held-out MME-RealWorld **lite version**. The results, presented in Table 3, demonstrate a substantial advantage for Dual-Priv Pruning over both DP-SGD and DPZO across all tested privacy budgets ($\epsilon \in \{1, 3, 8\}$). Notably, under the strict $\epsilon = 1$ setting, our method achieves **43.98** accuracy, significantly surpassing DP-SGD (**35.44**). The significant performance gain on this challenging benchmark underscores the effectiveness of Dual-Priv Pruning. Our method, appears better equipped to preserve the crucial reasoning capabilities, even under stringent privacy constraints. This suggests Dual-Priv Pruning is a promising approach for deploying privacy-preserving MLLMs in real-world applications demanding high visual fidelity and complex reasoning.

5.5 Computational Efficiency Analysis

Figure 2 illustrates the average GPU memory usage during fine-tuning for our method compared to the baselines. Across the evaluated datasets, scienceqa on 4 A100s, Dual-Priv Pruning achieves an average **reduction in average GPU memory usage of approximately 14.34%**. Although DPZO slightly reduces 1.74% GPU memory compared with our approach. (It costs 16.7% more time per training step and causes a 56.3% performance loss). But during tested in H20, our method achieve the lowest consumption of GPU memory. This highlights Dual-Priv Pruning's strength in achieving a favorable balance between model performance and robust computational efficiency, thereby making DP fine-tuning for MLLMs more practical. Dual-Priv enables the DP fine-tuning of MLLMs with more constrained resources.

5.6 Ablation Study

Ablation results on ScienceQA ($\epsilon = 1$) are presented in Table 4. The "w/o Fusion Noise" setting, which omits the input-level noise, shows performance decrease (83.50/76.47 ACC/IMG) compared to the Full Method (84.20/78.43 ACC/IMG). This suggests that our strategy of preconditioning the input with noise consistent with the DP optimization offers a beneficial, albeit auxiliary, contribution to performance,

Table 4: Ablation on ScienceQA.

Configuration	ACC	IMG
Full Method	**84.20**	**78.43**
w/o Fusion Noise	83.50	76.47
Mechanism 2 Only	83.00	76.96
Mechanism 1 + Uniform DP	82.80	74.51

without adding a tunable hyperparameter for this noise. Omitting Mechanism 1's token pruning entirely (Mechanism 2 Only) lowers accuracy to 83.00/76.96 and eliminates the computational efficiency benefits. Furthermore, replacing Mechanism 2's selective update with uniform DP-SGD noise significantly degrades performance to 82.80/74.51, confirming the effectiveness of our adaptive
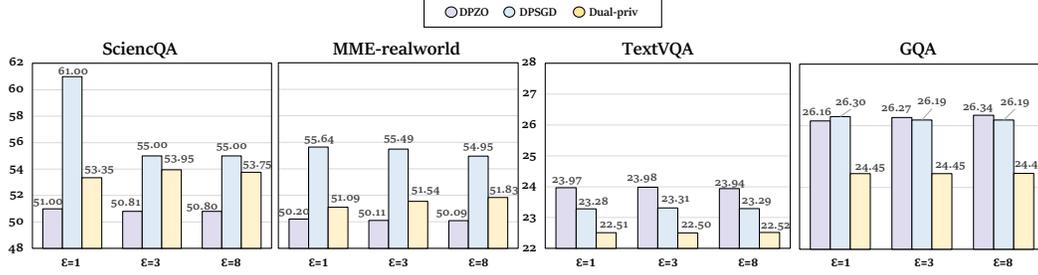
Figure 2: Average GPU memory consumption (in GB) during fine-tuning for DPZO, DP-SGD, and our Dual-Priv Pruning across four datasets: ScienceQA, MME-RealWorld (evaluated on 4xA100 40G GPUs), and TextVQA, GQA (evaluated on a single H20 96GB GPU). Experiments were conducted with varying privacy budgets ($\epsilon \in \{1, 3, 8\}$). Lower bars indicate greater memory efficiency.

update strategy. These findings demonstrate that both Mechanism 1 and Mechanism 2 are crucial components to the overall performance of dual-private pruning.

5.7 Impacts of Pruning Ratios

We examine the impact of different pruning ratios within the Dual-Priv Pruning framework on the ScienceQA dataset ($\epsilon = 1$). Figure 3 (a) displays the relationship between the **gradient-update pruning ratio** and overall accuracy (ACC). The ACC peaks at **84.20** when the top 80% of blocks are updated. Updating all blocks yields a lower ACC of 82.80. Reducing the update ratio further (60%, 50%, 10%) leads to ACC values of 81.60, 82.00, and 81.10. Figure 3 (b) illustrates how the **visual token retention**
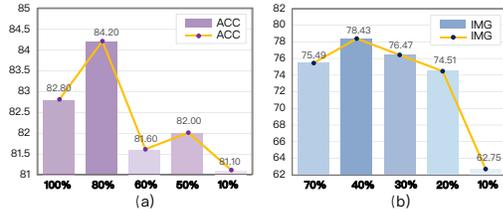

Figure 3: Pruning ratios impacts on ScienceQA ($\epsilon = 1$). (a) percentage of top K% gradient blocks updated (Mechanism 2). (b) percentage of visual tokens retained (Mechanism 1).

ratio affects image-based accuracy (IMG). The IMG shows a peak of **78.43%** when retaining 40% of the visual tokens. Retaining more tokens (e.g., 70%) results in a lower IMG of 75.49, while retaining fewer tokens progressively reduces performance (76.47 at 30%, 74.51 at 20%), with a sharp drop to 62.75 when only 10% of tokens are kept. These experiments highlight that both pruning mechanisms involve a delicate trade-off. Optimal performance requires retaining sufficient signal (visual features or gradient updates) while pruning elements that might be redundant or overly affected by DP noise.

5.8 Performance Under Membership Inference Attacks

To further test the privacy protection capability of our approach, we validate the performance through membership inference attack[39]. The latest MIA design for MLLM [26] was adopted as the evaluation pipeline. Models were DP-finetuned on privacy sensitive medical image caption dataset **ROCOV2**[38] with the batchsize of 12 following the standard setup (Section 5.1). Extensive experiments demonstrate that our work outperform both DPZO and DPSGD across metrics include AUC and ACC, especially in protecting visual information as it benefit from adding heuristic fusion noise in Mechanism 1. As it is shown in Figure 4, the AUC obtained by attacking our model is the lowest under almost every order of Rényi entropy and percentage of top entropies selected, which means that the possibility of distinguishing the member from database is the lowest for Dual-Priv Pruning under the same attack
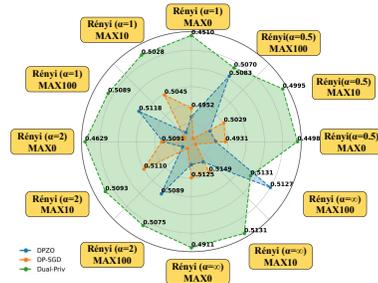

Figure 4: Radar chart of AUC under varying Rényi entropy orders and top entropy percentages. Metrics use strict privacy budget ($\epsilon = 1$). Distribution places smaller values near edges.

pipeline compared with other methods. Further data suggests that our method has a strong ability to protect MLLM from assigning a higher "membership score" to a randomly chosen member than to a randomly chosen non-member, which makes Membership inference attack nearly approaches random guessing. Additional details are in Appendix I.

6 Conclusion

In this work, we introduced **Dual-Priv Pruning**, the first framework for efficient differential private fine-tuning of Multimodal Large Language Models. Our approach combines *visual token pruning* with an input noise strategy aligned with DP noise intensity, and a *gradient-update pruning* mechanism. Extensive experiments demonstrate that Dual-Priv Pruning achieves a compelling privacy-utility trade-off, significantly reducing computational overhead while maintaining competitive performance, especially under stringent privacy budgets. This work represents a crucial first step towards practical privacy-preserving MLLM deployment.

References

[1] 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*, pages 308–318, 2016.

[2] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736, 2022.

[4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 1(2):3, 2023.

[5] Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

[6] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXI*, volume 15139 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2024. doi: 10.1007/978-3-031-73004-7_2. URL https://doi.org/10.1007/978-3-031-73004-7_2.

[7] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. URL https://arxiv.org/abs/2305.06500.

[8] Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. Security and privacy challenges of large language models: A survey. *ACM Computing Surveys*, 57(6):1–39, 2025.

[9] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

[10] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[11] Anmol Goel, Yaxi Hu, Iryna Gurevych, and Amartya Sanyal. Differentially private steering for large language model alignment. *arXiv preprint arXiv:2501.18532*, 2025.

[12] Joakim Bruslund Haurum, Sergio Escalera, Graham W Taylor, and Thomas B Moeslund. Which tokens to use? investigating token reduction in vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 773–783, 2023.

[13] Xuehai He, Yichen Zhang, Luntian Mou, Eric Xing, and Pengtao Xie. Pathvqa: 30000+ questions for medical visual question answering. *arXiv preprint arXiv:2003.10286*, 2020.

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1 (2):3, 2022.

[15] Alyssa Huang, Peihan Liu, Ryumei Nakada, Linjun Zhang, and Wanrong Zhang. Safeguarding data in multimodal ai: A differentially private approach to clip training. *arXiv preprint arXiv:2306.08173*, 2023.

[16] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[17] Georgios Kaissis, Alexander Ziller, Jonathan Passerat-Palmbach, Théo Ryffel, Dmitrii Usynin, Andrew Trask, Ionésio Lima Jr, Jason Mancuso, Friederike Jungmann, Marc-Matthias Steinborn, et al. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nature Machine Intelligence*, 3(6):473–484, 2021.

[18] Gavin Kerrigan, Dylan Slack, and Jens Tuyls. Differentially private language models benefit from public pre-training. *arXiv preprint arXiv:2009.05886*, 2020.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *European conference on computer vision*, pages 620–640. Springer, 2022.

[21] Jason J Lau, Soumya Gayen, Asma Ben Abacha, and Dina Demner-Fushman. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 5(1): 1–10, 2018.

[22] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36: 28541–28564, 2023.

[23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[24] Xianzhi Li, Ran Zmigrod, Zhiqiang Ma, Xiaomo Liu, and Xiaodan Zhu. Fine-tuning language models with differential privacy through adaptive noise allocation. *arXiv preprint arXiv:2410.02912*, 2024.

[25] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

[26] Zhan Li, Yongtao Wu, Yihang Chen, Francesco Tonin, Elias Abad Rocamora, and Volkan Cevher. Membership inference attacks against large vision-language models. *Advances in Neural Information Processing Systems*, 37:98645–98674, 2024.

[27] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[28] Zhihao Liu, Jian Lou, Wenjie Bao, Yuke Hu, Bo Li, Zhan Qin, and Kui Ren. Differentially private zeroth-order methods for scalable large language model finetuning. *arXiv preprint arXiv:2402.07818*, 2024.

[29] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35: 2507–2521, 2022.

[30] Zelun Luo, Yuliang Zou, Yijin Yang, Zane Durante, De-An Huang, Zhiding Yu, Chaowei Xiao, Li Fei-Fei, and Animashree Anandkumar. Differentially private video activity recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6657–6667, 2024. URL https://arxiv.org/abs/2306.01054.

[31] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[32] Bertalan Meskó. The impact of multimodal large language models on health care's future. *Journal of medical Internet research*, 25:e52865, 2023.

[33] Ilya Mironov. Rényi differential privacy. In *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*, pages 263–275. IEEE Computer Society, 2017. doi: 10.1109/CSF.2017.11. URL https://doi.org/10.1109/CSF.2017.11.

[34] Hyeonwoo Noh, Tackgeun You, Jonghwan Mun, and Bohyung Han. Regularizing deep neural networks by noise: Its interpretation and optimization. *Advances in neural information processing systems*, 30, 2017.

[35] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[37] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

[38] Johannes Rückert, Louise Bloch, Raphael Brüngel, Ahmad Idrissi-Yaghir, Henning Schäfer, Cynthia S Schmidt, Sven Koitka, Obioma Pelka, Asma Ben Abacha, Alba G. Seco de Herrera, et al. Rocov2: Radiology objects in context version 2, an updated multimodal image dataset. *Scientific Data*, 11(1):688, 2024.

[39] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[40] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.

[41] Xinyu Tang, Ashwinee Panda, Vikash Sehwag, and Prateek Mittal. Differentially private image classification by learning priors from random processes. *Advances in Neural Information Processing Systems*, 36:35855–35877, 2023. URL https://arxiv.org/abs/2301.12707.

[42] Xinyu Tang, Ashwinee Panda, Milad Nasr, Saeed Mahloujifar, and Prateek Mittal. Private fine-tuning of large language models with zeroth-order optimization. *arXiv preprint arXiv:2401.04343*, 2024.

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

[44] Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokul Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, et al. Fastvlm: Efficient vision encoding for vision language models. *arXiv preprint arXiv:2412.13303*, 2024.

[45] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[46] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, et al. Cogvlm: Visual expert for pretrained language models. *Advances in Neural Information Processing Systems*, 37:121475–121499, 2024.

[47] Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2092–2101, 2023.

[48] Jungang Yang, Liyao Xiang, Size Peng, Yifan Bao, Hui Xu, Pengzhi Chu, Xinbing Wang, and Chenghu Zhou. Improving differentially-private deep learning with gradients index pruning, 2023. In *URL https://openreview. net/forum*.

[49] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. *arXiv preprint arXiv:2412.04467*, 2024.

[50] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*, 2021.

[51] Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Large scale private learning via low-rank reparametrization. In *International Conference on Machine Learning*, pages 12208–12218. PMLR, 2021.

[52] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. URL `https://arxiv.org/abs/2205.01068`.

[53] Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257*, 2024.

[54] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Key Differential Privacy Facts

The following facts elucidate key DP properties essential for MLLM fine-tuning:

Fact A.1 (Sensitivity and the Gaussian Mechanism [10]). *To protect the output of a function f using noise, we first need to leverage its L_2 sensitivity Δf. This measures the maximum possible change $\|f(\mathcal{D}) - f(\mathcal{D}')\|_2$ when the input dataset changes by one record. If Δf is known (or bounded), the Gaussian Mechanism adds noise $\mathcal{N}(0, \sigma_{GM}^2 I)$ where the standard deviation σ_{GM} is related to Δf and depends on the desired single-step privacy (ϵ, δ), calculated as:*

$$\sigma_{GM} \geq \frac{\Delta f \sqrt{2\ln(1.25/\delta)}}{\epsilon}. \tag{12}$$

Fact A.2 (Privacy Accounting via RDP [33]). *Privacy accounting methods are essential for tracking this privacy loss. Rényi Differential Privacy (RDP) [33] is widely used for such accounting [1]. The RDP accountant's practical role is to, given a target overall privacy budget (ϵ, δ), total training steps T, and sampling rate q, compute the required per-step noise multiplier σ (3.1.1) and suggest a clipping norm C to meet the (ϵ, δ)-DP guarantee. Mathematical details are in Appendix C.*

B Sensitivity Analysis for DP-SGD under Add-or-Remove Adjacency

In DP-SGD (Definition 3.1.1), we apply the Gaussian Mechanism to the average of per-sample clipped gradients. The choice of adjacency definition for datasets $\mathcal{D}$ and $\mathcal{D}'$ (i.e., how they differ by "one record") impacts the sensitivity calculation. As stated in Definition 3.1, our work considers add-or-remove adjacency, where neighboring datasets differ by the addition or removal of a single image-text pair.

Consider the function $f(\mathcal{D}, \theta) = \sum_{x_i \in \mathcal{D}} \hat{g}_i(\theta)$, which is the sum of clipped gradients $\hat{g}_i$ for all samples x_i in a dataset $\mathcal{D}$. Each per-sample clipped gradient satisfies $\|\hat{g}_i\|_2 \leq C$. If we consider two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ where $\mathcal{D}' = \mathcal{D} \cup \{x^*\}$ (i.e., x^* is added), then:

$$\|f(\mathcal{D}, \theta) - f(\mathcal{D}', \theta)\|_2 = \| \sum_{x_i \in \mathcal{D}} \hat{g}_i(\theta) - (\sum_{x_i \in \mathcal{D}} \hat{g}_i(\theta) + \hat{g}_{x^*}(\theta))\|_2 = \| -\hat{g}_{x^*}(\theta)\|_2 = \|\hat{g}_{x^*}(\theta)\|_2 \leq C.$$

Similarly, if $\mathcal{D}' = \mathcal{D} \setminus \{x^*\}$ (i.e., x^* is removed), the difference is also bounded by C. Thus, the L_2 sensitivity of the sum of clipped gradients is $\Delta_2 f = C$.

In DP-SGD, we typically compute gradients over a minibatch ξ_t of size m sampled from the full dataset $\mathcal{D}_{\text{train}}$ (of size N) with sampling probability $q = m/N$. The noisy update is applied to the *average* of these clipped gradients: $\bar{g} = \frac{1}{m} \sum_{i \in \xi_t} \hat{g}_i$. For the per-iteration application of DP-SGD with minibatch sampling, the effective L_2 sensitivity of the quantity to which noise is added (the average gradient) is commonly taken as $\frac{C}{m}$ under the add-or-remove model when considering the privacy implications for each individual sample's contribution to this average [1, 10]. More precisely, the clipping ensures that the maximum influence of any single user's data on the sum of gradients in a batch is C. When this sum is averaged over m samples, the change due to one user's data (if that user were removed from the entire dataset) can be bounded appropriately, leading to the noise calibration based on C.

The noise added in DP-SGD (Eq. (2)) is $\mathcal{N}(0, \sigma^2 C^2 I_d / m^2)$. This formulation inherently uses C as the sensitivity for the sum of gradients in the batch if we consider each sample's gradient to be a distinct quantity to be protected, and then this noise is scaled by $1/m$ due to averaging (equivalently, the sensitivity of the average is C/m). The critical point is that clipping each per-sample gradient to C bounds its maximum possible contribution. The privacy analysis with subsampling (accounted for by the RDP accountant) then correctly tracks the privacy loss given this per-sample bound C.

Therefore, the clipping norm C directly bounds the L_2 norm of each individual's contribution before aggregation and noise addition. The Gaussian Mechanism (Fact A.1) is then applied using this understanding, where the effective sensitivity for the noisy average gradient computation in DP-SGD is appropriately scaled by C.

C Rényi Differential Privacy (RDP) Accounting

RDP [33] provides a way to track privacy loss using Rényi divergence of order α, denoted $D_\alpha(P||Q)$. An algorithm $\mathcal{A}$ is (α, ρ)-RDP if for all neighboring datasets $\mathcal{D}, \mathcal{D}'$, $D_\alpha(\mathcal{A}(\mathcal{D})||\mathcal{A}(\mathcal{D}')) \leq \rho$. Key properties include:

- **Composition:** If $\mathcal{A}_1$ is (α, ρ_1)-RDP and $\mathcal{A}_2$ is (α, ρ_2)-RDP, their composition $\mathcal{A}_2 \circ \mathcal{A}_1$ is $(\alpha, \rho_1 + \rho_2)$-RDP. This simplifies tracking loss over multiple steps.
- **Gaussian Mechanism RDP:** Adding $\mathcal{N}(0, \sigma_{GM}^2 I)$ noise to a function with L_2 sensitivity Δf satisfies $(\alpha, \frac{\alpha(\Delta f)^2}{2\sigma_{GM}^2})$-RDP for any $\alpha > 1$.
- **Subsampling Amplification:** Sampling a minibatch with rate q before applying a DP mechanism amplifies privacy. RDP provides tight bounds for this, especially for Poisson sampling [1] and uniform sampling without replacement.
- **Conversion to (ϵ, δ)-DP:** If an algorithm is (α, ρ)-RDP for all α in some range, it satisfies (ϵ, δ)-DP where $\epsilon = \rho + \frac{\log(1/\delta)}{\alpha - 1}$. We typically optimize over α to find the tightest ϵ for a given δ.

The privacy accountant takes T, q, the per-step RDP parameters (derived from the Gaussian mechanism using C and σ), applies composition and subsampling rules to get the total RDP parameters (α, ρ_{total}), and converts this to the final (ϵ, δ). It can also work backwards: given target (ϵ, δ), T, q, find the required σ.

D Post-Processing Property of Differential Privacy

The post-processing property is a fundamental and powerful feature of differential privacy [10]. It states that applying any arbitrary data-independent computation to the output of a differentially private algorithm does not compromise its privacy guarantee.

Formal Statement: Let $\mathcal{A} : \mathcal{D}^n \to \mathcal{R}$ be an (ϵ, δ)-differentially private algorithm, where $\mathcal{D}^n$ is the space of possible datasets and $\mathcal{R}$ is the output range. Let $f : \mathcal{R} \to \mathcal{R}'$ be any arbitrary randomized or deterministic function whose computation does not depend on the original private dataset $\mathcal{D}$ (it only takes the output of $\mathcal{A}$ as input). Then the composite algorithm $f \circ \mathcal{A}$ (which first runs $\mathcal{A}$ on the input dataset and then applies f to the result) is also (ϵ, δ)-differentially private.

Intuition: The privacy guarantee provided by $\mathcal{A}$ ensures that its output $Y = \mathcal{A}(D)$ is already "privacy-safe" – observing Y reveals limited information about any individual in D. The function f only gets access to this already protected output Y. Since f has no additional access to the original sensitive data D, it cannot "undo" the privacy protection or learn anything more about individuals in D than what was already bounded by the (ϵ, δ)-DP guarantee of $\mathcal{A}$.

Proof Sketch: We want to show that for any neighboring datasets D, D' and any set of outcomes $S' \subseteq \mathcal{R}'$:
$$\Pr[(f \circ \mathcal{A})(D) \in S'] \leq e^\epsilon \cdot \Pr[(f \circ \mathcal{A})(D') \in S'] + \delta$$
Let $Y = \mathcal{A}(D)$ and $Y' = \mathcal{A}(D')$. The event $(f \circ \mathcal{A})(D) \in S'$ means $f(Y) \in S'$. Let $S_f = \{y \in \mathcal{R} \mid f(y) \in S'\}$ be the set of outputs from $\mathcal{A}$ that f maps into S'. Then, $\Pr[(f \circ \mathcal{A})(D) \in S'] = \Pr[Y \in S_f] = \Pr[\mathcal{A}(D) \in S_f]$. Similarly, $\Pr[(f \circ \mathcal{A})(D') \in S'] = \Pr[\mathcal{A}(D') \in S_f]$. Since $\mathcal{A}$ is (ϵ, δ)-DP and S_f is a valid subset of its output range $\mathcal{R}$, we know from Definition 1:
$$\Pr[\mathcal{A}(D) \in S_f] \leq e^\epsilon \cdot \Pr[\mathcal{A}(D') \in S_f] + \delta$$

Substituting back, we get:
$$\Pr[(f \circ \mathcal{A})(D) \in S'] \leq e^\epsilon \cdot \Pr[(f \circ \mathcal{A})(D') \in S'] + \delta$$

This holds for any S', proving that $f \circ \mathcal{A}$ is (ϵ, δ)-DP.

Relevance to MLLM Fine-tuning: In our context, the DP-SGD algorithm $\mathcal{A}$ takes the private dataset $\mathcal{D}_{\text{fine}}$ and produces the model parameters θ_{fine}. The act of generating a prediction for a new input x, i.e., computing $M_{\theta_{\text{fine}}}(x)$, can be viewed as a post-processing function f applied to θ_{fine}. Therefore, the generated predictions inherit the same (ϵ, δ)-DP guarantee with respect to the fine-tuning dataset $\mathcal{D}_{\text{fine}}$.

E   Motivation for Mechanism 1: Visual Token Pruning and Fusion

This appendix details the motivation behind the visual input preprocessing performed in Mechanism 1 of our Dual-Priv Pruning method (Section 4). This stage operates *before* the formal Differentially Private (DP) fine-tuning in Mechanism 2 (§4.2) and is designed to address key challenges in applying DP to Multimodal Large Language Models (MLLMs). Specifically, it aims to reduce computation cost and potentially improve the utility outcome under DP constraints by modifying the visual token sequence.

E.1   Addressing Computation Cost and Visual Redundancy

Fine-tuning MLLMs using DP-SGD can be computationally demanding, due to the high number of visual tokens (n) generated by the vision encoder. It has been observed that considerable redundancy exists within the visual tokens generated by Vision Transformers (ViTs), and not all tokens are equally important for downstream task performance [12, 20]. Building on the insight that attention scores often correlate with token importance [12], Mechanism 1 identifies and retains only the top-K tokens receiving the highest aggregated attention from the [CLS] token. This selective pruning significantly shortens the sequence length processed in Mechanism 2, thereby directly reducing computation overhead. This strategy aligns with research exploring attention-based token pruning in ViTs [20, 37].

E.2   Preserving Context via Token Fusion

While pruning reduces costs, simply discarding less attended tokens might remove valuable contextual information. To mitigate this, Mechanism 1 adopts a fusion strategy inspired by techniques that aim to compress information from pruned parts of a network or input [47]. We merge the non-dominant tokens ($\mathcal{V}_{nd}$) into selected context tokens (c). This allows us to maintain a drastically reduced sequence length for efficiency while still incorporating a summarized representation of the less critical visual information, aiming for a better balance between computational savings and information preservation.

E.3   Heuristic Noise Injection: Motivations and Potential Benefits

The final step of Mechanism 1 introduces heuristic Gaussian noise to the fused context tokens (c) (Eq. (7)). This deliberate noise injection is multifaceted, aiming to potentially enhance the subsequent DP fine-tuning process:

- **Regularization against DP Noise:** Adding noise is a known regularization technique [5, 34]. Injecting noise specifically into the summarized, less critical token representation might act as **targeted input regularization**. This could potentially improve the model's robustness against the gradient perturbations inherent in the DP mechanism.

- **Encouraging Focus on Critical Tokens:** By introducing stochasticity primarily to the fused context token, the model might be implicitly encouraged during fine-tuning to rely more heavily on the stable, un-noised dominant tokens ($\mathcal{V}_d, v_{cls}$). This could help **preserve utility related to salient image features**.

- **Connection to Learning with Noise Priors:** Although mechanically different, this strategy shares a conceptual link with methods improving DP training by incorporating knowledge from noisy processes [41].Our direct noise injection might serve a similar purpose by **preconditioning the model with input stochasticity**, potentially enhancing its resilience to the noise required for the DP guarantee in Mechanism 2.

- **Conceptual Input-Level Obfuscation:** While not contributing to the formal DP guarantee, manipulating the representation of less critical tokens with heuristic noise offers a degree of **data obfuscation at the input level**. This might provide some practical hardening against certain inference attacks targeting those specific, less informative image regions.

It is crucial to emphasize that the noise added in Mechanism 24.1 (σ^2_{fuse}) is **heuristic**. It is not calibrated according to DP principles and serves as a hyperparameter tuned for its potential benefits to utility and robustness.

F Motivation for Mechanism 2 Gradient-Update Pruning

The post-noise adaptive update mechanism described in Section 4.2 is motivated by the goal of enhancing model utility under the constraints imposed by DP-SGD noise. Standard DP-SGD applies the noisy gradient $\tilde{g}$ (Eq. (9)) uniformly to all trainable parameters θ_{train}. However, the added noise can significantly perturb or even dominate the true gradient signal, especially for parameter blocks where the original gradient magnitude was small or when operating under strict privacy budgets (requiring large σ). Applying updates based on such noise-dominated gradients might hinder convergence or lead to suboptimal performance.

Our strategy addresses this by analyzing the noisy gradient $\tilde{g}$ *after* the privacy-preserving noise has been added. By partitioning $\tilde{g}$ into logical blocks $\tilde{g}_j$ and examining their L_2 norms $N_j = \|\tilde{g}_j\|_2$, we attempt to identify blocks where the signal likely outweighs the noise. The assumption is that a relatively large norm N_j suggests that the original aggregated gradient component $\bar{\tilde{g}}_j$ was sufficiently strong to persist despite the noise addition, thus indicating a more reliable update direction. Conversely, a small norm N_j might indicate that the true signal was weak or was largely cancelled by the random noise vector.

The gating mechanism (Eq. (10)) leverages this analysis. By generating a mask M that selectively allows updates only for blocks with high noisy-gradient norms (i.e., where $M_j = 1$), we filter out potentially detrimental updates arising from low-signal or noise-dominated gradient components. The final masked update (Eq. (11)) focuses the optimization process on parameter blocks associated with stronger, potentially more informative, noisy gradient signals. This aims to improve the effective signal-to-noise ratio of the updates applied to the model, potentially leading to better convergence, improved utility, and a more favorable privacy-utility trade-off for the given privacy budget (ϵ, δ).

G Baseline Details

This section provides detailed descriptions, algorithms, and hyperparameter configurations for the baseline methods used in our comparative experiments.

G.1 DP-SGD Baseline

We implement the standard Differentially Private Stochastic Gradient Descent (DP-SGD) algorithm [1], formally defined in 3.1.1. This method involves computing per-sample gradients, clipping their L_2 norms, averaging the clipped gradients, and adding calibrated Gaussian noise before updating the model parameters. It serves as the primary benchmark for differentially private optimization in deep learning. The hyperparameter configuration used for DP-SGD is detailed in Table 5.

Table 5: Hyperparameter Configuration for DP-SGD Baseline.

Parameter	Value
Base Model	LLAVA-7B [27]
Fine-tuning Method	LoRA [14]
LoRA Rank (r)	128
LoRA Alpha (α)	256
Optimizer	Adam [19]
Learning Rate	2e-4
Batch Size	12
Epochs	1
DP Parameters	
Clipping Norm (C)	1.0
Target δ	$\approx 1/N$ (Inverse dataset size)
Target ϵ Values	$\{1, 3, 8, \infty\}$
Noise Multiplier (σ)	Calculated via RDP [33] based on target (ϵ, δ), C, q, and total steps.

G.2 DPZO Baseline

DPZO (Differentially Private Zeroth-Order Optimization) [42] is a gradient-free DP optimization method. It approximates the gradient direction using finite differences based on random perturbations and privatizes only a scalar value representing the estimated directional derivative (loss difference). This avoids the memory overhead associated with storing per-sample gradients, but often requires significantly more iterations for convergence compared to DP-SGD. Algorithm 3 outlines the core mechanism adapted from [42]. The specific configuration used in our experiments is presented in Table 6.

Table 6: Hyperparameter Configuration for DPZO Baseline.

Parameter	Value
Base Model	LLAVA-7B [27]
Fine-tuning Method	LoRA [14]
LoRA Rank (r)	128
LoRA Alpha (α)	256
Learning Rate (η)	2e-4
Perturbation Scale (ϕ)	0.15
Batch Size	12
Epochs	1
DP Parameters	
Clipping Norm (C_{ZO})	1.0
Target δ	$\approx 1/N$
Target ϵ Values	$\{1, 3, 8, \infty\}$
Noise Multiplier (σ_{ZO})	Calculated via RDP accountant based on target (ϵ, δ), C_{ZO}, $q = m/N$, T.

H Detailed Results on Medical Datasets

This section provides the detailed performance results for the experiments on the PathVQA and VQA-RAD datasets, as referenced in Section 5.1. All experiments used a batch size (BS) of 12.

Table 7: Detailed performance on PathVQA (BS=12). Higher is better for BLUE, EXT, F1. Best DP results in **bold**.

ϵ	Ours (Dual-Priv)			DPZO			DP-SGD		
	BLUE	EXT	F1	BLUE	EXT	F1	BLUE	EXT	F1
1	**0.7385**	**0.3840**	**0.3792**	0.6534	0.0301	0.0592	0.7222	0.3732	0.3675
3	**0.7263**	**0.3738**	**0.3701**	0.6534	0.0301	0.0592	0.7257	0.3712	0.3653
8	**0.7195**	**0.3763**	**0.3713**	0.6534	0.0301	0.0592	0.7140	0.3683	0.3635
∞	0.7430	0.3880	0.3841	0.6534	0.0301	0.0592	0.7182	**0.3927**	**0.3879**

Table 8: Detailed accuracy (%) on VQA-RAD (BS=12). Higher is better. Best DP result in **bold**.

ϵ	Ours (Dual-Priv)	DPZO	DP-SGD
1	**48.6**	0.0	47.3
3	**48.8**	0.0	48.1
8	**49.0**	0.0	46.8
∞	47.9	0.0	**48.3**

I Additional Details on Membership Inference Attack

This section provides the additional details for the experiments with membership inference attack, as referenced in Section 5.8. All experiments used a batch size(BS) of 12. We randomly sample 6,000 image-text pairs from the ROCOV2 dataset for evaluation and randomly sampled 3000 image-text pairs as the member dataset for training. To fit the LLaVA-VQA formulation, we randomly use these

prompts:"Please describe the image in detail.", "What is shown in this medical image?", "Describe the contents of this image.", "What does this image depict?", "Provide a detailed description of this image.", "Please analyze this medical image.", "Describe the medical image in detail.", "Describe the condition depicted in the image.", "Please provide a caption for this image."

J   Limitations

Our study demonstrates the effectiveness of Dual-Priv Pruning for DP fine-tuning MLLMs. While our evaluations on a 7B MLLM are thorough, extending the assessment to MLLMs of substantially different scales would provide a broader understanding of the approach's scalability.

K   Broader Impacts

The development of Dual-Priv Pruning contributes to the critical area of privacy-preserving machine learning, particularly for Multimodal Large Language Models (MLLMs). The primary societal benefit lies in its potential to significantly enhance data privacy when fine-tuning MLLMs on sensitive datasets. By integrating Differential Privacy (DP) with improved efficiency and utility, our work can empower the responsible use of MLLMs in domains handling personal information, such as healthcare or finance, thereby protecting individuals from data leakage. This advancement may also lower barriers to adopting privacy-enhancing technologies, encouraging a broader shift towards responsible AI practices and facilitating research on valuable sensitive datasets that might otherwise remain underutilized due to privacy risks. Ultimately, robust privacy measures like those explored can foster greater public trust in AI systems, which is vital for their ethical and successful integration into society.

However, it is important to consider the broader context. While DP offers strong mathematical privacy guarantees, these are contingent upon correct implementation and careful parameter selection, and they address specific threats related to individual data contributions rather than all conceivable privacy concerns. A nuanced understanding is crucial to avoid a false sense of absolute security. The inherent trade-off between privacy protection and model utility, though mitigated by our approach, persists; in certain high-stakes applications, even minor performance degradation due to DP noise could have notable implications if not carefully weighed. Furthermore, the expertise required to effectively implement and tune DP mechanisms remains a consideration for broader accessibility. While our method focuses on the privacy of training data, the underlying MLLM technology itself, regardless of how it's fine-tuned, could still be subject to misuse if its outputs are leveraged for unintended or harmful purposes.

Our research is a step towards more responsible AI development. We believe continued efforts in the community are essential to further refine the balance between privacy and utility, enhance the usability of DP tools, and promote comprehensive education on both the capabilities and limitations of such privacy-enhancing technologies. Addressing fairness and bias within DP-trained models also remains an important ongoing pursuit. This work is presented as foundational research to advance privacy in MLLM fine-tuning, with the anticipation that its net impact will be positive by enabling more secure and trustworthy AI applications.

L   Algorithm for Baselines and Dual-priv Pruning

Algorithm 1 provides the detailed step-by-step procedure for the Stage 2 DP fine-tuning process described in Section 4.2 of the main paper. While Algorithm 2 outlines the standard DP-SGD baseline, and Algorithm 3 details the DPZO baseline.

Algorithm 1 Dual-Priv Pruning: Mechanism 2 (DP Fine-tuning with Gradient-Update Pruning)

Require: Initial trainable parameters θ_{train_0}, dataset $D = \{(\mathcal{V}'_i, \mathcal{T}_i)\}_{i=1}^N$ (with pre-processed visual inputs $\mathcal{V}'_i$), learning rate schedule η_t, gradient clipping norm C, noise multiplier σ (derived from target ϵ, δ), batch size m, total training steps T, number of logical parameter blocks J in θ_{train}, top-K percentage P_K.

1: **for** $t = 1, \ldots, T$ **do**
2:     Sample minibatch $\xi_t = \{(\mathcal{V}'_k, \mathcal{T}_k)\}_{k=1}^m \subset D$ of size m.
3:     Initialize list of per-sample gradients $G_{list} = []$.
4:     **for** each sample $(\mathcal{V}'_k, \mathcal{T}_k) \in \xi_t$ **do**
5:         Compute gradient $g_k \leftarrow \nabla_{\theta_{\text{train}_{t-1}}} \mathcal{L}(\theta_{\text{train}_{t-1}}; (\mathcal{V}'_k, \mathcal{T}_k))$.
6:         Clip gradient: $\hat{g}_k \leftarrow g_k / \max(1, \|g_k\|_2 / C)$.
7:         Append $\hat{g}_k$ to G_{list}.
8:     **end for**
9:     Aggregate clipped gradients: $\bar{\hat{g}} \leftarrow \frac{1}{m} \sum_{\hat{g}_k \in G_{list}} \hat{g}_k$.
10:    Add Gaussian noise: $\tilde{g} \leftarrow \bar{\hat{g}} + \mathcal{N}\left(0, \frac{\sigma^2 C^2}{m^2} I_{d_{\text{train}}}\right)$.
11:    Partition $\tilde{g}$ into J components $\{\tilde{g}_1, \ldots, \tilde{g}_J\}$ corresponding to logical parameter blocks.
12:    Compute L_2 norms for each block: $N_j \leftarrow \|\tilde{g}_j\|_2$ for $j = 1, \ldots, J$.
13:    Initialize mask M as a zero tensor with the same block structure as θ_{train}.
14:    Let $K_{\text{count}} \leftarrow \lceil (P_K/100) \cdot J \rceil$.
15:    Let $\mathcal{S}_{\text{top\_indices}}$ be the set of indices of the K_{count} blocks with the largest norms N_j.
16:    **for** each block index $j \in \mathcal{S}_{\text{top\_indices}}$ **do**
17:        Set corresponding part of mask $M_j \leftarrow \mathbf{1}$ (vector/matrix of ones for block j).
18:    **end for**
19:    Update parameters: $\theta_{\text{train}_t} \leftarrow \theta_{\text{train}_{t-1}} - \eta_t \cdot (M \odot \tilde{g})$.
20: **end for**
21: **return** θ_{train_T}.

Algorithm 2 Differentially Private Stochastic Gradient Descent (DP-SGD, adapted from [1])

Require: Initial model parameters θ_0, dataset $D = \{(\mathcal{I}_i, \mathcal{T}_i)\}_{i=1}^N$ (or generic x_i), learning rate η_t, clipping norm C, noise multiplier σ, batch size m, total steps T.

1: **for** $t = 1, \ldots, T$ **do**
2:     Sample minibatch $\xi_t = \{x_k\}_{k=1}^m \subset D$ of size m.
3:     Initialize list of per-sample gradients $G_{list} = []$.
4:     **for** each sample $x_k \in \xi_t$ **do**
5:         Compute gradient $g_k \leftarrow \nabla_{\theta_{t-1}} \mathcal{L}(\theta_{t-1}; x_k)$.
6:         Clip gradient: $\hat{g}_k \leftarrow g_k / \max(1, \|g_k\|_2 / C)$.
7:         Append $\hat{g}_k$ to G_{list}.
8:     **end for**
9:     Aggregate clipped gradients: $\bar{\hat{g}} \leftarrow \frac{1}{m} \sum_{\hat{g}_k \in G_{list}} \hat{g}_k$.
10:    Add Gaussian noise: $\tilde{g} \leftarrow \bar{\hat{g}} + \mathcal{N}\left(0, \frac{\sigma^2 C^2}{m^2} I_d\right)$.
11:    Update parameters: $\theta_t \leftarrow \theta_{t-1} - \eta_t \cdot \tilde{g}$.
12: **end for**
13: **return** θ_T.

Algorithm 3 DPZO Core Mechanism (Simplified, adapted from [42])

Require: Model parameters θ, dataset D, learning rate η, perturbation scale ϕ, clipping threshold C_{ZO}, noise scale σ_{ZO}, batch size m, total steps T.

1: **for** $t = 1, \ldots, T$ **do**
2:      Sample batch $B \subset D$.
3:      Sample random direction $z_t \sim \mathcal{N}(0, I_d)$.
4:      Set $\theta^+ \leftarrow \theta_{t-1} + \phi z_t$, $\theta^- \leftarrow \theta_{t-1} - \phi z_t$.
5:      Initialize loss differences list $L_{\text{diff}} = []$.
6:      **for** each sample $(\mathcal{I}_i, \mathcal{T}_i) \in B$ **do**
7:          Compute $l_i = \mathcal{L}(\theta^+; (\mathcal{I}_i, \mathcal{T}_i)) - \mathcal{L}(\theta^-; (\mathcal{I}_i, \mathcal{T}_i))$.
8:          Clip difference: $\hat{l}_i \leftarrow \max(-C_{ZO}, \min(l_i, C_{ZO}))$.
9:          Append $\hat{l}_i$ to L_{diff}.
10:     **end for**
11:     Aggregate clipped differences: $\bar{l} \leftarrow \frac{1}{|B|} \sum_{\hat{l}_i \in L_{\text{diff}}} \hat{l}_i$.
12:     Add noise to privatize the average difference: $s \leftarrow \bar{l} + \mathcal{N}(0, \sigma_{ZO}^2 C_{ZO}^2 / |B|^2)$.
13:     Update parameters: $\theta_t \leftarrow \theta_{t-1} - \eta \cdot s \cdot z_t / (2\phi)$.
14: **end for**
15: **return** θ_T.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See Appendix J

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: See Section 3, Appendix A, Appendix D and Appendix C

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We claim the details of methods and the experiments settings in our paper.

Guidelines:

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

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

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

Answer: [Yes]

Justification: We include the code in our supplemental material.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification:: The experimental settings are detailed in Section 5.1

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: We believe the pattern is clear.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

Answer: [Yes]

Justification:See Section 5

Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: It does.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: See Appendix K

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

26

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We cite all works properly.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: We offer documentation alongside our code. The anonymized repository including code and documentation can be found at: `https://anonymous.4open.science/r/Dual-priv-pruning-AE7E`

Guidelines:

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

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

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

Answer: [NA]

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

Guidelines:

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

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

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

Answer: [Yes]

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

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: The core methodology of this research is centered on the differential private fine-tuning of Multimodal Large Language Models (MLLMs).

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.