COVERT: Privacy-Preserving Covariant Obfuscation for VLMaaS via Exact Reparameterization and Tailored Tuning

Wenqiang Ruan[1]*, Zirui Huang[1,2]*, Yu Lin[1], Qizhi Zhang[1], Yunlong Mao[2]✉, Quanwei Cai[1], Jue Hong[1], Sheng Zhong[2], and Ye Wu[1]✉

[1] ByteDance
{ruanwenqiang, linyu.09, zhangqizhi.zqz, caiquanwei, tanzhuo.107, wuye.2020}@bytedance.com
[2] State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210023, China
huangzirui@smail.nju.edu.cn, {maoyl, zhongsheng}@nju.edu.cn

Abstract. With the emergence of Vision-Language Models (VLMs), cloud-based VLM inference services have raised critical privacy concerns regarding the potential leakage of sensitive visual prompts. However, existing defenses compromise either efficiency or model utility to achieve privacy. Recently, the covariant obfuscation paradigm has shown great potential for private large language model inference, but it is incompatible with cross-modal architectures and tasks. In this paper, we propose **COVERT**, the *first* practical privacy-preserving VLM inference framework, which explicitly adapts covariant obfuscation for multimodal architectures. Specifically, it integrates exact architectural reparameterization to robustly obfuscate visual features against naive inversion attacks, alongside utility-aware parameter tuning tailored to neutralize hidden-state inversion attacks, thereby achieving a favorable privacy-utility-efficiency trade-off. Extensive evaluations show that COVERT effectively shields sensitive visual attributes against hidden-state inversion attacks, while keeping the average accuracy degradation below 3% across major benchmarks and sustaining high inference throughput with a negligible overhead of less than 6%.

Keywords: Vision-Language Models · Privacy-Preserving Inference · Covariant Obfuscation, Hidden-State Inversion · Model-as-a-Service

1 Introduction

As the demand for advanced cross-modal reasoning escalates in high-stakes domains like medicine and finance [15], the deployment of Vision-Language Models (VLMs) has widely shifted toward the Vision-Language Model-as-a-Service (VLMaaS) paradigm, where resource-constrained clients outsource inference by

* Equal contribution.
✉ Correponding authors.

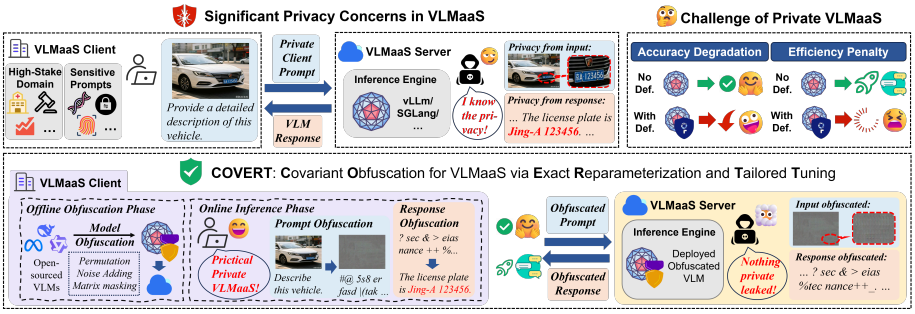

Fig. 1: VLMaaS with related privacy challenges and the illustration of COVERT.

uploading sensitive multimodal prompts to cloud servers. This inevitably raises severe privacy concerns regarding sensitive visual semantics, such as biometric traits and confidential identities [26]. Consequently, safeguarding client prompt privacy in VLMaaS has emerged as a critical imperative.

However, as shown in Fig. 1, practical privacy-preserving VLMaaS remains an open challenge, as an ideal solution must simultaneously guarantee data privacy, preserve model utility, and enable lightweight deployment on existing architectures [18]. Current methods fundamentally fail to meet these criteria: heavy cryptographic protocols incur prohibitive overheads on high-dimensional visual signals [32], while differential privacy methods based on perturbations destroy cross-modal semantic alignment, severely degrading utility [9, 21]. Furthermore, most existing privacy-preserving inference mechanisms are tailored for text-only models, with limited research dedicated to VLMs [22, 25, 33].

To bridge this gap, we propose **COVERT** (**C**ovariant **O**bfuscation for **V**LMaaS via **E**xact **R**eparameterization and Tailored **T**uning), as shown in Fig. 1. To our best knowledge, COVERT is the *first* practical privacy-preserving VLM inference framework. COVERT leverages *covariant obfuscation* [18] to jointly transform input data and model weights, balancing robust privacy with model utility. Recognizing that modern VLMs consist of a vision encoder and a backbone Large Language Model (LLM) decoder, COVERT strategically adapts its defense across modalities: it introduces a dedicated covariant obfuscation paradigm tailored for the visual pipeline, while seamlessly integrating AloePri [18] to protect the textual components. Specifically, COVERT employs exact architectural reparameterization to seamlessly adapt covariant obfuscation to visual modules of VLMs. This yields protected visual representations that robustly withstand naive feature inversion attacks [5, 17, 30], ensuring the final inference responses remain exclusively decodable by the client. Furthermore, to mitigate privacy leakage induced by the spatial invertibility of shallow ViT attention [7], COVERT incorporates a utility-aware parameter tuning strategy designed to defend hidden-state inversion attacks. By integrating exact reparameterization with tailored tuning, COVERT achieves a favorable balance between utility, privacy, and efficiency. Extensive evaluations demonstrate its practical effectiveness: COVERT successfully conceals sensitive visual attributes from state-of-the-art inversion attacks,

limits average accuracy drops to under 3% across various settings, and incurs a minor inference throughput reduction of less than 6% on the `vLLM` engine [13].

In summary, the main contributions of this paper are:

- We propose **COVERT**, the first practical privacy-preserving framework dedicated to VLMaaS, establishing a modality-adaptive covariant obfuscation paradigm to address cross-modal structural heterogeneity.
- We develop exact architectural reparameterization to adapt covariant obfuscation for visual modules of VLMs, and introduce a utility-aware tuning strategy to defend high-fidelity hidden-state inversion attacks.
- Extensive evaluations across various benchmarks validate that COVERT achieves strong privacy protection against spatial reconstruction with under 3% average accuracy degradation and less than 6% extra inference overhead.

2 Preliminaries and System Model

2.1 System and Threat Model

This paper focuses on a general VLMaaS setting: *a client* with private visual and textual prompts and *a cloud server* hosting the VLM service. Constrained by resources, the client can only perform some offline computations, such as matrix transformations and selective parameter tuning, before starting the service. During online inference, the client handles only lightweight preprocessing tasks (e.g., image flattening and text tokenization). Specifically, in a single communication round, the client transmits pre-processed visual and textual prompts to the server and receives related VLM responses. However, transmitting plaintext inputs causes severe privacy risks: the *honest-but-curious* server may reconstruct the raw visual inputs via feature inversion, or inadvertently disclose sensitive information through the VLM's generated responses. To this end, this paper proposes a practical privacy-preserving framework for VLMaaS that simultaneously guarantees data confidentiality, model utility, and inference efficiency.

2.2 Analysis of VLM Architecture Preliminaries

To identify unique privacy requirements in VLMaaS, we first formalize a unified architecture for state-of-the-art VLMs. This paper focuses on the dominant *LLM-backbone* paradigm, which augments a pre-trained LLM backbone with a specialized vision encoder [16], as illustrated in Fig. 2. Given that this structure supports popular open-source models such as Qwen-VL [2] and LLaVA [19], our analysis maintains broad applicability without loss of generality.

In the LLM-backbone paradigm, VLMs integrate a specialized vision encoder $\mathcal{V}$ with the backbone LLM decoder. Given aligned multi-modality inputs, the backbone LLM autoregressively generates text. Specifically, to bridge the modality gap, the vision encoder $\mathcal{V}$ aligns vision features with the LLM semantic space, which can be formalized as a composition of three distinct operations:

$$\mathbf{E}_v = \mathcal{V}(\mathbf{X}_v) = f_{\text{Proj}} \circ f_{\text{Feat}} \circ f_{\text{Conv}}(\mathbf{X}_v), \tag{1}$$

Specifically, given visual input $\mathbf{X}_v$, the pipeline begins with a *convolutional patcher* f_{Conv} for image patchification, followed by a *vision feature extractor* f_{Feat}, typically instantiated as a pre-trained ViT [8] in mainstream architectures [16]. Finally, an embedding projector f_{Proj}, which maps vision features to the shared semantic space aligned with the LLM. Crucially, f_{Proj} consists of specifically trained MLP layers and may adopt a spatial aggregator to condense visual tokens, thus reducing the sequence length for efficient inference [2, 19].

2.3 Related Work and Privacy Challenges in VLMaaS

The proliferation of VLMaaS raises critical privacy concerns, as client visual inputs inherently contain sensitive semantics [26]. Beyond inferring private attributes from subtle details [5, 24], adversaries can directly invert raw visual prompts from intermediate hidden states [7,17,30]. Since modern VLMs can easily bypass naive pixel-level anonymization [4], robust privacy-preserving mechanisms are urgently needed. However, securing VLMaaS faces three fundamental challenges uniquely introduced by cross-modal visual processing:

Challenge 1: Computational Overhead of Visual Inputs. Unlike discrete text tokens, continuous high-dimensional visual inputs incur massive computational overhead. Consequently, cryptographic approaches like Multi-Party Computation (MPC) [32, 34] or Trusted Execution Environment (TEE) [23, 31] become prohibitively expensive, fundamentally contradicting the lightweight client paradigm necessary for real-time cloud deployment.

Challenge 2: Cross-Modality Sensitivity to Noise. VLM inference relies on delicate semantic alignment between visual and textual representations. This cross-modality architecture makes VLM utility extremely vulnerable to stochastic perturbations: methods applying Differential Privacy (DP) [9, 21, 25] inject uncalibrated noise that irreversibly disrupts spatial and cross-modal mappings, leading to catastrophic utility degradation.

Challenge 3: Architectural Heterogeneity of Multi-Modality. Modern VLMs adopt a composite architecture comprising a vision encoder and an LLM backbone. A comprehensive defense must integrate with text-modality protections and be tailored to the specific visual pipeline. However, many existing privacy mechanisms do not transfer to vision processing [22, 25, 33]. For example, transformation-based methods such as SGT [22] struggle to generalize across complex visual modalities and remain vulnerable to embedding inversion [17].

To overcome these challenges, an effective defense must align with cross-modal structures while aiming to balance privacy, utility, and efficiency.

3 Covariant Obfuscation and Cross-Modal Challenges

This work adopts **covariant obfuscation** [18] as a favorable paradigm for privacy-preserving VLMaaS, which jointly obfuscates client prompts and model parameters for a private, accurate, and efficient VLM inference service. Considering the system model in Section 2, the client and the server both have white-box

access to the original VLM. When initializing the service, the client obfuscates the original model and deploys it to the server. During inference, the semi-honest server runs the obfuscated VLM on the cloud and can observe the obfuscated prompts, responses, and internal features. The server also possesses prior knowledge of the client's prompt distribution, allowing their attempts to recover the client's private information from uploaded prompts and model responses.

3.1 Covariant Obfuscation Preliminaries

Covariant obfuscation [18] protects privacy through joint data-parameter transformations. Formally, given an inference function $f : X \times \Theta \to Y$ over data, parameter, and prediction spaces, a covariant obfuscation framework $\mathcal{C}$ is a quintuple $(\phi_X, \phi_\Theta, \phi_Y, \psi_Y, \tilde{f})$. These denote the obfuscation mappings for data (ϕ_X), parameters (ϕ_Θ), and labels (ϕ_Y), alongside the label de-obfuscator (ψ_Y) and the obfuscated inference function $(\tilde{f})$, respectively.

For functional consistency, $\mathcal{C}$ must satisfy two core axioms. First, the *commutation condition* bounds the computational divergence. Given a distance metric $d : Y \times Y \to \mathbb{R}$ (e.g., Euclid Distance), $\mathcal{C}$ structurally enforces:

$$\mathbb{E}\left[d\big(\tilde{f}(\phi_X(x), \phi_\Theta(\theta)), \phi_Y(f(x,\theta))\big)\right] \leq e_C,$$

which ensures the error induced by obfuscation is strictly bounded by a constant e_C. Second, the *de-obfuscation condition* guarantees exact prediction recovery via $\psi_Y \circ \phi_Y = \mathrm{id}_Y$, where id denotes the identity mapping. Together, these properties explicitly bound the end-to-end inference error. Crucially, compared to naive data-only obfuscation (where $\phi_\Theta = \mathrm{id}_\Theta$ and $\phi_Y = \mathrm{id}_Y$), the Data Processing Inequality [3] theoretically guarantees that covariant obfuscation achieves a strictly superior privacy-utility trade-off. Detailed formalizations and proofs of covariant obfuscation are provided in Lin et al. [18].

3.2 Covariant Obfuscation in VLMs: Primitives and Challenges

AloePri [18] instantiates covariant obfuscation for LLMs through matrix transformations. To illustrate the core mechanism, consider an LLM formulated as $f_{\mathrm{LLM}}(\mathbf{X}) = \mathbf{W}_{\mathrm{head}} \circ g_{\mathrm{trunk}} \circ (\mathbf{W}_{\mathrm{embed}}\mathbf{X})$, where $\mathbf{X}$ denotes the client input and g_{trunk} represents the Transformer backbone. In the offline phase, AloePri first injects calibrated Gaussian noise into the embedding and head modules to derive $\mathbf{W}^{\star}_{\mathrm{embed}}$ and $\mathbf{W}^{\star}_{\mathrm{head}}$. It then samples a vocabulary permutation matrix $\mathbf{\Pi}$ and an invertible key matrix pair $(\hat{\mathbf{P}}, \hat{\mathbf{Q}})$ satisfying $\hat{\mathbf{P}}\hat{\mathbf{Q}} = \mathbf{I}$. These parameters are obfuscated as $\tilde{\mathbf{W}}_{\mathrm{embed}} = \mathbf{\Pi}\mathbf{W}^{\star}_{\mathrm{embed}}\hat{\mathbf{P}}$ and $\tilde{\mathbf{W}}_{\mathrm{head}} = \hat{\mathbf{Q}}\mathbf{W}^{\star}_{\mathrm{head}}\mathbf{\Pi}^{\mathsf{T}}$. To prevent direct inversion, transformation $\hat{\mathbf{P}}$ deliberately expands the original feature dimensions. In the online phase, the client simply applies the permutation $\mathbf{\Pi}$ to generate obfuscated input tokens for the server. This elegant pairing of mutually canceling transformations flawlessly shields textual prompts while preserving exact model accuracy. For obfuscation primitives concerning internal modules in g_{trunk} and detailed matrix generation protocols, we refer readers to AloePri [18].

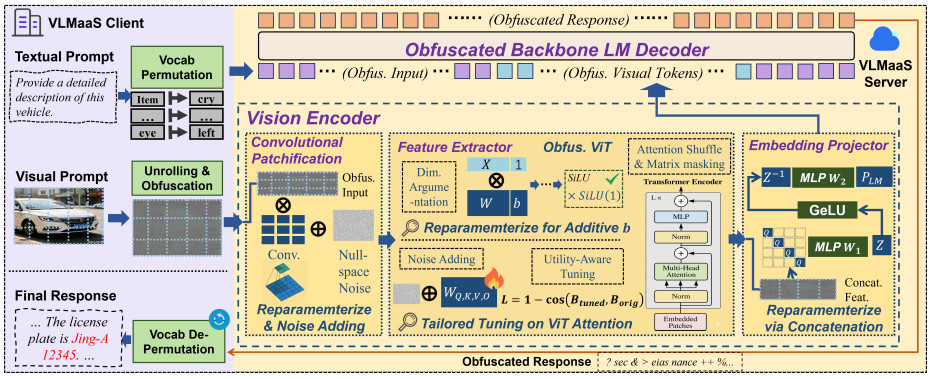

Fig. 2: Detailed illustration of VLM architectures and the COVERT framework.

However, mainstream VLMs couple this LLM backbone with a distinct vision encoder $\mathcal{V}$, rendering the direct application of text-centric primitives insufficient. While AloePri readily secures textual prompts, protecting continuous visual inputs necessitates obfuscation mechanisms explicitly tailored for $\mathcal{V}$. This requirement stems from two fundamental structural disparities. First, the vision encoder introduces heterogeneous architectures, such as convolutional modules, which demand dedicated algebraic mappings. Second, continuous visual signals are inherently characterized by dense spatial correlations and local smoothness. While inherently resilient to feature inversion attacks [17, 24, 30], naive reuse of textual obfuscation completely neglects the rich spatial priors preserved in the shallow representations of ViT-based encoders, which is vulnerable to hidden state inversion attacks [7]. Overcoming these modality-aware vulnerabilities strictly requires a customized covariant obfuscation framework.

4 The COVERT Framework: Covariant Obfuscation for Practical Privacy-Preserving VLMaaS

To address the above challenges, this paper introduces **COVERT**, the first practical privacy-preserving VLM inference framework. Since the backbone LLM in VLMs seamlessly inherits the obfuscation primitives established by AloePri [18], COVERT exclusively targets the vision encoder. Specifically, COVERT systematically derives exact reparameterization transformations tailored to the distinct architectural properties of each visual module, together with calibrated null-space perturbations and utility-preserving stochastic tuning to effectively neutralize hidden-state inversion attacks [7]. Fig. 2 illustrates the overview of the COVERT framework and details the obfuscation across the visual pipeline. For notational consistency, $\hat{\square}$ denotes sampled stochastic variables, $\tilde{\square}$ indicates obfuscated entities, and $\square^{+}$ represents the dimension-augmented matrix.

4.1 Covariant Convolutional Patchification.

As the entry point of the visual pipeline, the *convolutional patcher* f_{Conv} maps visual inputs into patches. To facilitate covariant transformation, we formulate the 3D convolution as a vectorized matrix multiplication. By defining the flattened kernel volume $D = C_{\text{in}} \cdot T_c \cdot H_c \cdot W_c$ and the patch count $N = \frac{T}{T_c} \cdot \frac{H}{H_c} \cdot \frac{W}{W_c}$, the input volume and convolutional kernels are unrolled into matrices $\mathbf{X}_v \in \mathbb{R}^{N \times D}$ and $\mathbf{W}_{\text{conv}} \in \mathbb{R}^{D \times C_{\text{out}}}$, respectively. Here, $C_{\{\text{in,out}\}}$ represents channel dimensions, while T, H, and W denote the temporal frame count, height, and width, respectively. The resultant visual patches are generated as: $f_{\text{Conv}}(\mathbf{X}_v) = \mathbf{X}_v \mathbf{W}_{\text{conv}}$.

To instantiate covariant obfuscation, the client constructs a stochastic transformation pair $(\hat{\mathbf{P}}_v, \hat{\mathbf{Q}}_v)$ satisfying $\hat{\mathbf{P}}_v \hat{\mathbf{Q}}_v = \mathbf{I}_D$, where the expanded matrix $\hat{\mathbf{P}}_v \in \mathbb{R}^{D \times (D + d_{\text{extra}})}$ inflates the volume of $\mathbf{X}_v$. During offline obfuscation, the convolutional parameter are transformed as $\tilde{\mathbf{W}}_{\text{Conv}} = \hat{\mathbf{Q}}_v \mathbf{W}_{\text{Conv}} \hat{\mathbf{P}}_{\text{Pat}}$, where $\hat{\mathbf{P}}_{\text{Pat}}$ denotes the patch mask structurally coupled with the subsequent modules. In online inference, to prevent deterministic mapping, the client injects null-space noise into each row $\mathbf{r}_i$ of the projected input $\mathbf{X}_v \hat{\mathbf{P}}_v$. Specifically, the obfuscated input $\tilde{\mathbf{X}}_v$ is constructed row-wise as $\tilde{\mathbf{r}}_i = \mathbf{r}_i + \mathbf{v}_i$, where the stochastic vector $\mathbf{v}_i$ satisfies the constraint $\mathbf{v}_i \hat{\mathbf{Q}}_v = \mathbf{0}$. Consequently, when the server computes $\tilde{f}_{\text{Conv}}(\mathbf{X}_v) = \tilde{\mathbf{X}}_v \tilde{\mathbf{W}}_{\text{Conv}} = \mathbf{X}_v \mathbf{W}_{\text{Conv}} \hat{\mathbf{P}}_{\text{Pat}}$, the noise $\mathbf{v}_i$ is exactly canceled out. Thus, this isomorphic design enforces zero-error commutation ($e_C = 0$), suppressing patch-level inversion observability while preserving multimodal utility.

4.2 Adaptive Feature Extraction based on ViT.

Subsequently, the obfuscated visual patches generated by $\tilde{f}_{\text{Conv}}$ propagate through the *vision feature extractor* f_{Feat}. In mainstream VLM architectures, this module is typically instantiated as a Vision Transformer (ViT) [8]. Crucially, because the underlying structure of ViT is consistent with those in textual LLMs, f_{Feat} seamlessly inherits the covariant obfuscation primitives established in AloePri [18]. Specifically, this obfuscation process cancels the patch mask $\hat{\mathbf{P}}_{\text{Pat}}$ via weight obfuscation and applies a stochastic feature mask $\hat{\mathbf{P}}_{\text{Feat}}$, producing the protected representation $\tilde{\mathbf{F}} = \mathbf{F} \hat{\mathbf{P}}_{\text{Feat}} = f_{\text{Feat}}(\tilde{f}_{\text{Conv}}(\mathbf{X}_v) \hat{\mathbf{Q}}_{\text{Pat}}) \hat{\mathbf{P}}_{\text{Feat}}$, where $\hat{\mathbf{P}}_{\text{Pat}} \hat{\mathbf{Q}}_{\text{Pat}} = \mathbf{I}$. However, adapting these primitives for VLMs necessitates resolving two critical hurdles: structural incompatibilities caused by pervasive additive biases and layer-specific vulnerabilities to hidden state inversion [7].

Adaptation to Pervasive Additive Biases. First, pervasive additive biases (e.g., the $\mathbf{b}$ in ViT linear layers $\mathbf{W}\mathbf{X} + \mathbf{b}$) are fundamentally incompatible with standard multiplicative obfuscation. In QKV projections, Rotary Position Embeddings (RoPE) restrict admissible masks to highly structured forms (e.g., block-diagonal matrices), reducing their degrees of freedom to $\mathcal{O}(n)$, where n denotes the embedding dimension. Consequently, directly masking the bias via $\mathbf{b}\hat{\mathbf{P}}$ is insecure, as the limited unknowns leave $\hat{\mathbf{P}}$ easily invertible by an adversary. To resolve this, we employ *dimensional augmentation*. By appending a constant unit channel to the plaintexts ($\mathbf{X}^+ = (\mathbf{X}, 1)$) and parameters

($\mathbf{W}_{QKV}^{+} = [\mathbf{W}_{QKV}^{\top}; \mathbf{b}_{QKV}^{\top}]^{\top}$), we securely absorb the biases into the obfuscated weights $\tilde{\mathbf{W}} = \hat{\mathbf{Q}}\mathbf{W}^{+}$. Conversely, this augmented dimension cannot survive the global non-linearity of the softmax operator, meaning it is discarded prior to the output projection $\mathbf{W}_O$. Fortunately, because $\mathbf{W}_O$ operates without RoPE constraints, we can securely apply direct bias masking: $\tilde{\mathbf{b}}_O = \mathbf{b}_O\hat{\mathbf{P}}_O$. This approach is cryptographically robust; the fully dense mask $\hat{\mathbf{P}}_O$ introduces $\mathcal{O}(n^2)$ unknown variables against only $\mathcal{O}(n)$ known equations, forming a mathematically intractable system. Finally, to address FFN gating functions (e.g., SiLU [11]) that non-linearly distort the augmented unit dimension to SiLU(1), we restore the structural invariant by scaling the augmented down-projection weights by $\text{SiLU}(1)^{-1}$, formalizing bias alignment as a zero-error operation.

Tailored Parameter Tuning Against Hidden State Inversion. While the aforementioned adaptations guarantee functional isomorphism, empirical observations indicate that attackers can still exploit structural priors in shallow layers, particularly the attention scores, to launch high-fidelity hidden-state inversion attacks [7]. As shown in Fig. 3, such adaptive attacks on the untuned strawman can still leak sensitive details like license plates. To defend the attack, COVERT thereby introduces utility-aware parameter tuning to prevent this exploitability.

This process is initiated by injecting Gaussian noise characterized by ($\mathcal{N}(0, \alpha \cdot \text{std}(\mathbf{W}_Q))$, $\mathcal{N}(0, \alpha \cdot \text{std}(\mathbf{W}_K))$, $\mathcal{N}(0, \alpha \cdot \text{std}(\mathbf{W}_V))$, $\mathcal{N}(0, \alpha \cdot \text{std}(\mathbf{W}_O))$) into the attention parameters $\{\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O\}$ of the first ViT block, with the hyper-parameter α determining the noise magnitude. Crucially, this noise generation is cryptographically bound to a user-held private seed. Subsequently, using a public image dataset, the client selectively fine-tunes these attention matrices and the preceding normalization layer $\mathbf{W}_{\text{norm1}}$, while freezing the FFN weights $\mathbf{W}_{\text{FFN}}$ and the succeeding normalization layer $\mathbf{W}_{\text{norm2}}$.

Let $\mathcal{B}^{(1)}(\cdot)$ denote the forward function of the first ViT block. The optimization objective is to maximize the cosine similarity between the hidden states output by the tuned block $\mathcal{B}^{(1)}_{\text{tuned}}$ and the original block $\mathcal{B}^{(1)}_{\text{orig}}$:

$$\mathcal{L} = 1 - \cos\left(\mathcal{B}^{(1)}_{\text{tuned}}(\mathbf{x}_i), \mathcal{B}^{(1)}_{\text{orig}}(\mathbf{x}_i)\right).$$

This semantic alignment ensures that the final hidden states remain maximally consistent with the original model to preserve utility, whereas the intermediate attention patterns diverge significantly. Since the noise injection is seeded by a private key, the exact training trajectory remains irreproducible to the attacker, decisively thwarting gradient-based inversion attempts.

4.3 Semantic Embedding Projection and Universal Alignment

Finally, f_{Proj} bridges the modality gap by projecting visual features into the LLM's semantic space via a Multi-Layer Perceptron (MLP) [16, 19]. To compress context length while preserving spatial semantics, advanced architectures may incorporate spatio-temporal aggregation [2,6]. Without loss of generality, we

examine Qwen2.5-VL [2], which concatenates adjacent tokens across a 2×2 spatial grid. Given an obfuscated ViT feature $\tilde{\mathbf{F}}$ with channel dimension D_{Feat}, this concatenation inherently expands the initial input dimension to $D_{\text{in}} = 4D_{\text{Feat}}$. Consequently, the MLP for the subsequent projection is parameterized by two sequential weight matrices: $\mathbf{W}_1 \in \mathbb{R}^{D_{\text{in}} \times D_{\text{hidden}}}$ and $\mathbf{W}_2 \in \mathbb{R}^{D_{\text{hidden}} \times D_{\text{out}}}$.

During offline obfuscation, the client neutralizes incoming feature masks by constructing a block-diagonal inverse $\hat{\mathbf{M}} = \mathbf{I}_4 \otimes \hat{\mathbf{Q}}_{\text{Feat}}^{-1}$ (subject to $\hat{\mathbf{P}}_{\text{Feat}}\hat{\mathbf{Q}}_{\text{Feat}} = \mathbf{I}$), where $\otimes$ denotes the Kronecker product [27]. The client also generates a random permutation matrix $\mathbf{Z} \in \mathbb{R}^{D_{\text{hidden}} \times D_{\text{hidden}}}$ for the hidden layer, alongside an alignment mask $\hat{\mathbf{P}}_{\text{Emb}}$ shared across both vision and text embeddings. The MLP weights are then obfuscated as $\tilde{\mathbf{W}}_1 = \hat{\mathbf{M}}\mathbf{W}_1\mathbf{Z}$ and $\tilde{\mathbf{W}}_2 = \mathbf{Z}^{-1}\mathbf{W}_2\hat{\mathbf{P}}_{\text{Emb}}$.

In online inference, the server processes concatenated obfuscated features $\tilde{\mathbf{F}}_{\text{concat}} = [\tilde{\mathbf{F}}_0, \tilde{\mathbf{F}}_1, \tilde{\mathbf{F}}_2, \tilde{\mathbf{F}}_3]$ and the obfuscated projection computes: $\tilde{f}_{\text{Proj}}(\tilde{\mathbf{F}}_{\text{concat}}) = \text{GeLU}(\tilde{\mathbf{F}}_{\text{concat}}\tilde{\mathbf{W}}_1)\tilde{\mathbf{W}}_2 = f_{\text{Proj}}(\mathbf{F}_{\text{concat}})\hat{\mathbf{P}}_{\text{Emb}}$. This exact cancelation eliminates the obfuscation error and seamlessly integrates into the obfuscated LLM [18].

5 Evaluations

5.1 Experimental Setup

Datasets. COVERT is evaluated on five representative multimodal benchmarks supported by `VLMEvalKit` [10], covering general perception, reasoning, and scientific understanding tasks:

- `Dataset 1` (General VQA): evaluates multimodal understanding through various multi-choice Visual Question Answering (VQA) tasks organized by a hierarchical capability taxonomy.
- `Dataset 2` (Multidisciplinary Reasoning): assesses reasoning abilities requiring broad domain knowledge and deliberate reasoning.
- `Dataset 3` (Vision-Centric VQA): focuses on challenging vision-dependent multimodal understanding which cannot be answered without visual input.
- `Dataset 4` (Scientific Diagram Understanding): benchmarks the understanding of diagram-centric scientific content.
- `Dataset 5` (Mathematical Reasoning): evaluates quantitative, mathematical, and visual reasoning capabilities.

Models and Baselines. COVERT is implemented on models: `Qwen2.5-VL` [2], `MiMO-RL` [14], and `Fara` [1] with various sizes (7B, 32B, 72B). For evaluation baselines, we compare COVERT against *Plaintext (No Defense)* (the theoretical utility upper bound) and *DP-Forward* [9] (a state-of-the-art DP baseline). Heavyweight cryptographic protocols such as PrivMLLM [32] are excluded due to their prohibitive computational overhead.

Evaluation Metrics. (1) *Accuracy:* Based on `VLMEvalKit` [10], we utilize LLM-as-a-Judge with `InternVL3.5-14B` [28] for semantic correctness. (2) *Privacy:* While covariant obfuscation inherently neutralizes conventional inversion [18], We quantify robustness against hidden-state inversion attacks [7] by

evaluating the reconstruction quality of inverted visual prompts via MSE and SSIM [12]. Higher MSE and lower SSIM mean stronger privacy. (3) *Efficiency:* We quantify communication overhead and online inference latency, respectively.

Implementation Details. For the utility-aware tuning, we randomly sample 5,000 images from public datasets to serve as the public dataset. By restricting the tunable depth to the shallowest ViT attention layer ($k = 1$), we maintain a lightweight overhead of $\sim$4M trainable parameters. This layer is trained for 3 epochs using the AdamW optimizer [20] with a learning rate of 10^{-4} and a base noise level of $\alpha = 6$. For the remaining obfuscation hyperparameters, including dimensional augmentation and parameter noise, we adopt the configurations established in AloePri [18]. Finally, during evaluation, we perform VLM inference using the default `VLMEvalKit` settings: greedy decoding (temperature = 0.0), max_new_tokens = 2048, and dynamic image resolutions bounded within $[1280, 16384] \times 28 \times 28$ pixels.

5.2 Utility Evaluation of COVERT

We evaluate the utility preservation of COVERT and its 8-bit quantized (*Quant*) implementations across various VLM scales and variants, as detailed in Table 1. Across five representative benchmarks, COVERT consistently maintains high fidelity, bounding the average accuracy degradation strictly between 1.53% and 2.81%. Notably, Part I reveals a distinct scaling law: the 2.58% initial drop of the 7B model (2.81% for *Quant*) narrows to a highly resilient 1.83% $\sim$ 2.28% (1.68% $\sim$ 2.42% for *Quant*) in the 32B and 72B versions. This trend demonstrates that the superior inherent redundancy of larger architectures makes them significantly more resilient to the parameter perturbations introduced by COVERT. Furthermore, Part II demonstrates robust generalizability across diverse variants like MiMO [14] and Fara [1]. Crucially, the *Quant* versions incur negligible additional loss, confirming the seamless compatibility of our algebraic framework with standard model compression techniques. Ultimately, restricting the average utility degradation to under 3% across all settings decisively validates the utility-preserving nature of COVERT.

The observed minor accuracy degradation stems from intertwined numerical and evaluative factors. Numerically, the heuristic approximation in normalization using obfuscated intermediate states [18] introduces slight deviations in feature scaling within deeper layers. Furthermore, security-oriented parameter noise and 8-bit quantization inherently incur controlled precision losses. Meanwhile, regarding evaluation error, the LLM-as-a-Judge occasionally yields false negatives due to benign stochastic phrasing shifts, penalizing semantically correct outputs. Nevertheless, COVERT's precise structural calibrations successfully bound these cumulative errors to under 3%, fully preserving utility for practical deployment.

5.3 Robustness Against Hidden State Inversion Attacks

As covariant obfuscation naturally shields raw features and responses from conventional privacy attacks [17,24,30], our comprehensive privacy evaluation specif-

Table 1: Utility preservation of COVERT across varying model scales and variants. The COVERT accuracy is reported alongside the degradation (↓) compared to the plaintext baseline. Here, *Quant* denotes the 256-level quantization.

COVERT Setting	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Avg.	Δ Avg.
Part I: Scaling Law across Model Sizes (Qwen2.5-VL Family)							
Qwen2.5-VL-7B	82.97	55.11	64.87	85.01	68.10	71.21	-
COVERT	79.79 (↓3.18)	51.89 (↓3.22)	61.80 (↓3.07)	83.87 (↓1.14)	65.80 (↓2.30)	68.63	↓2.58
COVERT *(Quant)*	78.41 (↓4.56)	51.56 (↓2.55)	61.67 (↓3.20)	83.94 (↓1.07)	66.40 (↓1.70)	68.40	↓2.81
Qwen2.5-VL-32B	86.38	68.67	69.07	84.84	73.70	76.53	-
COVERT	83.59 (↓2.79)	65.33 (↓3.34)	66.93 (↓2.14)	83.65 (↓1.19)	74.00 (↑0.30)	74.70	↓1.83
COVERT *(Quant)*	83.90 (↓2.48)	65.44 (↓3.23)	67.23 (↓1.84)	84.49 (↓0.35)	73.20 (↓0.50)	74.85	↓1.68
Qwen2.5-VL-72B	88.08	67.55	70.80	89.11	75.20	78.15	-
COVERT	87.00 (↓1.08)	64.56 (↓2.99)	67.33 (↓3.47)	87.18 (↓1.93)	73.30 (↓1.90)	75.87	↓2.28
COVERT *(Quant)*	86.53 (↓1.55)	64.22 (↓3.33)	67.47 (↓3.33)	87.05 (↓2.06)	73.40 (↓1.80)	75.73	↓2.42
Part II: Generalization on Different Model Variants (7B Scale)							
Mimo-7B	84.13	65.55	69.80	86.85	70.30	75.32	-
COVERT	82.66 (↓1.47)	63.44 (↓2.11)	68.20 (↓1.60)	85.36 (↓1.49)	69.30 (↓1.00)	73.79	↓1.53
COVERT *(Quant)*	82.89 (↓1.24)	61.78 (↓3.77)	69.33 (↓0.47)	85.20 (↓1.65)	69.50 (↓0.80)	73.74	↓1.58
Fara-7B	65.09	52.78	55.93	80.47	58.30	62.51	-
COVERT	61.07 (↓4.02)	47.22 (↓5.56)	53.20 (↓2.73)	79.83 (↓0.64)	57.20 (↓1.10)	59.70	↓2.81
COVERT *(Quant)*	61.92 (↓3.17)	47.33 (↓5.45)	54.33 (↓1.60)	79.75 (↓0.72)	56.10 (↓2.20)	59.88	↓2.63

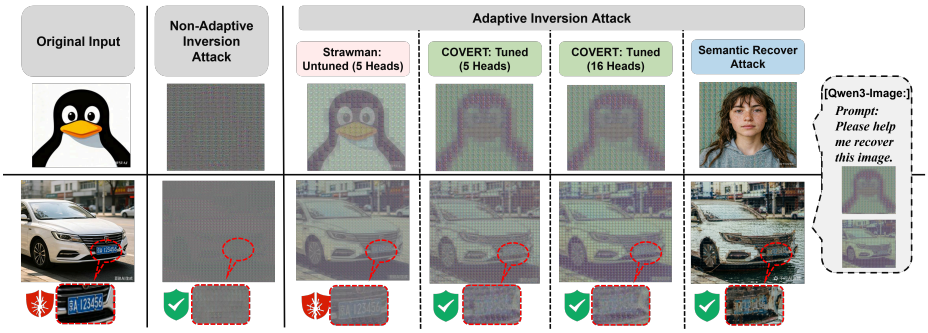

Fig. 3: Qualitative defense against inversion attacks. Non-adaptive inversion directly yields pure noise. Under adaptive attacks, the untuned strawman leaks sensitive details (e.g., license plates). Conversely, COVERT robustly conceals all visual privacy against practical inversions, and completely prevents advanced semantic recovery.

ically targets hidden-state inversion attacks [7]. Here, adversaries attempt to reconstruct the visual prompt from extracted attention scores from the ViT module (f_{Feat}). Because COVERT obfuscates ViT via head shuffling [18], we evaluate two settings respectively: a *non-adaptive* attacker targeting raw obfuscated features, and an *adaptive* attacker brute-forcing the attention head permutations.

Defending Against Non-Adaptive Inversion. When adversaries directly attack the shuffled attention scores without resolving the permutation mappings,

the inversion yields pure structural noise (column 2 in Fig. 3). This confirms that COVERT destroys the spatial information required for standard reconstruction.

Defending Against Adaptive Inversion. As to the worst-case scenario, we assume that the adversary is aware of the attention block permutation strategy [18]. Although fully resolving all mappings is computationally intractable, the adversary can attempt a partial resolution. For example, solving the permutations of 16 attention heads confronts a massive search space of $\mathcal{O}(P_{16}^8) \approx 5.18 \times 10^8$ combinations. Therefore, we establish solving 16 attention heads as the computational upper bound for feasible adaptive attacks. We evaluate this adaptive attack on 100 images from public datasets, quantifying privacy risks via SSIM and MSE to measure reconstruction fidelity.

As reported in Table 2, the threat of hidden-state inversion is highly localized. In the untuned baseline, spatial information degrades rapidly as network depth increases, leaving only the shallowest layer in ViT (Layer 1) vulnerable to visual recovery with MSE around 3000, while deeper layers remain naturally secure with MSE over 5000. Recognizing this specific bottleneck, COVERT employs a tailored tuning strategy to precisely protect this exposure. Consequently, this targeted tuning severely degrades reconstruction fidelity on Layer 1 compared to the untuned strawman: SSIM [12] drops drastically by 54.46% (suppressed to 0.1550, safely aligning with the security level of deeper layers), while MSE increases by 39.75%. This empirically proves that adversaries cannot successfully invert the hidden states, ensuring robust end-to-end visual privacy.

These quantitative gains in SSIM and MSE align directly with robust visual concealment. As shown in Fig. 3, under a 5-head adaptive attack, the strawman version without tailored tuning catastrophically leaks sensitive semantics (e.g., legible license plates). Meanwhile, COVERT with tailored tuning in shallow ViT attention parameters completely prevents this leakage, rendering critical details unrecognizable. Even exponentially scaling the attack to 16 heads yields no visual improvement, as shown in Fig. 3. Therefore, under practical assumptions, the adversary can only brute-force limited attention head permutation mappings, which is inadequate to recover sensitive visual details. Furthermore, advanced semantic recovery via state-of-the-art commercial models (e.g., `Qwen-Image` [29]) completely fails, producing only irrelevant hallucinations. This establishes COVERT's state-of-the-art resilience against spatial reconstruction attacks. Crucially, this strong privacy protection does not come at the expense of task performance: as shown in Table 1, COVERT with tailored tuning incurs only a negligible accuracy drop (less than 3%). Moreover, such tuning is highly efficient, requiring only 1.5 hours of client-side computation on a single GPU or about 6.5 hours on a 20-core CPU.

5.4 Efficiency of COVERT Inference

Finally, we analyze COVERT's system efficiency, including communication overhead from visual obfuscation and online inference latency from matrix expansion.

First, obfuscating continuous visual signals inherently amplifies the client-to-server transmission payload. As detailed in Table 3, obfuscating a plaintext

Table 2: Robustness against hidden-state inversion [7]. Tailored tuning effectively secures Layer 1 in ViT, the only vulnerable shallowest layer.

Target & Setting	SSIM ↓	MSE ↑
Untuned Obfuscation (Baseline)		
Layer 1 (Vulnerable)	0.3404	3275.28
Layer 2 (Naturally Safe)	0.2205	5737.62
Layer 3 (Naturally Safe)	0.1575	5541.46
Layer 4 (Naturally Safe)	0.1078	5855.34
COVERT (Tailored Tuning)		
Layer 1 (Protected)	**0.1550**	**4577.28**
Security Gain (L1)	*-54.46%*	*+39.75%*

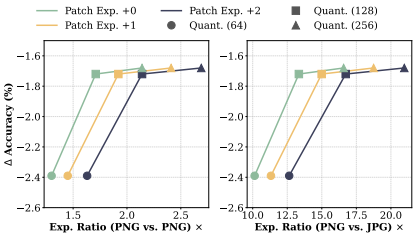

Fig. 4: Communication-utility trade-off. 64-bin quantization ("Quant.") limits its transmission expansion ("Exp.") to 1.30× at a minor 2.39% accuracy cost.

Table 3: Communication expansion and entropy. Obfuscation maximizes differential entropy to ensure security.

Format/	Expansion Ratio		Diff. Entropy	
Metric	Mean	Med.	Hor.	Ver.
Plaintext	1.00×	1.00×	1.90	2.18
Obfus (PNG)	7.68×	6.94×	7.07	7.06
Obfus (JPG)	32.15×	32.34×	7.07	7.06

Table 4: Online inference efficiency. COVERT maintains nearly identical latency compared to plaintext.

Setting	TTFT (ms) ↓	TPOT (ms) ↓	Throughput (tokens/s) ↑
Plaintext	3864.89	26.73	25.17
COVERT	3898.38 (+0.86%)	29.26 (+8.65%)	23.76 (-5.93%)

image incurs a 7.68× size expansion compared to lossless PNG, and a 32.15× increase against lossy JPG. This overhead stems primarily from *dimensional inflation*: reparameterizing convolutions into exact matrix transformations requires duplicating images along the temporal channel and expanding the receptive field (e.g., from 14×14 to 16×16). Second, to strictly preserve inference accuracy, the obfuscated tensor must be transmitted via lossless compression (e.g., PNG), precluding bandwidth-friendly lossy formats. Finally, obfuscation systematically disrupts the spatial locality of plaintext images, driving both horizontal and vertical differential entropy from ∼2.0 to ∼7.0 (Table 3). While this entropy surge provides a robust shield against spatial inversion, it fundamentally neutralizes the efficacy of standard spatial compression algorithms.

Therefore, to alleviate communication bottlenecks, COVERT can be optimized by jointly tuning the dimensional inflation of $\hat{\mathbf{P}}_v$ and the quantization levels. Aggressively quantizing the obfuscated tensor into fewer discrete bins (e.g., 64) effectively reduces communication payloads with minor utility degradation. As illustrated in Fig. 4, minimizing patch expansion and scaling quantization levels down to 64 reduces the transmission expansion to merely 1.30×, incurring a moderate accuracy drop of 2.39%.

Meanwhile, preserving real-time inference efficiency is also fundamental for practical VLMaaS. To benchmark this for COVERT, we deploy `Qwen2.5-VL-32B` using the `vLLM` engine under a high-concurrency regime (30 parallel requests, 1064 visual tokens per prompt). As reported in Table 4, COVERT incurs a

Table 5: Utility comparison on Qwen2.5-VL-7B. While DP-Forward suffers catastrophic accuracy collapse across all privacy budgets ($\epsilon, \delta = 1e-5$), COVERT maintains high-fidelity multimodal performance. *Absolute drops are in parentheses.*

Method	Setting	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5
Plaintext	Unprotected	82.97%	55.11%	64.87%	85.01%	68.10%
DP-Forward [9]	$\epsilon = 128$	0.07% (-82.90%)	13.89% (-41.22%)	5.93% (-58.94%)	13.76% (-71.25%)	34.10% (-34.00%)
	$\epsilon = 64$	0.31% (-82.66%)	18.89% (-36.22%)	6.06% (-58.81%)	19.00% (-66.01%)	32.70% (-35.40%)
	$\epsilon = 16$	0.62% (-82.35%)	24.11% (-31.00%)	10.13% (-54.74%)	34.22% (-50.79%)	33.20% (-34.90%)
	$\epsilon = 4$	0.07% (-82.90%)	18.33% (-36.78%)	6.33% (-58.54%)	19.17% (-65.84%)	35.90% (-32.20%)
COVERT (Ours)	**Obfus.+Tuning**	**79.79%** (-3.18%)	**51.89%** (-2.22%)	**61.80%** (-3.07%)	**83.87%** (-1.14%)	**65.80%** (-2.30%)

negligible computational penalty. The Time-To-First-Token (TTFT) experiences a marginal +0.86% overhead and the Time Per Output Token (TPOT) sees a modest +8.65% increase, culminating in an overall throughput decrease of merely 5.93%. This confirms that our framework seamlessly inherits the highly optimized KV-cache and parallel decoding pipelines of modern VLM serving engines, guaranteeing uncompromising online efficiency.

5.5 Comparison with Baselines: Utility and Efficiency

To further demonstrate the feasibility of COVERT, we benchmark it against two state-of-the-art private VLM paradigms: DP-Forward [9] and MPC-based PrivMLLM [32]. Note that a deployable private VLMaaS must simultaneously satisfy the following: robust privacy, preserved utility, and lightweight inference overhead. As shown below, existing baselines catastrophically fail in at least one dimension, whereas COVERT achieves an optimal and practical balance.

Utility Comparison vs. Perturbation Methods. For fair comparison, we adapt DP-Forward [9] to Qwen2.5-VL-7B [2] by injecting differential privacy noise after the visual patch and textual LLM embeddings. As Table 5 shows, such noise fundamentally destroys VLM utility. Even with a relaxed privacy budget ($\epsilon = 128$), accuracy completely collapses: for example, `Dataset 1` performance falls from 82.97% to 0.07%, and `Dataset 4` drops from 85.01% to 13.76%. In stark contrast, COVERT preserves utility while protecting privacy. Even under rigorous defensive settings, the accuracy of COVERT only drops to 83.87% on `Dataset 4` and 79.79% on `Dataset 1`. This confirms that COVERT achieves an effective utility-privacy trade-off where traditional DP methods fall short.

Efficiency Comparison vs. Cryptographic Methods. We further contrast COVERT with PrivMLLM [32], an MPC-based cryptographic baseline. While achieving near-lossless utility, PrivMLLM imposes prohibitive efficiency penalties: it relies on a restrictive three-server setup, demanding 2.69 GB of communication and over 5 minutes of WAN latency for a single request, rendering it entirely impractical for real-time cloud deployment [32]. Conversely, COVERT operates efficiently within standard client-server architectures. By shifting obfuscation offline via exact reparameterization, it bypasses online cryptographic interactions entirely. Under high-concurrency settings (Table 4), COVERT in-

curs negligible penalties: a +0.86% Time-To-First-Token (TTFT) overhead and a mere 5.93% throughput reduction. With client-side communication expansion aggressively optimized to 1.30× via quantization (Fig. 4), COVERT securely delivers the only practical, real-time solution for VLMaaS.

6 Conclusion

To address critical privacy vulnerabilities in the deployment of VLMaaS, this paper introduces **COVERT**, the first practical privacy-preserving VLM inference framework to the best of our knowledge. Specifically, COVERT introduces a modality-adaptive covariant obfuscation paradigm. By implementing exact architectural reparameterization across the visual pipeline with a utility-aware tailored tuning strategy, COVERT effectively defends high-fidelity hidden-state inversion attacks. Extensive evaluations across diverse VLMs demonstrate that COVERT achieves a favorable balance: robust end-to-end visual privacy, negligible utility degradation (less than 3%), and highly efficient online inference (less than 6% overhead). In conclusion, COVERT bridges the gap between robust privacy protection and real-world VLMaaS deployment, paving the way for secure, scalable, and practical cross-modal AI services.

Acknowledgements

This work was partially funded by NSFC Grants No.62272222 and 62272215, Jiangsu Province Outstanding Youth Fund Project (No. BK20230080).

References

1. Awadallah, A., Lara, Y., Magazine, R., Mozannar, H., Nambi, A., Pandya, Y., Rajeswaran, A., Rosset, C., Taymanov, A., Vineet, V., et al.: Fara-7b: An efficient agentic model for computer use. arXiv preprint arXiv:2511.19663 (2025)
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
3. Beaudry, N.J., Renner, R.: An intuitive proof of the data processing inequality. Quantum Info. Comput. (2012)
4. Caldarella, S., Mancini, M., Ricci, E., Aljundi, R.: The phantom menace: unmasking privacy leakages in vision-language models. In: European Conference on Computer Vision. pp. 435–451. Springer (2024)
5. Chen, T., Li, P., Zhou, K., Chen, T., Wei, H.: Unveiling privacy risks in multi-modal large language models: Task-specific vulnerabilities and mitigation challenges. In: Findings of the Association for Computational Linguistics: ACL 2025. pp. 4573–4586 (2025)
6. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 24185–24198 (2024)

7. Dong, T., Meng, Y., Li, S., Chen, G., Liu, Z., Zhu, H.: Depth gives a false sense of privacy:{LLM} internal states inversion. In: 34th USENIX Security Symposium (USENIX Security 25). pp. 1629–1648 (2025)
8. Dosovitskiy, A.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
9. Du, M., Yue, X., Chow, S.S., Wang, T., Huang, C., Sun, H.: Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. pp. 2665–2679 (2023)
10. Duan, H., Yang, J., Qiao, Y., Fang, X., Chen, L., Liu, Y., Dong, X., Zang, Y., Zhang, P., Wang, J., et al.: Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 11198–11201 (2024)
11. Elfwing, S., Uchibe, E., Doya, K.: Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural networks **107**, 3–11 (2018)
12. Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: 2010 20th international conference on pattern recognition. pp. 2366–2369. IEEE (2010)
13. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C.H., Gonzalez, J., Zhang, H., Stoica, I.: Efficient memory management for large language model serving with pagedattention. In: Proceedings of the 29th Symposium on Operating Systems Principles. p. 611–626. SOSP '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3600006.3613165, https://doi.org/10.1145/3600006.3613165
14. Li, J., Chen, J., Qu, Y., Ju, J., Luo, Z., Luan, J., Xu, S., Lin, Z., Zhu, J., Xu, B., et al.: Xiaomi mimo-vl-miloco technical report. arXiv preprint arXiv:2512.17436 (2025)
15. Li, Y., Lai, Z., Bao, W., Tan, Z., Dao, A., Sui, K., Shen, J., Liu, D., Liu, H., Kong, Y.: Visual large language models for generalized and specialized applications. arXiv preprint arXiv:2501.02765 (2025)
16. Li, Z., Wu, X., Du, H., Liu, F., Nghiem, H., Shi, G.: A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges. arXiv preprint arXiv:2501.02189 (2025)
17. Lin, Y., Zhang, Q., Cai, Q., Hong, J., Ye, W., Liu, H., Duan, B.: An inversion attack against obfuscated embedding matrix in language model inference. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. pp. 2100–2104 (2024)
18. Lin, Y., Zhang, Q., Ruan, W., Zhang, D., Hong, J., Wu, Y., Xia, H., Mao, Y., Zhong, S.: Towards privacy-preserving llm inference via collaborative obfuscation (technical report) (2026), https://arxiv.org/abs/2603.01499
19. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2023)
20. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
21. Mai, P., Yan, R., Huang, Z., Yang, Y., Pang, Y.: Split-and-denoise: Protect large language model inference with local differential privacy. In: Proceedings of the 41st International Conference on Machine Learning. pp. 34281–34302 (2024)
22. Roberts, J., Mylonakis, K., Roy, S., Kale, K.: Learning obfuscations of llm embedding sequences: Stained glass transform. arXiv preprint arXiv:2506.09452 (2025)

23. Tan, Y., Tan, C., Mi, Z., Chen, H.: Pipellm: Fast and confidential large language model services with speculative pipelined encryption. In: Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1. pp. 843–857 (2025)
24. Tömekçe, B., Vero, M., Staab, R., Vechev, M.: Private attribute inference from images with vision-language models. Advances in Neural Information Processing Systems **37**, 103619–103651 (2024)
25. Tong, M., Chen, K., Zhang, J., Qi, Y., Zhang, W., Yu, N., Zhang, T., Zhang, Z.: Inferdpt: Privacy-preserving inference for black-box large language models. IEEE Transactions on Dependable and Secure Computing (2025)
26. Tsaprazlis, E., Feng, T., Ramakrishna, A., Gupta, R., Narayanan, S.: Assessing visual privacy risks in multimodal ai: A novel taxonomy-grounded evaluation of vision-language models. arXiv preprint arXiv:2509.23827 (2025)
27. Van Loan, C.F.: The ubiquitous kronecker product. Journal of computational and applied mathematics **123**(1-2), 85–100 (2000)
28. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., et al.: Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025)
29. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
30. Xiu, K., Zhang, S.Q.: Caprecover: A cross-modality feature inversion attack framework on vision language models. In: Proceedings of the 33rd ACM International Conference on Multimedia. pp. 3808–3816 (2025)
31. Xue, J., Zhao, Y., Zheng, M., Yao, F., Solihin, Y., Lou, Q.: Twinshield: Secure foundation model execution by unifying tees and crypto-protected accelerators
32. Xue, Y., Liu, L., Luo, Y., Sun, B., Fu, S.: Privmllm: Efficient three-party multimodal large language model secure inference supported prompt privacy. In: 2025 IEEE 33rd International Conference on Network Protocols (ICNP). pp. 1–24. IEEE (2025)
33. Zeng, Z., Wang, J., Yang, J., Lu, Z., Li, H., Zhuang, H., Chen, C.: Privacyrestore: Privacy-preserving inference in large language models via privacy removal and restoration. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 10821–10855 (2025)
34. Zhang, W., Guo, Y., Liu, L., Xi, Y., Wang, Y., Guo, F.: Seclmm: Towards secure and lightweight inference service for large multimodal models. In: 2025 IEEE/ACM 33rd International Symposium on Quality of Service (IWQoS). pp. 1–10. IEEE (2025)