UAQB: UTILITY-AWARE QUANTIZATION BOTTLENECK FOR PRIVATE SPLIT VLM INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Split inference allows resource-constrained devices to offload vision-language model (VLM) computation to a server by transmitting intermediate visual tokens instead of raw images, thereby avoiding direct exposure of private image content. However, these visual tokens remain susceptible to reconstruction attacks because substantial spatial and semantic information is retained in them. Existing defenses mitigate leakage largely at the expense of downstream task performance. To address this challenge, we propose the Utility-Aware Quantization Bottleneck (UAQB), a plug-in defense deployed on-device prior to token transmission. UAQB applies adaptive quantization to visual tokens to restrict image information flowing through the bottleneck, coupled with utility adaptation that restores compatibility of the resulting information-limited representation with the downstream VLM. We derive a tractable upper bound on the information bottleneck objective, enabling joint optimization of privacy and utility. Extensive evaluation across multiple utility benchmarks and reconstruction metrics demonstrate that UAQB achieves a more favorable utility-privacy trade-off than existing defenses, with particularly high utility retention at stronger protection levels.

1 INTRODUCTION

Vision-language models (VLMs) support a wide range of visual understanding, question answering, and multimodal reasoning tasks (Liu et al., 2024a; Chen et al., 2024b). Their memory and computation requirements, however, make full on-device inference difficult for resource-constrained devices (Ye et al., 2025; Liang et al., 2025). Cloud inference alleviates these constraints but requires sending potentially private images to a remote service (Liu et al., 2026). Thus, split inference offers a practical alternative: the device runs the vision encoder on the input image and transmits the intermediate visual tokens to the server, which completes inference using the remaining computationally intensive model components (Athanasiou et al., 2026). This keeps the private image on the user's device, avoiding direct exposure of raw image content to the server.

Visual tokens, however, are not a privacy boundary. Intermediate representations can retain substantial instance-specific information and enable reconstruction (He et al., 2019; 2021). Recent reconstruction results further show that even final-layer features of modern vision encoders retain recoverable spatial and semantic structure (Allakhverdov et al., 2026; Athanasiou et al., 2026; Zheng et al., 2026). The same rich visual tokens that support accurate server-side reasoning can therefore expose private image content to attackers.

Existing defenses commonly adopt two strategies: (1) **direct perturbation** of transmitted representations to hinder reconstruction, typically through noise injection, feature dropout, or feature pruning (Mireshghallah et al., 2020; He et al., 2021; Singh et al., 2021); and (2) **adversarial learning** of protected representations against a surrogate reconstruction model (Li et al., 2021; Athanasiou et al., 2026). However, direct perturbations can disrupt visual evidence needed for downstream reasoning, severely degrading utility (Li et al., 2026). Meanwhile, adversarial learning can favor defenses tailored to the chosen adversaries, undermining the reliability of the resulting privacy protection (Singh et al., 2021; Yang et al., 2022).

To address these challenges, we propose the Utility-Aware Quantization Bottleneck (UAQB) defense approach for split VLMs based on the information bottleneck (IB) principle (Tishby et al.,

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

1999). Our bottleneck comprises two modules: a *utility-aware quantizer* that limits the image information retained in visual tokens, and a *utility adapter* that restores compatibility of the resulting information-limited representation with the downstream VLM to maintain utility. The two modules are deployed together on the device before visual-token transmission. They are jointly trained to maintain downstream utility while controlling a tractable upper bound on information leakage, without relying on a reconstruction model or updating any pretrained VLM components.

Our main contributions are as follows:

- We introduce UAQB, a defense method that couples learned quantization with utility adaptation to limit information leakage while maintaining downstream utility. It is deployed at the device only in a plug-and-play manner, without the need to retrain or modify any of the VLM components, including vision encoder, multimodal projector, and language model.
- We derive a tractable variational upper bound on the IB objective. This allows principled joint optimization of utility and privacy that does not depend on any specific reconstruction adversary.
- We evaluate UAQB across multiple VLM benchmarks and privacy metrics. Results show that UAQB achieves superior utility-privacy trade-off over existing defenses, retaining substantially higher utility under stronger privacy protection.

2 Related Work

2.1 Reconstruction Leakage in Split Inference

Feature inversion attacks reconstruct input images from exposed intermediate representations, posing a privacy threat to split inference. For conventional convolutional neural networks (CNNs), early reconstruction methods optimize candidate images to match the observed features, as in regularized maximum likelihood estimation (rMLE) (Mahendran & Vedaldi, 2015; He et al., 2019). Inverse networks can also be trained to map features back to input images using paired image-feature supervision (Dosovitskiy & Brox, 2016; He et al., 2021). Feature-consistency objectives also enable training inverse networks without auxiliary data (Yin et al., 2023). Some studies further exploit image priors encoded in pretrained generative adversarial networks (GANs), reconstructing images through latent-space search (Li et al., 2023b).

Beyond conventional CNNs, feature inversion has been extended to vision encoders used in modern VLMs, showing that image content remains recoverable even from deep-layer features. Exploiting the semantic and spatial information retained in visual features, several studies use diffusion-based generative models to reconstruct images, with these features providing conditioning or guidance (Chen et al., 2024a; Lei et al., 2025; Athanasiou et al., 2026). In addition, recent studies show that a Transformer-based decoder trained to map visual tokens directly to images suffices for high-quality reconstruction in a single forward pass (Allakhverdov et al., 2026; Zheng et al., 2026).

2.2 Defenses against Reconstruction Leakage

To protect input privacy, a common defense strategy is to directly perturb transmitted representations to make reconstruction more difficult for attackers. For CNN-based inference, Shredder learns Laplace noise distributions to perturb activations (Mireshghallah et al., 2020), while Cloak uses learned Gaussian perturbations to identify task-irrelevant features for suppression (Mireshghallah et al., 2021). Other defenses combine noise injection with representation regularization (Xia et al., 2025; Deng et al., 2026) or incorporate random activation dropout (He et al., 2021) and feature compression (Maeng et al., 2023). For language models, DP-Forward perturbs embeddings with calibrated Gaussian noise (Du et al., 2023), while Split-and-Denoise combines input perturbation with client-side denoising of returned embeddings (Mai et al., 2024). Besides noise injection, DEL combines differentially private stochastic quantization of token embeddings with soft-prompt adaptation (Gu et al., 2026). These language-model defenses address text privacy, whereas our setting concerns image reconstruction from transmitted visual tokens.

Another common defense strategy is adversarial learning, which trains protective representations against surrogate reconstruction models. Adversarial objectives guide both feature transformations

and dynamic channel pruning (Li et al., 2021; Singh et al., 2021). These objectives can also be derived from variational information bounds to suppress input and label leakage, as in InfoScissors (Duan et al., 2024). In split VLMs, TrustCLIP trains a feature projection against a fixed pretrained diffusion reconstructor while adapting the downstream task head (Athanasiou et al., 2026). In comparison, our proposed UAQB does not require a reconstruction model for defense training.

3 Problem Formulation

System setup. As in TrustCLIP (Athanasiou et al., 2026), we consider split inference for a pretrained VLM between a trusted user device and a remote server. Given a private image $\mathbf{x} \in \mathcal{X}$, the device applies a vision encoder f_{V} to obtain a sequence of visual tokens

$$\mathbf{h} = f_{\mathrm{V}}(\mathbf{x}), \qquad \mathbf{h} \in \mathbb{R}^{T \times D}, \tag{1}$$

where T and D denote the number of visual tokens and their dimensionality, respectively. In the unprotected pipeline, the visual tokens $\mathbf{h}$ are transmitted to the server, which processes them with the user's text prompt using a multimodal projector and language model.

In our setting, the vision encoder, multimodal projector, and language model remain frozen. The proposed UAQB method does not require any retraining or modification to them.

Threat model. Although the raw image $\mathbf{x}$ remains on the device, the transmitted visual tokens $\mathbf{h}$ may retain substantial instance-specific information about $\mathbf{x}$. We consider a passive reconstruction attacker that observes the visual tokens exposed at the device-server interface and applies a reconstructor f_{R} to recover the corresponding image using only visual tokens. Without protection, this gives

$$\hat{\mathbf{x}} = f_{\mathrm{R}}(\mathbf{h}). \tag{2}$$

Here, we assume a white-box attacker with full knowledge of the deployed split-inference pipeline and any protection mechanism. Using auxiliary images drawn independently of the targets from the same distribution, the attacker may train a reconstructor matched to the transmitted representation.

4 The UAQB Method

With the VLM backbone frozen, conventional defenses that directly perturb transmitted visual tokens can reduce information leakage but also compromise the tokens' compatibility with the downstream model (He et al., 2026), thus severely degrading utility (Li et al., 2026). To address this, we propose the UAQB defense method, which is deployed on the device in a plug-and-play manner before transmission. It limits information leakage about the input image while preserving visual-token compatibility with the frozen downstream model, thereby maintaining utility.

4.1 Utility-Aware Quantization Bottleneck

As illustrated in Figure 1, UAQB consists of a learned quantizer f_{Q} and a utility adapter f_{U}:

$$\mathbf{z} = f_{\mathrm{Q}}(\mathbf{h}), \tag{3}$$

$$\mathbf{y} = f_{\mathrm{U}}(\mathbf{z}), \tag{4}$$

where $\mathbf{z} \in \mathbb{R}^{T \times D}$ is the quantized representation and $\mathbf{y} \in \mathbb{R}^{T \times D}$ denotes the transmitted visual tokens. The two modules are Transformer-based and serve complementary roles: f_{Q} limits the information retained in the quantized representation to reduce leakage, while f_{U} adapts the information-constrained representation to the frozen VLM interface to maintain downstream utility. These components are implemented on the device. Only $\mathbf{y}$ is transmitted to the server and exposed to the attacker.

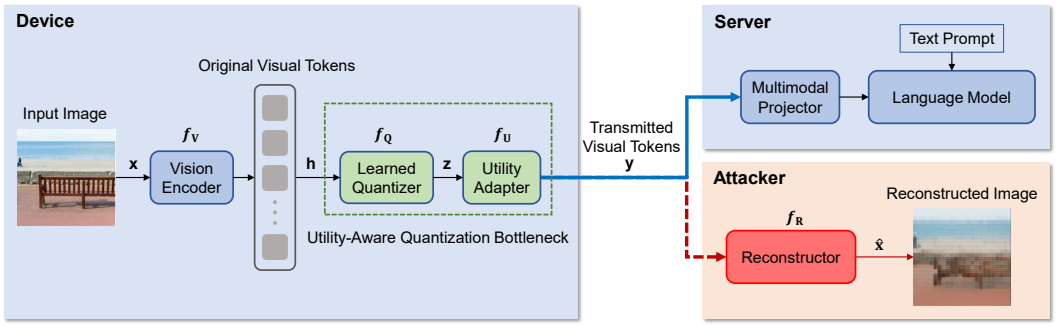


Figure 1: Overview of the proposed UAQB. Blue and green blocks denote frozen VLM components and trainable bottleneck modules, respectively. The red dashed branch denotes the attack path.

Learned quantizer. The learned quantizer f_{Q} consists of an input-dependent offset followed by scalar quantization:

$$\boldsymbol{\mu} = \mathbf{h} + \phi_{\mathrm{Q}}(\mathbf{h}), \tag{5}$$

$$[\mathbf{k}]_{t,d} = \mathrm{round}\left(\frac{[\boldsymbol{\mu}]_{t,d}}{[\boldsymbol{\delta}]_d}\right), \tag{6}$$

$$[\mathbf{z}]_{t,d} = [\boldsymbol{\delta}]_d [\mathbf{k}]_{t,d}, \tag{7}$$

where the learned offset $\phi_{\mathrm{Q}}(\mathbf{h})$ repositions each token feature relative to its quantization grid, the learned step sizes $\boldsymbol{\delta} \in \mathbb{R}^D_{++}$ control quantization precision along each feature dimension, the integer tensor $\mathbf{k}$ denotes the quantization indices, $\mathbf{z}$ is the corresponding dequantized representation mapped back to the pre-quantization feature scale, and $t = 1, \ldots, T$ and $d = 1, \ldots, D$ index token positions and feature dimensions, respectively.

Note that the offset $\phi_{\mathrm{Q}}(\mathbf{h})$ is input-dependent, allowing better bin assignments by incorporating context across visual tokens. In contrast, the shared step sizes $\boldsymbol{\delta}$ are input-independent, forming a fixed quantization grid at inference time that controls the image information passed to the utility adapter.

Utility adapter. Hard quantization can shift the visual-token distribution away from that expected by the frozen multimodal projector, severely degrading downstream utility. We therefore apply a learned utility adapter f_{U} after quantization to restore compatibility with the frozen VLM and maintain utility, yielding the transmitted visual tokens $\mathbf{y} = f_{\mathrm{U}}(\mathbf{z})$.

Since the utility adapter takes only $\mathbf{z}$ as input, the quantization and adaptation pipeline induces the Markov chain

$$X \to H \to K \to Z \to Y, \tag{8}$$

where X, H, K, Z, and Y are the random variables corresponding to $\mathbf{x}, \mathbf{h}, \mathbf{k}, \mathbf{z}$, and $\mathbf{y}$, respectively. Therefore, the data-processing inequality gives the following bound on mutual information:

$$I(X;Y) \le I(X;Z) \overset{(a)}{=} I(X;K) \overset{(b)}{=} H(K), \tag{9}$$

where $I(X;Y)$ measures the information that Y reveals about the private image X, and $H(K)$ is the entropy of the quantization indices. Here, (a) follows from the one-to-one correspondence between K and Z under the fixed quantization grid, and (b) holds because K is deterministic given X. Thus, although the adapter maintains utility, the image information carried by the transmitted adapted tokens $\mathbf{y}$ is bounded above by the image information retained in the quantization indices $\mathbf{k}$. This leads to a privacy guarantee that is stated in the following theorem.

Theorem 1 (Reconstruction-distortion guarantee)**.** *For an image source $X \in \mathbb{R}^N$, let*

$$\mathcal{D}_X(R) = \inf_{\substack{p(\widetilde{\mathbf{x}}|\mathbf{x}):\\ I(X;\widetilde{X}) \le R}} \mathbb{E}\left[\frac{1}{N}\|X - \widetilde{X}\|_2^2\right] \tag{10}$$

denote its distortion-rate function (Cover & Thomas, 2006). Here, N is the dimensionality of the vectorized image, $R \ge 0$ is the information budget, and $\widetilde{X}$ denotes a candidate reconstruction of X.

Under the threat model in Section 3, the expected reconstruction mean squared error (MSE) of any reconstructor satisfies

$$\mathbb{E}\left[\frac{1}{N}\|X - \hat{X}\|_2^2\right] \geq \mathcal{D}_X(H(K))\,. \tag{11}$$

The proof is provided in Appendix A.1. Thus, the learned quantizer places a theoretical limit on the attacker's best achievable reconstruction MSE, while the utility adapter maintains downstream utility without weakening this guarantee.

4.2 Information Bottleneck Optimization

To reduce information leakage while retaining information relevant to downstream tasks, we jointly optimize the quantizer f_{Q} and utility adapter f_{U} according to the IB principle (Tishby et al., 1999):

$$\min_{\theta_{\mathrm{Q}},\theta_{\mathrm{U}}} -I(Y;A) + \beta I(X;Y), \tag{12}$$

where θ_{Q} and θ_{U} denote the trainable parameters of the two modules, and A denotes the target answer-token sequence. The mutual information $I(Y;A)$ measures the task-relevant information retained in the transmitted representation Y. Here, all utility-related information quantities are conditioned on the user's text prompt, which is omitted from the notation. The trade-off parameter $\beta \geq 0$ controls the utility-privacy balance.

However, directly computing these mutual-information terms requires estimating high-dimensional joint distributions and is therefore intractable. We instead minimize a tractable upper bound on the objective in Equation (12), obtained by lower-bounding the utility term $I(Y;A)$ and upper-bounding the leakage term $I(X;Y)$.

Utility lower bound. Let $p(\mathbf{a} \mid \mathbf{y})$ denote the true conditional distribution of A given Y, and let $q(\mathbf{a} \mid \mathbf{y})$ be any variational distribution. Then the non-negativity of the Kullback-Leibler (KL) divergence gives a lower bound on the utility term:

$$\begin{aligned} I(Y;A) &= H(A) + \mathbb{E}_{\mathbf{y},\mathbf{a}}[\log p(\mathbf{a} \mid \mathbf{y})] \\ &= H(A) + \mathbb{E}_{\mathbf{y},\mathbf{a}}[\log q(\mathbf{a} \mid \mathbf{y})] + \mathbb{E}_{\mathbf{y}}[D_{\mathrm{KL}}(p(\cdot \mid \mathbf{y}) \| q(\cdot \mid \mathbf{y}))] \\ &\geq H(A) + \mathbb{E}_{\mathbf{y},\mathbf{a}}[\log q(\mathbf{a} \mid \mathbf{y})]\,. \end{aligned} \tag{13}$$

We take $q(\mathbf{a} \mid \mathbf{y}) = q_{\mathrm{VLM}}(\mathbf{a} \mid \mathbf{y})$, the predictive distribution of the frozen VLM. Since $H(A)$ is fixed by the data distribution, maximizing the lower bound is equivalent to minimizing

$$\mathcal{L}_{\mathrm{util}} = -\mathbb{E}_{\mathbf{y},\mathbf{a}}[\log q_{\mathrm{VLM}}(\mathbf{a} \mid \mathbf{y})]\,, \tag{14}$$

which corresponds to next-token cross-entropy over the target answer sequence.

Leakage upper bound. Equation (9) bounds the leakage term by $H(K)$. Let $p(\mathbf{k})$ denote the true probability mass function of K, and let $r(\mathbf{k})$ be any discrete variational distribution. KL non-negativity gives

$$H(K) = \mathbb{E}_{\mathbf{k}}[-\log p(\mathbf{k})] = \mathbb{E}_{\mathbf{k}}[-\log r(\mathbf{k})] - D_{\mathrm{KL}}(p\|r) \leq \mathbb{E}_{\mathbf{k}}[-\log r(\mathbf{k})]\,. \tag{15}$$

To obtain a tractable expression for $r(\mathbf{k})$, we adopt a factorized standard Gaussian as the variational reference density for pre-quantization features $\boldsymbol{\mu}$. The quantization rule in Equation (6) assigns each $[\boldsymbol{\mu}]_{t,d}$ to a bin of width $[\boldsymbol{\delta}]_d$. Integrating the reference density over these bins (Ballé et al., 2017; 2018) gives

$$r(\mathbf{k}) = \prod_{t=1}^{T}\prod_{d=1}^{D}\left[\Phi\big([\boldsymbol{\delta}]_d\left([\mathbf{k}]_{t,d} + \tfrac{1}{2}\right)\big) - \Phi\big([\boldsymbol{\delta}]_d\left([\mathbf{k}]_{t,d} - \tfrac{1}{2}\right)\big)\right], \tag{16}$$

where Φ is the standard Gaussian cumulative distribution function. The detailed derivation is provided in Appendix A.2.

Consequently, a tractable upper bound on the leakage term is

$$\begin{aligned} \mathcal{L}_{\mathrm{leak}} &= -\mathbb{E}_{\mathbf{k}}\left[\log r(\mathbf{k})\right] \\ &= -\sum_{t=1}^{T}\sum_{d=1}^{D}\mathbb{E}_{\mathbf{k}}\left[\log\left(\Phi\big([\boldsymbol{\delta}]_d\left([\mathbf{k}]_{t,d} + \tfrac{1}{2}\right)\big) - \Phi\big([\boldsymbol{\delta}]_d\left([\mathbf{k}]_{t,d} - \tfrac{1}{2}\right)\big)\right)\right]. \end{aligned} \tag{17}$$

Note that $\mathcal{L}_{\text{leak}}$ also yields the lower bound $\mathcal{D}_X(\mathcal{L}_{\text{leak}})$ on expected reconstruction MSE, as shown in Appendix A.1.

Combining the utility and leakage bounds gives

$$-I(Y;A) + \beta I(X;Y) \leq \mathcal{L}_{\text{util}} - H(A) + \beta \mathcal{L}_{\text{leak}}. \tag{18}$$

Since $H(A)$ is fixed by the data distribution, minimizing this upper bound is equivalent to minimizing the following loss function:

$$\mathcal{L} = \mathcal{L}_{\text{util}} + \beta \mathcal{L}_{\text{leak}}. \tag{19}$$

This objective trains the bottleneck to maintain downstream utility while limiting image information leakage, with β controlling the trade-off between them.

5 Experiments

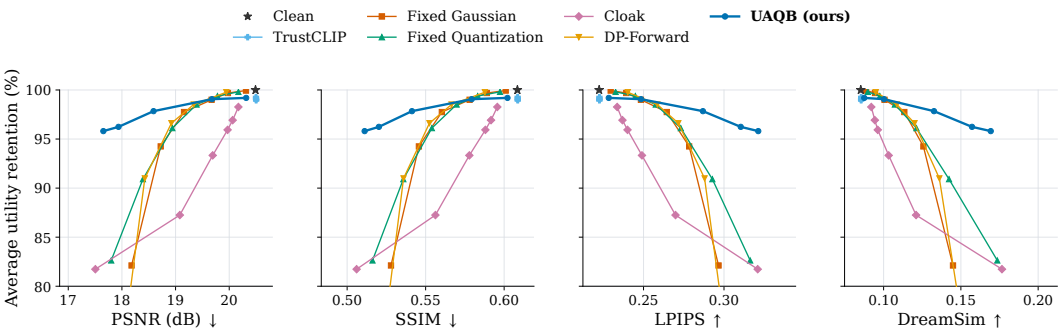
5.1 Experimental Setup

Models and data. We use LLaVA-1.5-7B (Liu et al., 2024a) and Qwen3.5-2B (Qwen Team, 2026) as the backbones, with thinking disabled for Qwen. Input images are square-padded and resized to 336×336 and 384×384, respectively. We place the split point before the multimodal projector in LLaVA and before spatial merging and projection in Qwen, with 576×1024 visual features transmitted across the split in both backbones. The vision encoder preceding the split point runs on the device, while the remaining VLM components run on the server, with all parameters frozen. The defenses are placed after the vision encoder on the device before transmission, and we train the learnable defenses on the LLaVA-665K dataset, retaining full multi-turn context. In UAQB, the quantizer and utility adapter each use a single Transformer block. A fixed set of 2,048 images is reserved for validation and reconstruction evaluation and excluded from both defense and attacker training.

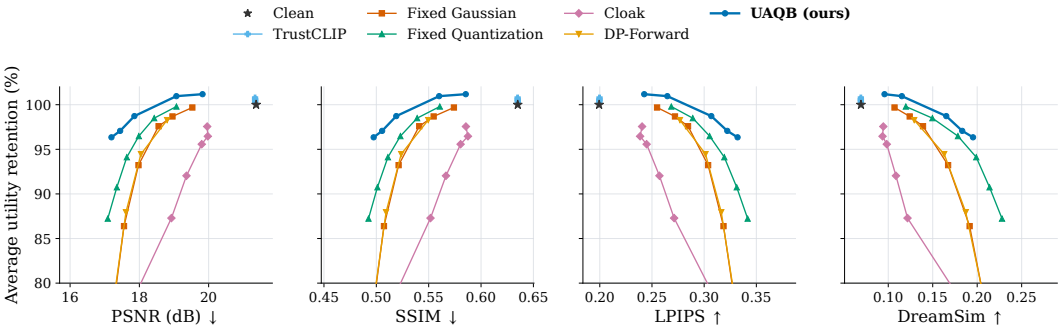
Baselines. We compare UAQB with five defense baselines that can be directly applied or adapted to the frozen VLM visual-token interface: (1) fixed Gaussian noise; (2) fixed uniform quantization; (3) Cloak, a learned-noise defense (Mireshghallah et al., 2021); (4) DP-Forward, a differentially private perturbation method for language models (Du et al., 2023); and (5) TrustCLIP, a VLM-specific adversarial learning defense (Athanasiou et al., 2026). We also include the unprotected VLM (Clean). All defenses operate at the same interface for each backbone. Baseline adaptations to the frozen VLM setting are detailed in Appendix B.1.

Utility evaluation. We evaluate utility with LMMs-Eval (Zhang et al., 2025) on six VLM benchmarks: MME (Fu et al., 2025), POPE (Li et al., 2023a), GQA (Hudson & Manning, 2019), TextVQA (Singh et al., 2019), MMBench EN-Dev v1.1 (Liu et al., 2024b), and VQAv2 Test-Dev (Goyal et al., 2017). We report the MME perception score, POPE F1, and accuracy for the remaining benchmarks. Overall utility is reported as average retention, computed as $\frac{100}{6}\sum_{j=1}^{6} s_j / s_j^{\text{clean}}$, where s_j and s_j^{clean} denote the protected and Clean scores on the j-th benchmark, respectively.

Reconstruction evaluation. Under the white-box threat model in Section 3, we adopt direct Transformer-based feature inversion, currently the most effective approach for reconstructing images from deep visual tokens (Allakhverdov et al., 2026; Zheng et al., 2026). Our reconstructor is a 12-layer Vision Transformer-style model. For each backbone, we first train a clean reconstructor on auxiliary images from the LLaVA-665K training split. For every evaluated defense checkpoint, we independently fine-tune all reconstructor parameters from the clean checkpoint on its transmitted representations to ensure a strong adaptive attack. The attacker observes only the transmitted visual tokens and is trained with an ℓ_1 plus learned perceptual image patch similarity (LPIPS) loss. Metrics are computed on the original image content excluding padding. We report peak signal-to-noise ratio (PSNR), structural similarity (SSIM) (Wang et al., 2004), LPIPS (Zhang et al., 2018), and DreamSim (Fu et al., 2023). Lower PSNR/SSIM and higher LPIPS/DreamSim indicate stronger reconstruction privacy.



(a) LLaVA-1.5-7B



(b) Qwen3.5-2B

Figure 2: Average utility retention vs. reconstruction quality.

Implementation details. We train UAQB using the objective in Equation (19) with different values of β to explore the utility-privacy trade-off. Training runs for one epoch with AdamW, an effective batch size of 128, and a peak learning rate of 10^{-4} followed by cosine decay to zero. We use hard rounding in the forward pass with a straight-through estimator (STE) for backpropagation. All results use the final defense and reconstructor checkpoints. Further experimental details are provided in Appendix B.2. The source code is included in the submitted supplementary material.

Inference profiling shows that UAQB increases the number of floating-point operations (FLOPs) by only 0.39% on LLaVA and 4.03% on Qwen, with corresponding increases in time to first token (TTFT) of 2.51% and 1.80%. Detailed profiling setup and results are provided in Appendix B.3.

5.2 Utility-Privacy Trade-off

Figure 2 compares average utility retention with reconstruction quality measured by PSNR, SSIM, LPIPS, and DreamSim on LLaVA-1.5-7B and Qwen3.5-2B under adaptive attacks. Perturbation-based defenses, including fixed Gaussian noise, fixed quantization, Cloak, and DP-Forward, incur substantial utility degradation as protection increases. In contrast, UAQB maintains considerably higher utility at stronger protection levels. On LLaVA, UAQB at $\beta = 1$ retains 95.8% average utility with an LPIPS of 0.321, whereas fixed quantization at $\delta = 3$ retains only 82.6% at a similar LPIPS of 0.316. On Qwen, UAQB at $\beta = 1$ achieves stronger protection across all four reconstruction metrics than fixed quantization at $\delta = 1.5$ at nearly identical utility retention (96.3% versus 96.5%), including an LPIPS of 0.332 versus 0.305.

Adversarially learned TrustCLIP exhibits a different limitation. On both backbones, it maintains high utility, but reconstruction quality under adaptive attacks remains close to the Clean level. On LLaVA, this behavior persists even when the server-side VLM is jointly trained as prescribed by TrustCLIP (Table 4 in Appendix C.1), illustrating that adversarially learned defenses can hinder the selected training adversary while retaining substantial information for image reconstruction.

UAQB instead jointly optimizes an explicit leakage upper bound and downstream utility, coupling information-limiting quantization with utility adaptation without relying on a reconstruction adver-

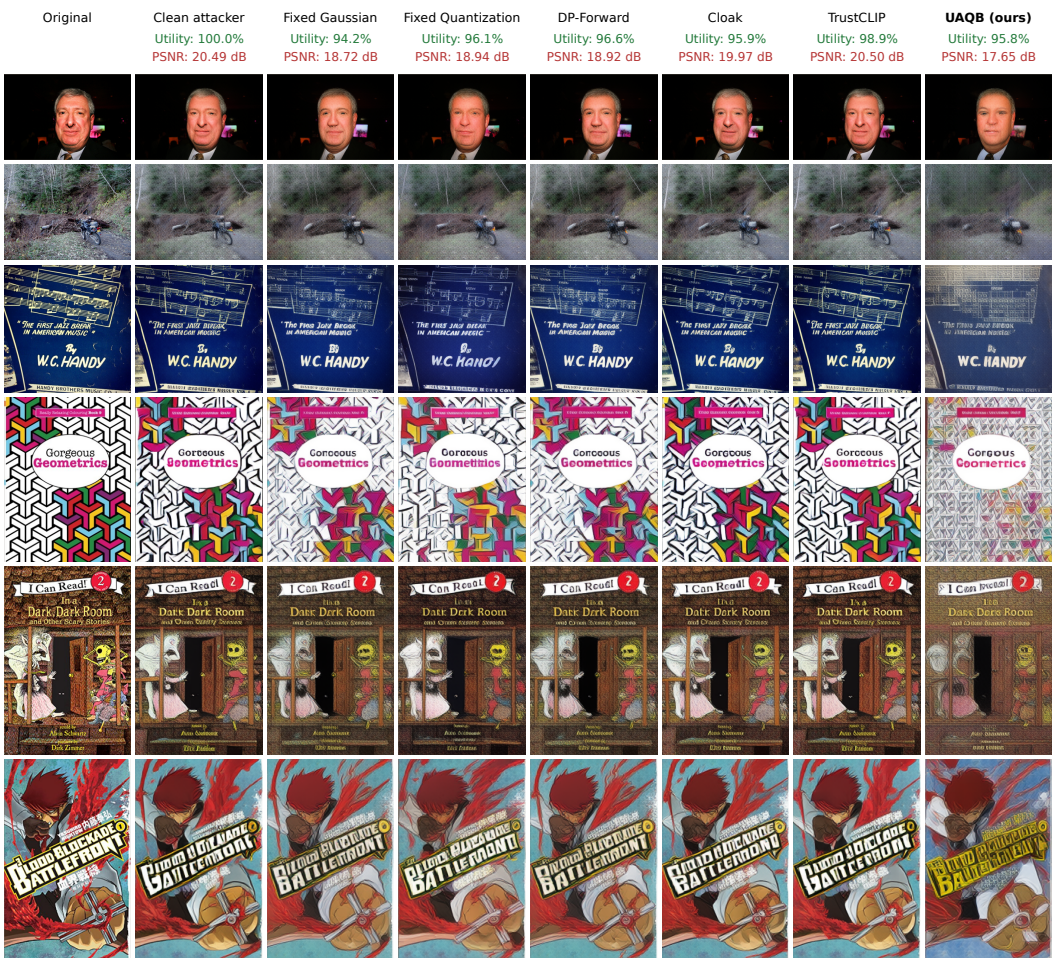


Figure 3: Attacker reconstructions on LLaVA-1.5-7B. Headers report average utility retention and validation-set PSNR.

sary. This enables stronger reconstruction protection while maintaining high utility. Full benchmark scores and reconstruction metrics are reported in Appendix C.1.

5.3 Attacker Reconstructions

Figure 3 presents reconstructions of validation images at comparable average utility retention on LLaVA-1.5-7B. Compared with the baselines, UAQB more severely alters facial features in the portrait, corrupts small lettering and musical notation, and disrupts the intricate geometric patterns on the book cover. While coarse visual content remains recognizable, these instance-specific details are less faithfully recovered.

5.4 Ablation Studies

Figure 4 compares UAQB with *Learned Quantizer Only* and *Utility Adapter Only* (with fixed quantization) on LLaVA-1.5-7B. The full model achieves a more favorable utility-privacy trade-off, particularly at stronger protection levels. At similar reconstruction privacy (LPIPS around 0.31), UAQB retains 96.24% average utility, compared with 94.67% for the learned quantizer alone and 93.62% for the utility adapter with fixed quantization. This improvement is consistent with the complementary roles of the two modules. The learned quantizer exploits context across visual tokens to learn utility-aware bin assignments, helping preserve useful information under quantization. The utility adapter restores compatibility with the frozen downstream VLM, enabling the jointly trained

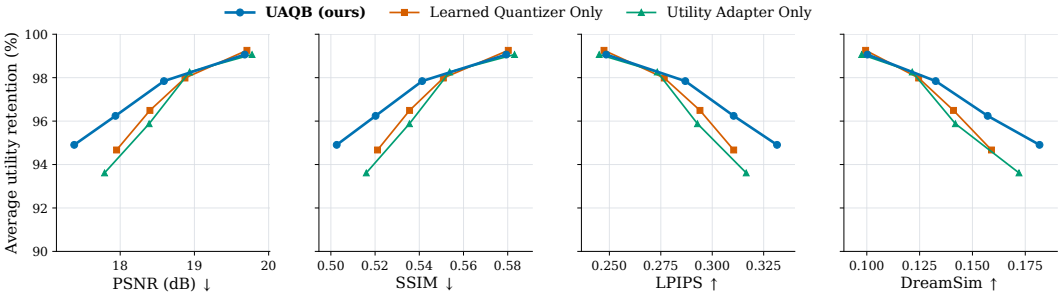


Figure 4: Utility-privacy trade-offs of the component ablations on LLaVA-1.5-7B.

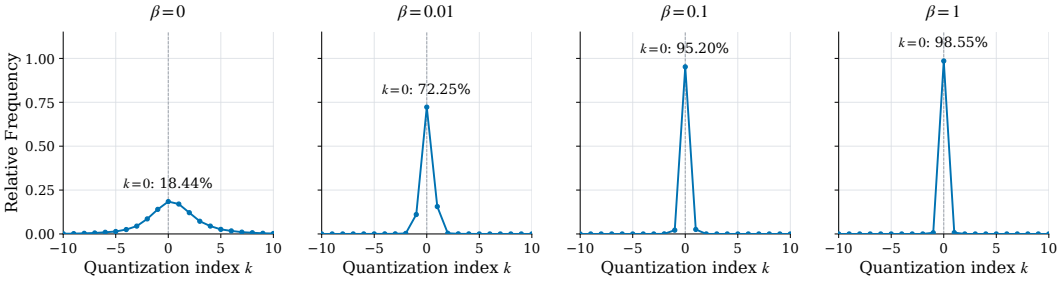


Figure 5: Empirical distribution of scalar quantization indices, pooled over all tokens and feature dimensions of validation images on LLaVA-1.5-7B.

bottleneck to accommodate coarser quantization while maintaining utility. Together, the two modules support stronger reconstruction protection at high utility. Full numerical results are provided in Appendix C.2.

5.5 Quantization-Index Analysis

Figure 5 presents the distribution of quantization indices $\mathbf{k}$ on LLaVA-1.5-7B. Increasing β concentrates the indices around zero, producing a highly sparse quantized representation. At $\beta = 0.1$, 95.20% of scalar entries are quantized to zero while UAQB retains 97.8% average utility. At $\beta = 1$, the zero fraction further increases to 98.55%, yet average utility retention remains at 95.8% (Table 2). These results show that the jointly learned bottleneck preserves sufficient task-relevant information for downstream inference even when nearly all scalar entries are quantized to zero to limit image information leakage.

6 Conclusion

We proposed UAQB, a plug-in defense that couples learned quantization with utility adaptation for private split VLM inference. We derived a tractable upper bound on the information bottleneck objective, enabling joint optimization of privacy and utility without a reconstruction adversary or updates to pretrained VLM components. The utility adapter restores compatibility with the frozen downstream VLM while preserving the quantizer's information bound. Experiments on LLaVA-1.5-7B and Qwen3.5-2B across six benchmarks demonstrate a more favorable utility-privacy trade-off than the compared defenses under adaptive reconstruction attacks, particularly at stronger protection levels. On LLaVA, UAQB retains 95.8% average utility even when 98.55% of quantized scalar entries are zero. These findings demonstrate that a learned information bottleneck can reduce reconstruction leakage while preserving sufficient information for downstream VLM inference.

Limitations and scope. Our evaluation focuses on image reconstruction from transmitted visual tokens. UAQB does not provide differential privacy. Protection against sensitive-attribute inference or attacks using additional side information remains to be evaluated.

AI USE STATEMENT

In this work, we used generative AI tools for providing feedback on background literature, the correctness of analysis, and coding work; we have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, or proposing or refining hypotheses; generating synthetic data sets, assisting with translation, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to assist with code implementation and refinement, sourcing and searching for information, and editing the research paper to improve readability. We have reviewed all AI-assisted work. All AI-assisted code was manually reviewed and tested, and all other AI-assisted outputs were checked and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work follows the ICLR Code of Ethics. All datasets and pretrained models used in this study are publicly available, and the study involves no new data collection or human-subject experiments. Reconstruction attacks are conducted only in controlled experiments to evaluate privacy leakage and support the development of stronger defenses; we acknowledge their dual-use nature. The reported protection holds only under the stated assumptions and threat model and does not constitute differential privacy or universal protection against all privacy attacks.

REPRODUCIBILITY STATEMENT

The threat model is described in Section 3, the model architecture and optimization objective in Section 4, and the evaluation protocol in Section 5.1. Appendix A provides the assumptions and proofs supporting the information-theoretic results. Appendix B provides implementation details for UAQB, the baseline adaptations, training hyperparameters, and the adaptive reconstruction attack protocol. All datasets and pretrained models used in our experiments are publicly available. The source code is included in the submitted supplementary material.

REFERENCES

Eduard Allakhverdov, Dmitrii Tarasov, Elizaveta Goncharova, and Andrey Kuznetsov. Feature inversion as a lens on vision encoders. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3598–3605, 2026.

Nikos Athanasiou, Ilya A. Petrov, Angela Yao, Shugao Ma, Eric Sauser, Edoardo Remelli, Shreyas Hampali, Johannes Schönberger, Fadime Sener, and Bugra Tekin. TrustCLIP: Learning private visual features via adversarial reconstruction. In *European Conference on Computer Vision (ECCV)*, pp. 190–208, 2026.

Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations (ICLR)*, 2017.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations (ICLR)*, 2018.

Dake Chen, Shiduo Li, Yuke Zhang, Chenghao Li, Souvik Kundu, and Peter A. Beerel. DIA: Diffusion based inverse network attack on collaborative inference. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 124–130, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24185–24198, 2024b.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, 2nd edition, 2006.

Ruijun Deng, Zhihui Lu, and Qiang Duan. InfoDecom: Decomposing information for defending against privacy leakage in split inference. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 20737–20745, 2026.

Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4829–4837, 2016.

Minxin Du, Xiang Yue, Sherman S. M. Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. DP-Forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 2665–2679, 2023.

Lin Duan, Jingwei Sun, Jinyuan Jia, Yiran Chen, and Maria Gorlatova. Reimagining mutual information for enhanced defense against data leakage in collaborative inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 44479–44500, 2024.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. DreamSim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 50742–50768, 2023.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6904–6913, 2017.

Yujie Gu, Richeng Jin, Xiaoyu Ji, Yier Jin, and Wenyuan Xu. Differentially private and communication efficient large language model split inference via stochastic quantization and soft prompt. *arXiv preprint arXiv:2602.11513*, 2026.

Landi He, Xiaoyu Yang, and Lijian Xu. The model knows which tokens matter: Automatic token selection via noise gating. *arXiv preprint arXiv:2603.07135*, 2026.

Zecheng He, Tianwei Zhang, and Ruby B. Lee. Model inversion attacks against collaborative inference. In *Annual Computer Security Applications Conference (ACSAC)*, pp. 148–162, 2019.

Zecheng He, Tianwei Zhang, and Ruby B. Lee. Attacking and protecting data privacy in edge–cloud collaborative inference systems. *IEEE Internet of Things Journal*, 8(12):9706–9716, 2021.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6700–6709, 2019.

Wa-Kin Lei, Jun-Cheng Chen, and Shang-Tse Chen. DRAG: Data reconstruction attack using guided diffusion. In *International Conference on Machine Learning (ICML)*, pp. 33883–33901, 2025.

Ang Li, Jiayi Guo, Huanrui Yang, Flora D. Salim, and Yiran Chen. DeepObfuscator: Obfuscating intermediate representations with privacy-preserving adversarial learning on smartphones. In *IEEE/ACM International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 28–39, 2021.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 292–305, 2023a.

Zhimin Li, Pan Wang, Jingxian Chen, Yuantao Tang, Anthony Chen, Qian Lou, and Jingtong Hu. Defending wearable VLMs against private attribute inference. *arXiv preprint arXiv:2608.28691*, 2026.

Ziang Li, Mengda Yang, Yaxin Liu, Juan Wang, Hongxin Hu, Wenzhe Yi, and Xiaoyang Xu. GAN you see me? Enhanced data reconstruction attacks against split inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 54554–54566, 2023b.

Yinan Liang, Ziwei Wang, Xiuwei Xu, Jie Zhou, and Jiwen Lu. EfficientLLaVA: Generalizable auto-pruning for large vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9445–9454, 2025.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, 2024a.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision (ECCV)*, pp. 216–233, 2024b.

Zixiang Liu, Chi Chen, Shuguang Yuan, Weilong Huang, Xiaojie Zhu, and Peizhuo Lv. Privacy-preserving for user-uploaded images and text in vision–language models. *Computers & Security*, 168:104931, 2026.

Kiwan Maeng, Chuan Guo, Sanjay Kariyappa, and G. Edward Suh. Bounding the invertibility of privacy-preserving instance encoding using Fisher information. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 51904–51925, 2023.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5188–5196, 2015.

Peihua Mai, Ran Yan, Zhe Huang, Youjia Yang, and Yan Pang. Split-and-Denoise: Protect large language model inference with local differential privacy. In *International Conference on Machine Learning (ICML)*, pp. 34281–34302, 2024.

Fatemehsadat Mireshghallah, Mohammadkazem Taram, Prakash Ramrakhyani, Ali Jalali, Dean Tullsen, and Hadi Esmaeilzadeh. Shredder: Learning noise distributions to protect inference privacy. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 3–18, 2020.

Fatemehsadat Mireshghallah, Mohammadkazem Taram, Ali Jalali, Ahmed Taha Elthakeb, Dean Tullsen, and Hadi Esmaeilzadeh. Not all features are equal: Discovering essential features for preserving prediction privacy. In *The Web Conference (WWW)*, pp. 669–680, 2021.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Abhishek Singh, Ayush Chopra, Ethan Garza, Emily Zhang, Praneeth Vepakomma, Vivek Sharma, and Ramesh Raskar. DISCO: Dynamic and invariant sensitive channel obfuscation for deep neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12125–12135, 2021.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8317–8326, 2019.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 368–377, 1999.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Song Xia, Yi Yu, Wenhan Yang, Meiwen Ding, Zhuo Chen, Ling-Yu Duan, Alex C. Kot, and Xudong Jiang. Theoretical insights in model inversion robustness and conditional entropy maximization for collaborative inference systems. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8753–8763, 2025.

Mengda Yang, Ziang Li, Juan Wang, Hongxin Hu, Ao Ren, Xiaoyang Xu, and Wenzhe Yi. Measuring data reconstruction defenses in collaborative inference systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 12855–12867, 2022.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. ATP-LLaVA: Adaptive token pruning for large vision language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24972–24982, 2025.

Yupeng Yin, Xianglong Zhang, Huanle Zhang, Feng Li, Yue Yu, Xiuzhen Cheng, and Pengfei Hu. Ginver: Generative model inversion attacks against collaborative inference. In *The Web Conference (WWW)*, pp. 2122–2131, 2023.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL*, pp. 881–916, 2025.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations (ICLR)*, 2026.

A Theoretical Details and Proofs

A.1 Proof of the Reconstruction-Distortion Bound

Proof of Theorem 1. For a target image independent of the auxiliary training data, the reconstructor observes only Y under the threat model in Section 3, inducing the Markov chain $X \to Y \to \hat{X}$. Combining data processing with Equation (9) gives

$$I(X;\hat{X}) \le I(X;Y) \le H(K). \tag{20}$$

Recall that $\mathcal{D}_X(R)$ in Equation (10) characterizes the minimum expected reconstruction MSE subject to $I(X;\widetilde{X}) \le R$. Hence,

$$\begin{aligned}\mathbb{E}\left[\frac{1}{N}\|X-\hat{X}\|_2^2\right] &\ge \inf_{\substack{p(\widetilde{\mathbf{x}}|\mathbf{x}):\\ I(X;\widetilde{X})=I(X;\hat{X})}} \mathbb{E}\left[\frac{1}{N}\|X-\widetilde{X}\|_2^2\right] \\ &\overset{\text{(a)}}{\ge} \inf_{\substack{p(\widetilde{\mathbf{x}}|\mathbf{x}):\\ I(X;\widetilde{X})\le H(K)}} \mathbb{E}\left[\frac{1}{N}\|X-\widetilde{X}\|_2^2\right] \\ &= \mathcal{D}_X(H(K))\,,\end{aligned} \tag{21}$$

where (a) follows from relaxing the information constraint using Equation (20). □

Moreover, increasing R enlarges the set of admissible reconstruction distributions, so $\mathcal{D}_X(R)$ is nonincreasing. Since Equations (15) and (17) give $H(K) \le \mathcal{L}_{\text{leak}}$, we obtain

$$\mathbb{E}\left[\frac{1}{N}\|X-\hat{X}\|_2^2\right] \ge \mathcal{D}_X(H(K)) \ge \mathcal{D}_X(\mathcal{L}_{\text{leak}})\,. \tag{22}$$

Thus, the leakage term $\mathcal{L}_{\text{leak}}$ in our loss function (19) guarantees a lower bound on the attacker's reconstruction MSE.

A.2 Derivation of the Discrete Variational Distribution

For token position t and feature dimension d, define the quantization bin for index $[\mathbf{k}]_{t,d}$ as

$$B_d([\mathbf{k}]_{t,d}) = \Big[[\boldsymbol{\delta}]_d([\mathbf{k}]_{t,d}-\tfrac{1}{2}),\ [\boldsymbol{\delta}]_d([\mathbf{k}]_{t,d}+\tfrac{1}{2})\Big). \tag{23}$$

Integrating the standard Gaussian variational reference density $\varphi([\boldsymbol{\mu}]_{t,d})$ over this bin gives the corresponding probability mass:

$$r_d([\mathbf{k}]_{t,d}) = \int_{B_d([\mathbf{k}]_{t,d})} \varphi(u)\,du = \Phi\big([\boldsymbol{\delta}]_d([\mathbf{k}]_{t,d}+\tfrac{1}{2})\big) - \Phi\big([\boldsymbol{\delta}]_d([\mathbf{k}]_{t,d}-\tfrac{1}{2})\big)\,. \tag{24}$$

Since the variational reference density is factorized, the distribution over the full index tensor is

$$r(\mathbf{k}) = \prod_{t=1}^{T}\prod_{d=1}^{D} r_d([\mathbf{k}]_{t,d}), \tag{25}$$

which gives Equation (16).

Since the bins partition $\mathbb{R}$, each factor satisfies

$$\sum_{[\mathbf{k}]_{t,d}\in\mathbb{Z}} r_d([\mathbf{k}]_{t,d}) = \int_{\mathbb{R}} \varphi(u)\,du = 1. \tag{26}$$

Consequently,

$$\sum_{\mathbf{k}\in\mathbb{Z}^{T\times D}} r(\mathbf{k}) = \prod_{t=1}^{T}\prod_{d=1}^{D}\left(\sum_{[\mathbf{k}]_{t,d}\in\mathbb{Z}} r_d([\mathbf{k}]_{t,d})\right) = 1. \tag{27}$$

Thus, r is a probability mass function on $\mathbb{Z}^{T\times D}$. Substituting this distribution into Equation (15) gives the leakage loss in Equation (17).

B Experimental and Implementation Details

B.1 Baseline Implementations

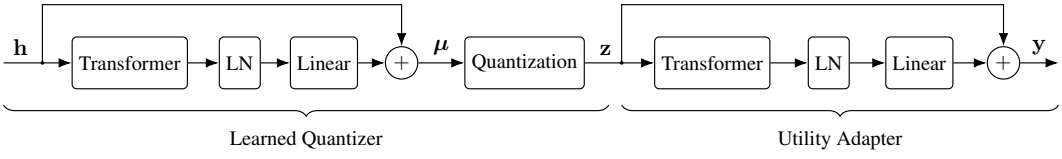
Unless otherwise specified, baseline hyperparameters and training settings follow the respective original papers. We implement the five baselines at the frozen visual-token interface as follows:

1. **Fixed Gaussian noise.** Device transmits the noisy visual tokens $[\mathbf{y}]_{t,d} = [\mathbf{h}]_{t,d} + \sigma\epsilon_{t,d}$ to server, where $\epsilon_{t,d}$ is sampled independently from $\mathcal{N}(0,1)$ for each image, token position, and feature dimension. The shared scalar σ controls the defense strength.
2. **Fixed quantization.** We apply scalar quantization as $[\mathbf{y}]_{t,d} = \delta\,\mathrm{round}([\mathbf{h}]_{t,d}/\delta)$. The scalar step size δ is shared across images, token positions, and feature dimensions and controls quantization strength.
3. **Cloak.** We use its first-stage Gaussian perturbation (Mireshghallah et al., 2021). A noise mean μ_d and standard deviation σ_d are learned for each feature dimension and shared across images and token positions, while noise is sampled independently for each component. The training objective combines answer-token cross-entropy with a variance regularizer $-\lambda^{-1}\log\Big(D^{-1}\sum_{d=1}^{D}\sigma_d^2\Big)$, which encourages larger noise variance. Smaller λ places more weight on this term and favors stronger perturbation.
4. **DP-Forward.** We adapt its clipping-and-noise mechanism (Du et al., 2023) by clipping each image's token matrix to a maximum Frobenius norm C and adding independent Gaussian noise to its components, where C is set to the median token-matrix norm. The noise standard deviation is calibrated using sensitivity $2C$ and privacy parameters ε and $\delta_{\mathrm{DP}} = 10^{-5}$. No model parameters are trained. Smaller ε yields a larger noise standard deviation and stronger protection.
5. **TrustCLIP.** A Stable Diffusion 1.5/IP-Adapter reconstructor is trained on clean visual tokens for 5,000 steps and then frozen (Athanasiou et al., 2026). A token-shared residual projection is trained on LLaVA-665K for one epoch to minimize answer-token cross-entropy and maximize the fixed reconstructor's reconstruction loss, with λ_{rec} controlling defense strength. All pretrained VLM components remain frozen.

Other information-theoretic defenses, such as InfoScissors (Duan et al., 2024) and InfoDecom (Deng et al., 2026), are designed and evaluated around CNN-based split classifiers and update split-model components. Their original implementations therefore do not operate directly at the frozen VLM visual-token interface considered here.

B.2 Experimental Details

UAQB. The proposed UAQB architecture used in our experiments is shown below, with layer normalization abbreviated as LN:

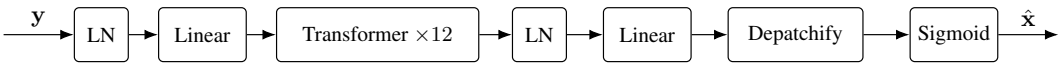


Each Transformer consists of a single block with 16 attention heads, a hidden dimension of 1024, and a feed-forward dimension of 4096. The block uses Gaussian error linear unit (GELU) activation and no dropout. The two linear heads are separate fully connected $1024 \to 1024$ projections, each shared across token positions. The channel-wise quantization step sizes $\boldsymbol{\delta}$ are shared across images and token positions, parameterized in log space to ensure positivity throughout training, with $[\boldsymbol{\delta}]_d \in [0.01, 10]$.

UAQB training largely follows the instruction-tuning settings of LLaVA-1.5 (Liu et al., 2024a), with the training details given below. At initialization, both linear heads have zero weights and biases and

$[\boldsymbol{\delta}]_d = 1$, yielding $\boldsymbol{\mu} = \mathbf{h}$ and $\mathbf{y} = \mathbf{z}$. Training therefore starts from fixed quantization, avoiding additional utility loss from untrained modules and providing a stable starting point for optimization. We train for one epoch on all 620,910 image-containing LLaVA-665K training conversations after excluding the held-out 2,048 validation images. We use full multi-turn conversations, truncating sequences at 2,048 tokens and applying cross-entropy to answer tokens. To keep the utility and leakage loss terms in Equation (19) on comparable scales, we use the per-index average $\overline{\mathcal{L}}_{\text{leak}} = \mathcal{L}_{\text{leak}}/(TD)$ in implementation. The β values in our experimental results weight this average. We use AdamW with coefficients $(0.9, 0.999)$, zero weight decay, and an effective batch size of 128. The peak base learning rate is 10^{-4}, with a 100-fold multiplier for $\boldsymbol{\delta}$ to accelerate its adaptation during the single training epoch. We use 3% linear warmup followed by cosine decay to zero. Gradients are clipped at norm 1. Trainable parameters and the leakage loss use 32-bit floating-point (FP32) precision, while the frozen VLM uses bfloat16 (BF16). We report results from the final defense checkpoint.

Reconstruction attacker. The reconstructor largely follows the decoder design and training settings in Zheng et al. (2026). The architecture used in our experiments is shown below:



The reconstructor takes a 24×24 grid of 1024-dimensional visual tokens. An input LayerNorm and token-wise fully connected $1024 \to 1024$ projection precede 12 Transformer blocks with 16 attention heads, a hidden dimension of 1024, 2D rotary positional embeddings (RoPE), query-key (QK) normalization, and a Swish-gated linear unit (SwiGLU) intermediate dimension of 2752. Dropout is disabled. After a final LayerNorm, a fully connected $1024 \to 3p^2$ head predicts one RGB patch per token, with $p = 14$ for LLaVA and $p = 16$ for Qwen. Depatchification and a sigmoid produce the reconstructed image.

For each backbone, the clean reconstructor is trained for 50 epochs using 346,986 unique images from the LLaVA-665K training split, without their associated conversations. For every evaluated defense checkpoint, we independently fine-tune all reconstructor parameters for 10 epochs from the final clean exponential moving average (EMA) checkpoint on the transmitted visual tokens, using the same training images as the clean reconstructor. Training minimizes equally weighted ℓ_1 and VGG-LPIPS losses on the original-image content after removing square padding. The content is resized to 336×336 for LLaVA and 384×384 for Qwen. LPIPS uses a 336×336 input in both cases. We use AdamW with coefficients $(0.9, 0.95)$, zero weight decay, an effective batch size of 256, and a peak learning rate of 2×10^{-4} with one epoch of linear warmup followed by cosine decay to 10% of the peak. The EMA decay is 0.999. Reconstructor parameters and losses use FP32 with BF16 autocast for the forward pass. Crop coordinates are used only for supervision and scoring. The attacker receives only visual tokens. Reconstruction metrics are computed from the final attacker checkpoint and averaged over the same 2,048 validation images.

B.3 INFERENCE COST

Table 1: On-device and full-model inference cost with and without UAQB. Full-model latency is TTFT. Parentheses show percentage increases relative to Clean within each measurement scope.

Backbone	Scope	Setting	Params (M)		FLOPs (G)		Latency (ms)	
LLaVA-1.5-7B	On-device	Clean	303.51		381.92		7.855	
		+UAQB	330.80	(+8.99%)	416.04	(+8.94%)	8.703	(+10.79%)
	Full model	Clean	7063.43		8678.93		35.183	
		+UAQB	7090.72	(+0.39%)	8713.06	(+0.39%)	36.068	(+2.51%)
Qwen3.5-2B	On-device	Clean	306.24		382.32		12.952	
		+UAQB	333.54	(+8.91%)	416.44	(+8.93%)	13.838	(+6.84%)
	Full model	Clean	2213.24		846.89		51.663	
		+UAQB	2240.54	(+1.23%)	881.02	(+4.03%)	52.590	(+1.80%)

We profile Clean and UAQB ($\beta = 0.1$) on an NVIDIA H100 80GB GPU in BF16 with batch size 1, using the same 2,048 held-out validation images and the prompt "Describe the image." Latencies are medians over these images after 50 warm-up runs.

On-device latency covers vision encoding and UAQB, and full-model latency is TTFT from GPU-resident inputs. Timing excludes preprocessing and data transfer. FLOP counts cover convolutions and matrix products, including attention, with two operations per multiply-accumulate.

As shown in Table 1, UAQB adds 27.30M parameters and 34.12 GFLOPs per image on both backbones, increasing TTFT by 2.51% on LLaVA and 1.80% on Qwen.

C Additional Results

C.1 Full Utility and Privacy Results

Table 2: Utility and privacy results for the points in Figure 2 on LLaVA-1.5-7B. MME perception (MME-P) is reported as a raw score, and other individual utility scores are percentages. Avg. is the mean utility retention (%) relative to Clean across six benchmarks. PSNR is in dB. Privacy metrics use the matched adaptive reconstruction attacks described in Section 5.1.

Method	Parameter	Utility ↑							Privacy			
		MME-P	POPE F1	GQA	TextVQA	MMBench	VQAv2	Avg.	PSNR ↓	SSIM ↓	LPIPS ↑	DreamSim ↑
Clean	–	1511.12	85.48	61.93	57.90	61.69	78.39	100.00%	20.49	0.6084	0.2224	0.0852
TrustCLIP	$\lambda_{\text{rec}} = 0.001$	1466.22	85.14	61.62	56.76	61.61	78.09	98.94%	20.50	0.6087	0.2226	0.0855
	$\lambda_{\text{rec}} = 0.01$	1468.96	85.44	61.78	56.71	61.46	78.10	99.02%	20.50	0.6087	0.2226	0.0854
	$\lambda_{\text{rec}} = 0.1$	1479.64	85.88	61.74	56.34	61.53	78.07	99.12%	20.50	0.6087	0.2226	0.0856
	$\lambda_{\text{rec}} = 1$	1505.89	85.15	61.66	56.56	61.53	77.94	99.28%	20.50	0.6086	0.2226	0.0855
Fixed Gaussian	$\sigma = 0.1$	1507.26	85.44	61.82	57.85	61.84	78.35	99.94%	20.31	0.6010	0.2296	0.0887
	$\sigma = 0.2$	1483.86	85.63	61.59	57.90	62.00	78.26	99.69%	19.98	0.5890	0.2393	0.0944
	$\sigma = 0.3$	1454.59	85.80	61.40	57.39	61.46	78.00	99.00%	19.67	0.5780	0.2484	0.1006
	$\sigma = 0.5$	1442.56	86.01	60.75	56.23	59.98	76.80	97.75%	19.16	0.5603	0.2644	0.1134
	$\sigma = 0.7$	1350.33	85.36	59.24	54.05	57.12	74.21	94.25%	18.72	0.5457	0.2784	0.1257
	$\sigma = 1$	1143.67	69.92	54.70	49.02	47.21	67.17	82.12%	18.18	0.5279	0.2966	0.1449
Fixed Quantization	$\delta = 0.5$	1497.89	85.52	61.58	57.80	62.00	78.32	99.81%	20.17	0.5974	0.2326	0.0898
	$\delta = 1$	1479.20	85.87	61.46	57.45	61.69	77.99	99.38%	19.78	0.5832	0.2449	0.0976
	$\delta = 1.5$	1455.41	86.05	61.19	56.98	60.60	77.36	98.52%	19.39	0.5697	0.2576	0.1073
	$\delta = 2$	1393.36	84.78	60.36	55.08	59.13	75.92	96.12%	18.94	0.5540	0.2732	0.1212
	$\delta = 2.5$	1305.03	80.19	58.09	53.16	53.87	72.40	90.91%	18.39	0.5358	0.2928	0.1423
	$\delta = 3$	1134.55	71.55	53.81	50.22	49.38	65.31	82.63%	17.80	0.5162	0.3163	0.1736
Cloak	$\lambda = 1$	1144.33	79.41	52.98	47.27	44.20	65.03	81.74%	17.51	0.5060	0.3209	0.1766
	$\lambda = 5$	1227.30	82.57	56.10	49.74	49.85	69.30	87.25%	19.08	0.5562	0.2699	0.1210
	$\lambda = 10$	1363.34	84.38	59.08	52.60	56.04	73.68	93.34%	19.69	0.5778	0.2490	0.1033
	$\lambda = 20$	1422.74	84.83	59.61	54.74	58.44	75.80	95.94%	19.97	0.5881	0.2400	0.0962
	$\lambda = 30$	1454.47	84.83	60.14	55.27	59.21	76.43	96.92%	20.06	0.5916	0.2369	0.0943
	$\lambda = 50$	1485.53	85.13	60.78	55.96	60.84	77.06	98.27%	20.17	0.5957	0.2336	0.0920
DP-Forward	$\varepsilon = 2^{25}$	1490.15	85.60	61.58	57.76	62.23	78.23	99.77%	19.95	0.5877	0.2402	0.0949
	$\varepsilon = 2^{23}$	1435.81	86.12	61.17	56.94	61.30	77.49	98.52%	19.34	0.5665	0.2586	0.1086
	$\varepsilon = 2^{22}$	1399.73	85.88	60.42	55.73	59.13	75.97	96.61%	18.92	0.5524	0.2716	0.1200
	$\varepsilon = 2^{21}$	1285.75	84.01	58.05	52.25	53.41	72.13	90.99%	18.43	0.5359	0.2879	0.1361
	$\varepsilon = 2^{20}$	802.34	37.10	50.87	45.50	39.55	59.90	66.29%	17.84	0.5167	0.3082	0.1608
	$\varepsilon = 2^{19}$	519.51	0.00	40.96	39.48	21.67	43.70	43.26%	17.14	0.4939	0.3344	0.2019
UAQB (ours)	$\beta = 0$	1475.01	85.17	61.40	57.05	61.92	78.28	99.19%	20.32	0.6021	0.2284	0.0875
	$\beta = 0.01$	1459.96	85.75	61.40	57.12	61.76	78.02	99.06%	19.68	0.5793	0.2485	0.1003
	$\beta = 0.1$	1448.64	85.13	60.62	55.64	61.07	77.31	97.84%	18.59	0.5413	0.2869	0.1326
	$\beta = 0.5$	1416.21	84.74	60.04	54.62	58.98	76.59	96.24%	17.94	0.5202	0.3104	0.1572
	$\beta = 1$	1408.33	85.27	60.11	54.81	57.43	76.10	95.81%	17.65	0.5111	0.3213	0.1693

Table 3: Utility and privacy results on Qwen3.5-2B corresponding to Figure 2. Metric definitions follow Table 2.

Method	Parameter	Utility ↑							Privacy			
		MME-P	POPE F1	GQA	TextVQA	MMBench	VQAv2	Avg.	PSNR ↓	SSIM ↓	LPIPS ↑	DreamSim ↑
Clean	–	1485.51	89.37	56.85	65.25	72.99	78.37	100.00%	21.37	0.6350	0.1994	0.0693
TrustCLIP	$\lambda_{\text{rec}} = 0.001$	1506.77	88.68	58.01	63.43	75.23	78.59	100.55%	21.34	0.6351	0.1996	0.0691
	$\lambda_{\text{rec}} = 0.01$	1499.58	88.39	57.89	63.31	75.08	78.54	100.30%	21.35	0.6351	0.1996	0.0692
	$\lambda_{\text{rec}} = 0.1$	1498.09	88.58	58.07	63.43	75.70	78.52	100.54%	21.34	0.6350	0.1997	0.0693
	$\lambda_{\text{rec}} = 1$	1514.25	88.22	58.01	63.54	76.32	78.43	100.78%	21.34	0.6349	0.1997	0.0692
Fixed Gaussian	$\sigma = 0.1$	1474.94	89.05	56.81	65.27	72.52	78.26	99.69%	19.53	0.5741	0.2549	0.1071
	$\sigma = 0.2$	1453.59	88.66	56.69	64.25	71.05	77.98	98.68%	18.96	0.5547	0.2720	0.1241
	$\sigma = 0.3$	1437.40	88.35	56.33	62.60	69.97	77.60	97.59%	18.56	0.5410	0.2845	0.1391
	$\sigma = 0.5$	1395.96	87.00	55.53	55.40	64.55	76.01	93.23%	17.97	0.5213	0.3038	0.1673
	$\sigma = 0.7$	1262.88	83.61	53.60	46.34	58.98	73.39	86.39%	17.56	0.5073	0.3185	0.1915
	$\sigma = 1$	993.28	66.83	48.24	38.09	48.68	65.89	72.61%	17.09	0.4919	0.3361	0.2186
Fixed Quantization	$\delta = 0.5$	1484.62	89.23	56.78	65.08	72.60	78.28	99.80%	19.07	0.5606	0.2685	0.1198
	$\delta = 1$	1475.87	88.88	56.78	63.12	70.05	78.02	98.49%	18.43	0.5390	0.2891	0.1496
	$\delta = 1.5$	1436.50	88.57	56.40	59.23	68.58	77.68	96.48%	17.98	0.5230	0.3052	0.1784
	$\delta = 2$	1386.18	88.21	56.03	54.54	67.34	76.99	94.11%	17.64	0.5111	0.3193	0.1990
	$\delta = 2.5$	1308.25	87.33	55.91	48.83	64.63	75.97	90.74%	17.35	0.5011	0.3312	0.2139
	$\delta = 3$	1246.51	86.19	55.03	44.75	60.29	74.48	87.23%	17.09	0.4925	0.3417	0.2279
Cloak	$\lambda = 1$	1023.41	76.41	42.80	35.34	35.45	56.52	67.42%	16.52	0.4739	0.3579	0.2508
	$\lambda = 5$	1305.37	83.79	55.22	45.82	60.06	72.47	87.29%	18.93	0.5517	0.2712	0.1217
	$\lambda = 10$	1368.88	85.98	57.04	51.99	64.40	74.91	92.03%	19.36	0.5665	0.2571	0.1087
	$\lambda = 20$	1473.03	86.75	58.10	54.78	68.19	76.50	95.57%	19.80	0.5805	0.2449	0.0983
	$\lambda = 30$	1484.56	86.99	58.24	56.05	69.43	76.81	96.46%	19.98	0.5875	0.2386	0.0936
	$\lambda = 50$	1517.84	86.70	58.46	57.42	70.59	77.26	97.56%	19.96	0.5856	0.2405	0.0944
DP-Forward	$\varepsilon = 2^{22}$	1453.92	88.56	56.45	63.81	70.20	77.90	98.27%	18.80	0.5495	0.2768	0.1292
	$\varepsilon = 2^{20}$	1413.56	87.32	55.64	57.28	66.33	76.29	94.46%	18.05	0.5237	0.3013	0.1629
	$\varepsilon = 2^{19}$	1289.69	84.41	54.01	48.39	60.45	74.00	87.95%	17.61	0.5093	0.3164	0.1872
	$\varepsilon = 2^{18}$	1043.97	71.34	49.55	39.57	49.69	67.76	75.41%	17.16	0.4944	0.3331	0.2137
	$\varepsilon = 2^{17}$	723.49	41.83	41.25	35.25	30.50	53.91	55.45%	16.67	0.4789	0.3517	0.2420
	$\varepsilon = 2^{16}$	587.15	14.16	34.66	33.87	15.17	42.25	40.49%	16.17	0.4629	0.3721	0.2718
UAQB (ours)	$\beta = 0$	1507.06	88.63	58.48	64.55	75.39	79.44	101.18%	19.82	0.5854	0.2426	0.0958
	$\beta = 0.01$	1512.95	88.87	58.15	63.71	75.39	79.35	100.96%	19.07	0.5600	0.2647	0.1153
	$\beta = 0.1$	1472.37	88.21	57.43	61.10	72.99	78.27	98.72%	17.86	0.5191	0.3068	0.1652
	$\beta = 0.5$	1451.00	87.82	56.37	59.25	71.44	77.28	97.07%	17.45	0.5055	0.3222	0.1832
	$\beta = 1$	1476.19	87.27	56.24	57.06	70.82	76.51	96.35%	17.19	0.4974	0.3320	0.1953

Table 4: Utility and reconstruction results for TrustCLIP with joint server-side training on LLaVA-1.5-7B ($\lambda_{\text{rec}} = 0.1$). Utility definitions follow Table 2.

Setting	Utility ↑							Reconstructor	Privacy			
	MME-P	POPE F1	GQA	TextVQA	MMBench	VQAv2	Avg.		PSNR ↓	SSIM ↓	LPIPS ↑	DreamSim ↑
Clean	1511.12	85.48	61.93	57.90	61.69	78.39	100.00%	IP-Adapter	10.12	0.3001	0.7692	0.6474
								Decoder (ours)	20.49	0.6084	0.2224	0.0852
TrustCLIP	1458.51	85.70	61.27	56.91	62.23	78.20	99.11%	IP-Adapter	8.13	0.0228	1.7488	0.9044
								Fine-tuned IP-Adapter	10.21	0.3333	0.7338	0.5664
								Decoder (ours)	20.50	0.6085	0.2226	0.0855

TrustCLIP with joint server-side training. We additionally evaluate TrustCLIP with its full training setup (Athanasiou et al., 2026), in which the server-side VLM is jointly trained with the defense module. Reconstruction is evaluated using both its diffusion-based training adversary (IP-Adapter) and our single-pass decoder, as reported in Table 4.

Joint training retains 99.11% average utility while reducing the training adversary's PSNR from 10.12 dB (comparable to the clean baseline reported in TrustCLIP) to 8.13 dB. However, both the adapted training adversary and our evaluation reconstructor achieve reconstruction quality close to their respective Clean references, indicating that substantial image information remains recoverable.

C.2 Component Ablation Results

Table 5: Component ablation results on LLaVA-1.5-7B, with Clean included for reference. MME-P uses its raw score, and other individual utility scores are percentages. Avg. is the six-benchmark mean utility retention (%) relative to Clean. PSNR is in dB.

Method	Parameter	Utility ↑							Privacy			
		MME-P	POPE F1	GQA	TextVQA	MMBench	VQAv2	Avg.	PSNR ↓	SSIM ↓	LPIPS ↑	DreamSim ↑
Clean	–	1511.12	85.48	61.93	57.90	61.69	78.39	100.00%	20.49	0.6084	0.2224	0.0852
UAQB (ours)	$\beta = 0.01$	1459.96	85.75	61.40	57.12	61.76	78.02	99.06%	19.68	0.5793	0.2485	0.1003
	$\beta = 0.1$	1448.64	85.13	60.62	55.64	61.07	77.31	97.84%	18.59	0.5413	0.2869	0.1326
	$\beta = 0.5$	1416.21	84.74	60.04	54.62	58.98	76.59	96.24%	17.94	0.5202	0.3104	0.1572
	$\beta = 2$	1354.69	85.16	59.92	54.56	57.12	75.71	94.91%	17.38	0.5026	0.3315	0.1817
Learned Quantizer Only	$\beta = 0.01$	1481.91	85.64	61.46	56.77	62.00	78.01	99.26%	19.71	0.5803	0.2474	0.0994
	$\beta = 0.1$	1459.82	85.10	61.04	55.83	60.60	77.23	97.98%	18.87	0.5509	0.2767	0.1245
	$\beta = 0.5$	1394.75	85.11	60.72	55.36	59.21	76.40	96.49%	18.40	0.5356	0.2940	0.1410
	$\beta = 2$	1358.95	83.84	59.46	54.54	57.74	75.41	94.67%	17.95	0.5211	0.3104	0.1590
Utility Adapter Only	$\delta = 1.0$	1485.17	85.53	61.36	56.70	61.30	78.13	99.06%	19.78	0.5831	0.2451	0.0976
	$\delta = 2.0$	1465.41	85.78	60.80	55.72	61.22	77.29	98.26%	18.94	0.5537	0.2733	0.1215
	$\delta = 2.5$	1396.75	84.84	59.93	55.04	58.44	76.07	95.88%	18.39	0.5355	0.2928	0.1419
	$\delta = 3.0$	1387.22	84.26	58.64	52.58	56.35	74.05	93.62%	17.79	0.5160	0.3165	0.1720