TASK-AWARE REPRESENTATION RELEASE WITH LOCAL DIFFERENTIAL PRIVACY FOR MULTIMODAL VLM INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Edge–cloud collaborative inference reduces on-device computation by sending locally encoded image and text tokens to a cloud-hosted vision-language model (VLM) backbone, but these uploads may reveal private input to an untrusted server. We seek to protect private images and contextual text jointly with local differential privacy (LDP), which randomizes the upload on the client without trusting the server. However, this perturbation can damage evidence needed for answering, and directly adding noise to high-dimensional representations can introduce substantial distortion. Preserving utility under LDP therefore presents three coupled challenges: **(i)** reducing the dimensionality of the representation to be perturbed while retaining task-relevant evidence; **(ii)** configuring perturbations for differently scaled visual and textual representations; and **(iii)** keeping perturbed representations usable by an unchanged cloud backbone. We address these challenges through a task-trained client-side compression–perturbation–decoding framework. It combines public-question-guided attention, a joint ellipsoidal Gaussian mechanism with calibrated modality scales, and noise-aware reconstruction of the frozen backbone's native token interface. The complete image–text upload satisfies (ϵ, δ)-LDP conditional on the public question and declared token structure. Only the client-side release modules are trained; the pretrained encoders and cloud backbone remain frozen. With InternVL3.5-8B, our framework improves ScienceQA accuracy by 12.2–23.5 percentage points and keyword exact match on a single-image WebQA subset by 13.8–39.2 points, compared with direct LDP perturbation of uncompressed representations at the three primary matched privacy budgets. Additional experiments show gains with a different VLM and reduced content recovery relative to unprotected uploads under the evaluated inversion attacks.

Keywords: Vision-Language Models; Local Differential Privacy; Task-Aware Representation Release; Ellipsoidal Gaussian Mechanism.

1 INTRODUCTION

Edge-cloud collaborative inference distributes neural-network computation between devices and servers through intermediate representation transmission, with demonstrated benefits in inference latency and device energy consumption (Kang et al., 2017). Recent systems apply this principle to multimodal models by separating modality encoders from downstream task modules (Yoon et al., 2025), including VLM deployments that perform visual encoding at the edge and language-model inference in the cloud (Li et al., 2025). We consider inference with frozen pretrained encoders and a frozen cloud backbone, using private images and textual context to answer a public question.

However, uploaded representations can reveal private inputs to an untrusted server. Text embedding inversion (Morris et al., 2023) and visual feature reconstruction attacks (Li et al., 2023; Liu et al., 2026a) can recover source content from the representations. We adopt local differential privacy (LDP) to protect inputs without trusting the server. Our protection target is the complete private image–text request, allowing both modalities to change simultaneously while the public question and declared structure remain fixed.

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.

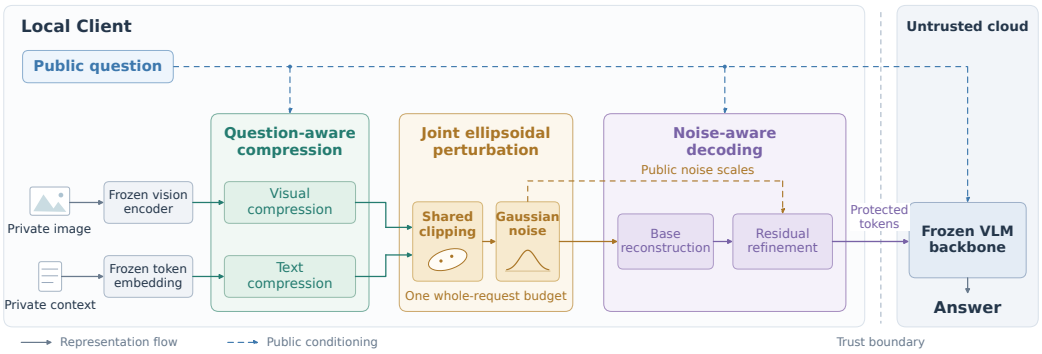


Figure 1: **Overview of our private VLM inference framework.** Compression, perturbation, and decoding run locally; the cloud receives privatized token representations and public conditions for inference with its frozen backbone.

Prior work develops private representation release and task-aware encoding and decoding (Du et al., 2023; Cheng et al., 2022). However, applying these ideas to complete image–text uploads requires coordinating dimensionality reduction, perturbation geometry, and reconstruction for a frozen VLM.

First, private release must balance dimensionality reduction with the preservation of task-relevant information. Native VLM uploads contain many token coordinates, increasing aggregate Gaussian noise energy at a fixed per-coordinate variance. Compression reduces this dimensional burden (Du et al., 2023), but the retained evidence must still support answering after perturbation.

Second, perturbations must accommodate the different scales of visual and textual representations. When two modalities have very different representation scales, the same absolute perturbation can affect one much more than the other. The design question is therefore how to configure clipping and noise across modalities according to downstream performance under a common privacy constraint.

Third, restoring token dimensions does not ensure that the frozen backbone can use perturbed evidence. Building on noise-aware learning and task-aware decoding (Du et al., 2023; Cheng et al., 2022), we seek task-effective reconstruction without adapting the cloud model.

We address these challenges by designing the complete client-side release process for downstream inference through an unchanged VLM. Our client-side framework (Figure 1) combines question-guided compression, joint ellipsoidal perturbation, and task-trained, noise-aware decoding to retain useful evidence in a compact private latent and restore the frozen backbone's token interface.

Our contributions are threefold. **(1) A task-aware private release framework for frozen VLMs.** We combine question-guided compression to retain task evidence at reduced dimensionality, joint ellipsoidal perturbation to accommodate modality differences, and noise-aware local decoding to restore backbone-compatible tokens. Both compression and decoding are trained for downstream utility. **(2) Whole-request privacy and release-geometry analysis.** We establish LDP for the complete private image–text upload and compare additive noise costs of joint ellipsoidal, isotropic, and modality-separated Gaussian releases under matched conditions. **(3) Improved privacy–utility tradeoffs across VLM configurations.** Across ScienceQA and a single-image WebQA subset, and two backbones, we demonstrate gains over direct and PCA-based LDP releases and quantify the contributions of learned compression and decoding.

2 Related Work

Representation-level LDP and task-aware release. LDP research studies both private representation release and learning representations to preserve downstream utility. DP-Forward establishes sequence-level LDP and considers optional learned maps for dimensionality reduction before perturbation and restoration afterward (Du et al., 2023). DP-BART and DP-Prompt produce private rewrites through perturbed encoder representations and calibrated language-model sampling, respectively (Igamberdiev & Habernal, 2023; Utpala et al., 2023). Cheng et al. (2022) learn an encoder–

decoder using downstream loss after latent perturbation. Split-and-Denoise uses token-level metric privacy and client-side recovery with realized perturbations after cloud computation (Mai et al., 2024). We design and evaluate compact latent privatization and task-trained local reconstruction for complete image–text uploads into a frozen VLM.

Structured and multimodal private release. Prior work adapts privacy perturbations to representation geometry and multimodal structure. MVG introduces directional Gaussian noise, while DP-Forward studies analytic matrix Gaussian calibration (Chanyaswad et al., 2018; Du et al., 2023). For multimodal inputs, Cai et al. (2022) privatize fused representations, while Struct-DP configures dimension-wise Gaussian noise using task-importance and privacy-attribute scores (Gao et al., 2026). We study modality-scaled joint clipping and Gaussian perturbation for complete image–text uploads, with additive noise comparisons under matched modality scales and whole-request budgets.

VLM input processing and privacy. VLM input processing addresses both inference efficiency and selective privacy protection. LVPruning uses language-guided cross-attention, and LAST selects query-relevant visual tokens before transmission (Sun et al., 2025; Yang et al., 2026). Progressive semantic communication learns compression and reconstruction for a frozen VLM (Hsu et al., 2026). For input privacy, Liu et al. (2026b) use cross-modal localization to protect sensitive image regions and associated text, while TGAP targets selected private-attribute inference (Li et al., 2026). We address complete image–text representation release under LDP, using compression and local decoding to preserve task utility at a frozen VLM's native token interface.

3 Problem Formulation

3.1 Private Inputs and Deployment

We consider a VLM inference request with private input $x = (I, T)$, where I denotes the images and T the contextual text. The public conditions are $h = (Q, s)$, where Q includes the question, task instructions, and answer options when applicable. The variable s specifies the declared public structure of the token representations, including their lengths and the arrangement of visual and textual tokens.

The client processes the encoded inputs locally and uploads token representations with the backbone's native layout and channel dimensions, while the compact noisy latent remains local. We model the cloud server as an adversary that observes the complete upload and public conditions, with full knowledge of the VLM, the client-side processing mechanism, and all fixed parameters. Client-side learning and calibration use public auxiliary data, and the resulting parameters are fixed before private inference.

3.2 Privacy and Utility Objectives

Our privacy unit is the complete private content of one request. For a fixed public state h, let $\mathcal{X}_h$ contain all allowed private inputs compatible with its declared structure, and let $\mathcal{R}_h(x)$ denote the randomized representation uploaded by the client. We require $\mathcal{R}_h$ to satisfy standard (ϵ, δ)-local differential privacy (LDP): for every $x, x' \in \mathcal{X}_h$ and measurable output event O,

$$\Pr[\mathcal{R}_h(x) \in O] \leq e^{\epsilon} \Pr[\mathcal{R}_h(x') \in O] + \delta. \tag{1}$$

The budget covers the complete client upload for one request, allowing all private image and contextual-text content to change simultaneously. The guarantee compares inputs with the same public question and declared token structure; the question, token counts, and layout remain public (Appendix B.1). Budgets reported for token-level protection require accounting under this replacement relation before numerical comparison (Dwork & Roth, 2014). Appendix C.3 specifies the accounting used for the evaluated methods.

Subject to this privacy constraint, we seek to preserve the frozen VLM's ability to answer the public question using the uploaded representations. Let G denote the frozen backbone, whose prediction is $\widehat{a} = G(Q, \mathcal{R}_h(x))$. We evaluate downstream task performance across privacy budgets, comparing release designs under the same whole-request replacement relation and declared (ϵ, δ)-LDP bound.

4 TASK-AWARE PRIVATE REPRESENTATION RELEASE

Our framework compresses the private representations, perturbs the compact latent, and decodes it locally into the frozen VLM's native token interface (Figure 1). Appendix Figure 5 details the compression and decoding architectures.

4.1 QUESTION-AWARE COMPRESSION

We compress the representations to reduce the dimension at which noise is added, using the public question to guide retention of task-relevant visual information. The frozen encoders produce visual tokens $H_v \in \mathbb{R}^{N_v \times d}$ and contextual-text tokens $H_t \in \mathbb{R}^{L \times d}$, where d is the native channel width and N_v, L are public token counts. After subtracting fixed public channel means, we reduce channel widths in both modalities and then compress visual positions.

The question guides the visual path at two steps. During channel reduction, its pooled embedding gates a low-rank projection branch. Before spatial compression, its token embeddings guide attention that lets visual states read both visual and question states. A learned spatial map then combines the updated visual positions into B coefficients; these mix information across positions rather than select B image regions. The textual path uses channel projection and within-text attention, retaining all L positions without interacting with the private visual stream. Channel and spatial maps are initialized from public PCA bases and trained with downstream task supervision.

The output tensors are $Z_v \in \mathbb{R}^{B \times k_v}$ and $Z_t \in \mathbb{R}^{L \times k_t}$, where k_v, k_t are the reduced channel widths. Writing θ for the compressor parameters, the complete mapping from private inputs to the flattened latent is

$$u = F_\theta(x; h) = (u_v, u_t) = (\operatorname{vec} Z_v, \operatorname{vec} Z_t). \tag{2}$$

The visual and textual blocks have dimensions $D_v = Bk_v$ and $D_t = Lk_t$ and are privatized together below. Appendix A.1 gives the internal mappings, attention masks, and normalization.

4.2 JOINT ELLIPSOIDAL PERTURBATION

The compact visual and textual blocks can still have different scales, which can make a common noise variance poorly suited to both. We give each modality its own scale within a shared clipping ellipsoid and match the Gaussian covariance to this geometry. For fixed public conditions h, let $c_h = (c_v, c_t)$ be a public latent center and $C_v, C_t > 0$ the modality scales, defining

$$A_h = \operatorname{diag}\big(C_v^2 I_{D_v}, C_t^2 I_{D_t}\big)\,. \tag{3}$$

We normalize the centered blocks by their scales and clip the concatenated vector to the unit ball, applying the same rescaling factor to both modalities. For a complete-request calibration parameter $\rho_0 > 0$, clipping and perturbation are

$$f_{E,h}(u) = c_h + \frac{u - c_h}{\max\{1, \|A_h^{-1/2}(u - c_h)\|_2\}}, \tag{4}$$

$$\widetilde{u} = f_{E,h}(u) + \sqrt{\frac{2}{\rho_0}} A_h^{1/2} \xi, \qquad \xi \sim \mathcal{N}(0, I_{D_v + D_t}). \tag{5}$$

The coordinate noise variances are $2C_v^2/\rho_0$ and $2C_t^2/\rho_0$; Section 5.1 converts ρ_0 to the reported (ϵ, δ) budget. The shared clipping constraint and whole-request analysis make this a joint release even though the Gaussian covariance is block diagonal. Shrinking a modality scale reduces its noise but strengthens shared clipping, so we select the scales using public calibration and development evaluation (Section 4.4).

4.3 NOISE-AWARE LOCAL DECODING

The noisy latent must be expanded into native token embeddings and adapted for inference by the frozen backbone. We first reconstruct the native layout, then refine the reconstructed tokens using the public question and noise level. The base decoder expands visual coefficients to native positions, mixes features within each modality, and restores channel dimensions and public means. Its output $H_0 = (H_{0,v}, H_{0,t})$ has the same block shapes as (H_v, H_t).

The refiner combines five streams in a shared attention space: the reconstructed visual and textual tokens, their two noisy latent blocks, and public-question embeddings. Reconstruction and latent states attend across modalities and to the question to predict corrections to H_0. The noise standard deviations $\sigma = (\sigma_v, \sigma_t)$ condition both the attention inputs and modality-specific channel modulation. For the joint ellipsoidal mechanism, these public inputs are $\sigma_g = C_g\sqrt{2/\rho_0}$ for $g \in \{v, t\}$. Writing $\mathcal{D}_{\phi_0}$ and $\mathcal{A}_\psi$ for the base decoder and residual refiner, with respective parameters ϕ_0, ψ, we obtain

$$\begin{aligned} H_0 &= \mathcal{D}_{\phi_0}(\widetilde{u}; s), \\ \widehat{H} &= H_0 + \mathcal{A}_\psi(\widetilde{u}, H_0; Q, \sigma, s). \end{aligned} \tag{6}$$

Together they form the decoder $\mathcal{D}_\phi$, with $\phi = (\phi_0, \psi)$, and the client uploads $\mathcal{R}_h(x) = \widehat{H}$. All private inputs to decoding derive from the same noisy latent; the cloud receives only the final tokens and public conditions. Appendix A.2 specifies normalization, attention masks, and noise conditioning.

4.4 Training and Calibration

Training uses public auxiliary data and keeps the pretrained encoders and cloud backbone frozen. We first train the compressor and base decoder with clean and privatized task supervision. Public latent statistics initialize the modality scales, while public calibration and development evaluation determine compression dimensions and release configurations.

For the main configurations, final refinement updates only ψ, keeping all preceding modules, release geometry, and normalization statistics fixed. It combines clean views that bypass clipping and noise with privatized views at multiple budgets. The objective combines full-vocabulary answer negative log-likelihood with a preservation penalty that discourages weakening correct base predictions from the same view. All parameters are frozen before private inference. Appendices A.3 and A.4 give the loss definitions, stage-specific training protocols, and calibration procedures.

5 Privacy Guarantee and Mechanism Analysis

5.1 Whole-Request LDP Guarantee

We establish privacy for the noisy latent and extend it to the complete upload through local decoding. Gaussian calibration uses zero-concentrated differential privacy (zCDP), a privacy measure with a standard conversion to (ϵ, δ)-DP (Bun & Steinke, 2016).

Theorem 1 (Privacy of the complete upload)**.** *Fix public conditions h and a budget $\rho_0 > 0$. Assume that the compressor parameters, center c_h, and positive-definite A_h are common to all $x \in \mathcal{X}_h$. Equation* (5) *gives ρ_0-zCDP for arbitrary replacement within $\mathcal{X}_h$, including simultaneous changes to images and contextual text. If decoding uses only the noisy latent, public information, and fixed parameters, the complete upload $\mathcal{R}_h(x)$ satisfies (ϵ, δ)-LDP for every $0 < \delta < 1$, with*

$$\epsilon = \rho_0 + 2\sqrt{\rho_0 \log(1/\delta)}. \tag{7}$$

Proof sketch. Joint clipping places the normalized means in the unit ball, bounding their pairwise distance by 2; the matched Gaussian noise therefore gives ρ_0-zCDP. Local decoding and cloud inference preserve this guarantee by post-processing (Dwork & Roth, 2014). Appendix B.1 gives the full proof, including the treatment of publicly disclosed variable-length layouts.

5.2 Noise Comparison under Matched Conditions

With compression, public conditions, center, positive modality scales, and request budget fixed, we compare expected noise energy $J = \mathbb{E}\|\eta\|_2^2$, where η is the additive Gaussian noise. This sums coordinate variances, excluding clipping and decoding effects. Each mechanism is calibrated over its full clipping range.

We denote the joint ellipsoidal mechanism by **Joint**. **ISO** reuses Joint's clipped representation and adds isotropic noise with variance $2C_{\max}^2/\rho_0$, where $C_{\max} = \max(C_v, C_t)$. In the modality-separated family, each block is clipped independently to radius C_g and perturbed with Gaussian

covariance $2C_g^2 I_{D_g}/\rho_g$, for $g \in \{v, t\}$. Because both modalities may change within a request, their budgets satisfy $\rho_v + \rho_t = \rho_0$. **MOD** denotes the member of this family whose budget allocation minimizes the expected additive noise energy J; Appendix B.2 gives the allocation.

Table 1: Expected noise energy at matched modality scales and request budget.

Mechanism	Expected noise energy
Joint	$J_E = 2(D_v C_v^2 + D_t C_t^2)/\rho_0$
ISO	$J_I = 2(D_v + D_t)C_{\max}^2/\rho_0$
MOD	$J_M^* = 2(C_v\sqrt{D_v} + C_t\sqrt{D_t})^2/\rho_0$

Proposition 1 (Comparison of additive noise)**.** *Under these full-range calibrations, with $D_v, D_t > 0$, the costs in Table 1 satisfy $J_E \le J_I$ and $J_E < J_M^*$. The first inequality is strict when $C_v \ne C_t$.*

At fixed modality scales, Joint's expected noise energy grows linearly with latent dimension. Relative to ISO, Joint reduces variance in the smaller-scale modality; relative to MOD, it trades stronger clipping for lower noise. Proposition 1 characterizes additive noise, while Appendix B.3 analyzes the resulting clipping–noise tradeoff.

6 Experiments

6.1 Experimental Setup

Datasets and tasks. We evaluate multiple-choice answering on ScienceQA (Lu et al., 2022) and answer generation on a single-image subset of WebQA (Chang et al., 2022). The evaluation includes 814 ScienceQA questions with images and textual hints from the official test split, after excluding known image overlap with our training data, and 755 WebQA questions from the official validation split. To examine questions less reliably answered from public information alone, we also report a fixed 427-question subset on which an earlier Public-only generation evaluation made at least one error across cyclic option orderings (Appendix C.1). Images and hints or captions are private; questions, answer options when present, task instructions, and declared token structure are public.

Models and implementation. We use InternVL3.5-8B (Wang et al., 2025) on both datasets and Qwen2.5-VL-7B-Instruct (Bai et al., 2025) on ScienceQA. The pretrained encoders and backbones remain frozen, while all release stages run locally. For the main study, we train separate client modules for each backbone using public auxiliary data, select refinement checkpoints on development data, and fix all parameters and calibrated scales across evaluation budgets (Appendix A.4). For ScienceQA with InternVL3.5-8B, the additional client modules contain 18.52M parameters, and the complete client reaches 1.90 GiB of peak PyTorch-allocated GPU memory for a representative request (Appendix C.6).

Baselines and references. We compare the full framework (Ours) with **Native-LDP**, which perturbs native representations, and **PCA-based release**, which uses fixed Principal Component Analysis (PCA) compression and linear reconstruction. All three use Joint with their own publicly calibrated scales. On ScienceQA/InternVL, we adapt DP-BART (Igamberdiev & Habernal, 2023) and DP-Prompt (Utpala et al., 2023) through three routes: privatizing an image caption together with the text (**Caption+context**), or composing text privatization with either **Visual-token LDP** or **DP-Pix** on RGB block means (Fan, 2018). Appendix C.2 details these adaptations. **Public-only** omits private inputs; **Raw input** uses original inputs without compression, clipping, or noise.

Privacy budgets and evaluation. Comparisons use whole-request (ϵ, δ)-LDP bounds with $\delta = 10^{-6}$, covering simultaneous replacement of private images and text. The main curves contain nine budgets from $\epsilon \approx 71$ to 102,351, with detailed comparisons at 954, 3,407, and 10,743. We extend ScienceQA/InternVL to $\epsilon = 0.1$–32 and selected larger baseline budgets (Appendices C.3 and D.4). ScienceQA accuracy uses the most probable answer letter, averaged over cyclic option orderings sharing the same private release; WebQA uses greedy generation and a custom, reference-derived keyword exact-match score (Appendix C.4). Main-curve scores average three noise realizations and

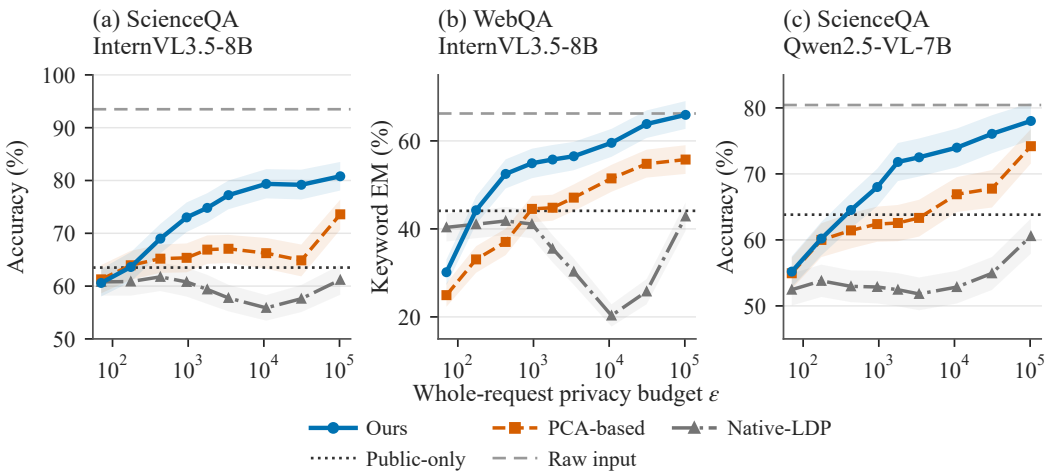


Figure 2: **Privacy–utility tradeoffs across datasets and backbones.** Whole-request ϵ uses $\delta = 10^{-6}$; intervals are pointwise 95% bootstrap intervals. Public-only excludes private inputs; Raw input is unmodified. ScienceQA reports accuracy; WebQA reports custom keyword exact match.

Table 2: **Utility at expanded baseline budgets on ScienceQA/InternVL.** Whole-request bounds use $\delta = 10^{-6}$; ratios compare the listed budgets with Ours.

Method	Evaluated ϵ	Accuracy (%)	Budget ratio
Ours	3.41×10^3	77.23	1×
Native-LDP	5.68×10^5	76.02	167×
PCA-based	1.00×10^9	74.96	293,568×
DP-BART + Visual-token LDP†	8.02×10^6	79.86	2,354×
DP-Prompt + Visual-token LDP†	3.18×10^6	76.10	932×

† Appendix D.4.2 gives the exploratory selection protocol, exact budgets, and paired intervals.

weight questions equally. Pointwise 95% bootstrap intervals resample image clusters for ScienceQA and questions for WebQA (Appendix C.5).

6.2 Privacy–Utility Tradeoffs

Overall utility across datasets and backbones. Our framework outperforms Native-LDP and PCA-based release at every measured budget from $\epsilon \approx 429$ onward on all three main curves (Figure 2). At $\epsilon \approx 3{,}407$, ScienceQA/InternVL accuracy is 77.23%, versus 67.07% for PCA and 57.76% for Native-LDP. Thus compact release already improves utility, and the task-trained pipeline improves it further. On WebQA, our framework reaches 56.51% keyword EM at the same budget, exceeding PCA-based release and Native-LDP by 9.40 and 26.23 points. The gains extend to Qwen on ScienceQA: accuracy reaches 72.51%, improving over the two baselines by 9.18 and 20.70 points.

Adapted text-private baselines. Our framework also provides higher utility than all six multimodal adaptations of DP-BART and DP-Prompt, exceeding the best adaptation at each primary budget by 8.0–15.2 points on ScienceQA/InternVL (Table 5).

Utility at expanded budgets. Table 2 compares our representative operating point with selected larger-budget baseline configurations. Native-LDP and DP-Prompt with visual-token LDP reach 76.02% and 76.10% at approximately 167× and 932× our reference budget, respectively, compared with our 77.23%.

Low-budget adaptation. To address noise levels beyond the main checkpoint's training range, we fine-tune the compressor and decoder at $\epsilon = 16$ with public-statistics-based Wiener-style shrinkage

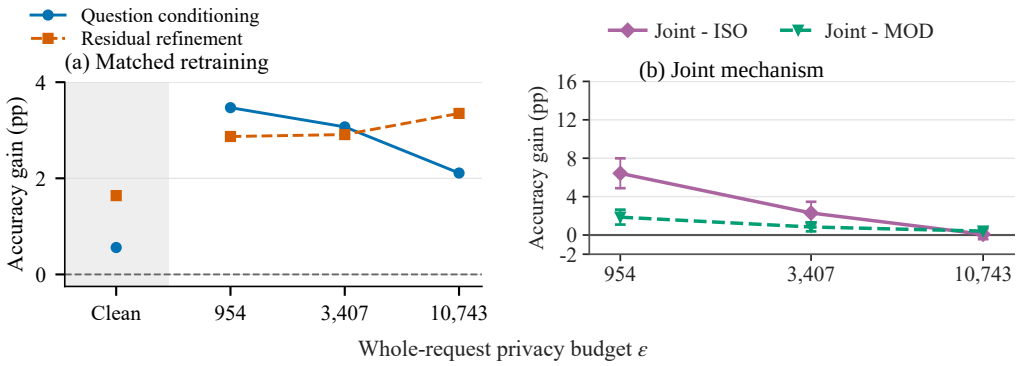


Figure 3: **Core component gains on ScienceQA/InternVL (814 questions).** (a) Question conditioning measures the gain from providing question inputs to the compressor; residual refinement measures the additional gain with residual refinement. (b) Joint versus ISO and MOD with the main model's fixed modules and modality scales held fixed; error bars show paired 95% bootstrap intervals. Clean bypasses clipping/noise; private budgets use $\delta = 10^{-6}$. Subset results are in Appendices D.2 and D.3.

before decoding. A single adapted checkpoint achieves 67.37–67.54% accuracy at $\epsilon = 8, 16, 32$, compared with 63.51% for Public-only. At $\epsilon = 8$, the gain is 4.03 percentage points (paired 95% CI: $[2.67, 5.40]$). At $\epsilon \approx 954$ and 3407, however, the adapted variant obtains 68.17% and 73.76%, falling below the main checkpoint by 4.84 and 3.47 points. This trade-off motivates treating it as a dedicated low-budget configuration (Appendix D.4.1).

6.3 Component Analysis

We first use independently retrained models to compare compressors with and without public-question conditioning, then add residual refinement to the selected question-conditioned codec. We separately evaluate module substitutions and release mechanisms using the main ScienceQA/InternVL checkpoint.

Question conditioning and residual refinement. Question-conditioning compression improves accuracy beyond task training alone (Figure 3). With architecture, initialization, task supervision, and training schedules matched, question conditioning compression outperforms the one without question by 3.47, 3.07, and 2.11 points across the three private budgets, highlighting the value of question conditioning under privatization. Subsequent residual refinement further improves accuracy by 2.87–3.35 points. On the fixed 427-question subset the respective gains at $\epsilon \approx 3{,}407$ are 4.48 and 6.28 points. Appendix D.3 gives the staged training protocol and complete accuracies.

Task-trained compression and decoding. These substitutions keep the main checkpoint's release geometry fixed, without retraining or recalibration. With its complete decoder fixed, replacing PCA compression with learned compression improves accuracy by 6.24 points at $\epsilon \approx 3{,}407$. This PCA substitution differs from the independently calibrated PCA-based pipeline in Figure 2. With the learned compressor fixed, the complete decoder improves over PCA reconstruction by 2.93 points. Both gains occur at all three private budgets; Appendix D.2 reports all substitutions.

Joint ellipsoidal perturbation. With trained modules fixed, Joint exceeds ISO and MOD by 6.43 and 1.86 points at $\epsilon \approx 954$ (Figure 3(b)). Under a privacy budget with higher noise levels, the advantages of the Joint mechanism become even more pronounced. On the 427 subset, which relies more heavily on private content, Joint demonstrated stronger private evidence retention capabilities. Appendix Figure 7 gives all budgets and subset results.

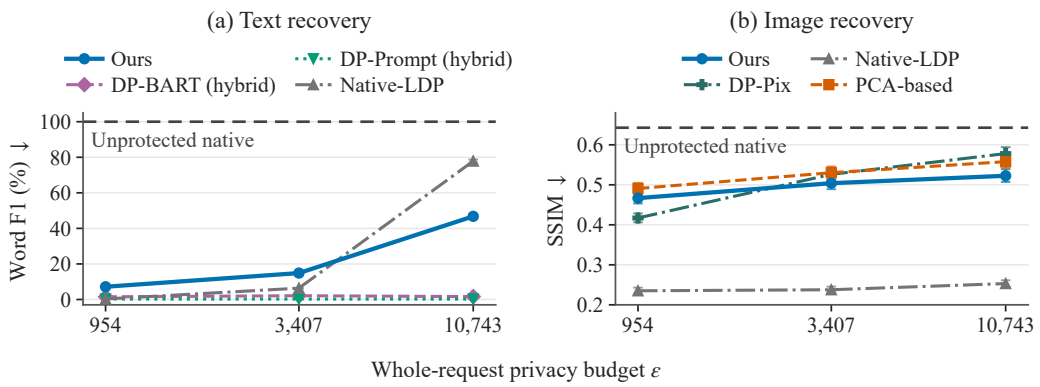


Figure 4: **Text and image recovery on ScienceQA/InternVL**; lower is better under each metric. Text scores use vocabulary matching for representation releases and directly evaluate the rewritten strings for DP-BART and DP-Prompt. Error bars show 95% image-cluster bootstrap intervals; dashed lines show unprotected native tokens. The DP-BART/DP-Prompt text curves and DP-Pix image curve use their hybrid allocations of 0.75ϵ and 0.25ϵ, respectively, while the horizontal axis gives the whole-request budget. Table 11 pairs these recovery metrics with task utility, and Figure 10 reports LPIPS (Zhang et al., 2018).

6.4 Empirical Inversion Analysis

We use fixed recovery procedures to compare content recoverability while controlling the attack component of the evaluation. For representation releases, we recover text by vocabulary matching and apply an image inverse trained on public native-token/image pairs and fixed across defenses. For DP-BART and DP-Prompt, text recovery is scored from their released rewritten strings. We then evaluate how each release pipeline changes recovery alongside downstream utility.

The complete release pipeline reduces content recovery under the evaluated attacks while retaining task utility. At $\epsilon \approx 3{,}407$, Word F1 falls from 100.00% for unprotected native tokens to 14.85%, and image SSIM (Wang et al., 2004) from 0.6428 to 0.5041, while QA accuracy remains 77.23%. Native-LDP achieves lower text recovery (6.37% Word F1), but its accuracy is only 57.76%. Appendix E details the attack interfaces and training and reports all baseline results and LPIPS.

7 Discussion and Conclusion

Discussion and limitations. The dimensional cost in Section 5.2 helps explain the difficulty of perturbing native tokens; prior sequence-level LDP work also reports utility difficulties when representations are perturbed near the model input (Du et al., 2023). The main comparisons use large budgets with loose worst-case indistinguishability bounds. Targeted adaptation achieves accuracy above Public-only at $\epsilon = 8, 16, 32$, with lower accuracy than the main checkpoint at the two larger budgets evaluated. The inversion results provide complementary evidence about content recovery under the evaluated fixed attacks.

Conclusion. We presented a task-aware framework for private VLM inference that combines question-guided compression, joint ellipsoidal perturbation, and noise-aware local decoding while keeping the cloud backbone frozen. We established LDP for the complete private image–text upload, conditional on the public question and declared structure, and analyzed additive noise costs under matched comparison conditions. Experiments on ScienceQA and WebQA, including a second VLM backbone, support improved empirical privacy–utility tradeoffs over the evaluated release pipelines across most tested budgets.

AI USE STATEMENT

We used generative AI tools to assist language editing, identifying and summarizing relevant literature, and developing, debugging, and executing experimental code. AI tools also supported experimental design, figure preparation and preparation of reference-derived WebQA scoring annotations. The authors take full responsibility for the final manuscript, methods, code, and reported results, including all AI-assisted contributions.

REPRODUCIBILITY STATEMENT

Appendix A describes the client modules, training procedures, and public calibration, while Appendix B provides the privacy proofs and mechanism analysis. Appendices C, D, and E document the evaluation protocols and supplementary results. The anonymous supplementary package includes code for reproducing the main experimental results using the provided trained client-module checkpoints.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. arXiv preprint arXiv:2502.13923, 2025.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography*, volume 9985 of *Lecture Notes in Computer Science*, pp. 635–658, 2016. doi: 10.1007/978-3-662-53641-4_24.

Chaoxin Cai, Yingpeng Sang, and Hui Tian. A multimodal differential privacy framework based on fusion representation learning. *Connection Science*, 34(1):2219–2239, 2022. doi: 10.1080/09540091.2022.2111406.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. WebQA: Multihop and multimodal QA. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16495–16504, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Chang_WebQA_Multihop_and_Multimodal_QA_CVPR_2022_paper.html.

Thee Chanyaswad, Alex Dytso, H. Vincent Poor, and Prateek Mittal. MVG mechanism: Differential privacy under matrix-valued query. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 230–246, 2018. doi: 10.1145/3243734.3243750.

Jiangnan Cheng, Ao Tang, and Sandeep Chinchali. Task-aware privacy preservation for multi-dimensional data. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3835–3851, 2022. URL https://proceedings.mlr.press/v162/cheng22a.html.

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 *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2665–2679, 2023. doi: 10.1145/3576915.3616592.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014. doi: 10.1561/0400000042.

Liyue Fan. Image pixelization with differential privacy. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pp. 148–162, 2018. doi: 10.1007/978-3-319-95729-6_10.

Yuan Gao, Yingjie Xia, Xuejiao Liu, Oubo Ma, Shuaifan Jin, and Jia-Nan Liu. Structure-aware local differential privacy for multimodal collaborative inference in federated learning systems. *IEEE Transactions on Dependable and Secure Computing*, pp. 1–14, 2026. doi: 10.1109/TDSC.2026.3698059. Early access.

Cyril Shih-Huan Hsu, Wig Yuan-Cheng Cheng, and Chrysa Papagianni. Progressive semantic communication for efficient edge-cloud vision-language models. arXiv preprint arXiv:2604.26508, 2026.

Timour Igamberdiev and Ivan Habernal. DP-BART for privatized text rewriting under local differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13914–13934, 2023. doi: 10.18653/v1/2023.findings-acl.874.

Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ACM SIGARCH Computer Architecture News*, 45(1):615–629, 2017. doi: 10.1145/3093337.3037698.

Yuyang Li, Devika Gumaste, Mehmet Kerem Turkcan, Javad Ghaderi, Gil Zussman, and Zoran Kostic. Distributed VLMs: Efficient vision-language processing through cloud-edge collaboration. In *2025 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 280–286, 2025. doi: 10.1109/PerComWorkshops65533.2025.00078.

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*, volume 36, pp. 54554–54566, 2023. doi: 10.52202/075280-2377.

Haoyu Liu, Yuwen Chen, and Zhen Yang. Adaptive generative model inversion attacks in edge–cloud collaborative inference systems. *IEEE Transactions on Mobile Computing*, 25(9):13846–13862, 2026a. doi: 10.1109/TMC.2026.3673987.

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, 2026b. doi: 10.1016/j.cose.2026.104931.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, volume 35, pp. 2507–2521, 2022. doi: 10.52202/068431-0182.

Peihua Mai, Ran Yan, Zhe Huang, Youjia Yang, and Yan Pang. Split-and-Denoise: Protect large language model inference with local differential privacy. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34281–34302, 2024. URL `https://proceedings.mlr.press/v235/mai24a.html`.

John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. Text embeddings reveal (almost) as much as text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12448–12460, 2023. doi: 10.18653/v1/2023.emnlp-main.765.

Yizheng Sun, Yanze Xin, Hao Li, Jingyuan Sun, Chenghua Lin, and Riza Batista-Navarro. LVPruning: An effective yet simple language-guided vision token pruning approach for multi-modal large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4299–4308, 2025. doi: 10.18653/v1/2025.findings-naacl.242.

Saiteja Utpala, Sara Hooker, and Pin-Yu Chen. Locally differentially private document generation using zero shot prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8442–8457, 2023. doi: 10.18653/v1/2023.findings-emnlp.566.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265, 2025.

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. doi: 10.1109/TIP.2003.819861.

Feng Yang, Xinrui Ju, Keyang Zhang, Xiandong Meng, Rongqun Lin, Howard Leung, Shiqi Wang, Haoliang Li, and Chris Xing Tian. LAST: The last query token guides visual token pruning for edge-cloud collaborative MLLM inference. arXiv preprint arXiv:2607.27952, 2026.

JinYi Yoon, JiHo Lee, Ting He, Nakjung Choi, and Bo Ji. S2M3: Split-and-share multi-modal models for distributed multi-task inference on the edge. In *2025 IEEE 45th International Conference on Distributed Computing Systems (ICDCS)*, pp. 868–878, 2025. doi: 10.1109/ICDCS63083.2025.00089.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

A METHOD AND TRAINING DETAILS

A.1 COMPRESSION AND BASE RECONSTRUCTION

The compression mapping in Equation (2) first reduces channels and then compresses visual positions. For $g \in \{v, t\}$, write $X_g = H_g - \mu_g$, where the fixed public channel means $\mu_g \in \mathbb{R}^d$ are broadcast over token positions and are distinct from the latent clipping center c_h. The internal mappings are

$$\begin{aligned} K_v &= X_v W_{v,e} + \mathcal{B}_\theta(X_v, Q), \\ Z_v &= S_e^\top \mathcal{G}_\theta(K_v, Q), \qquad Z_t = \mathcal{T}_\theta(X_t W_{t,e}). \end{aligned} \tag{8}$$

Here, $W_{g,e} \in \mathbb{R}^{d \times k_g}$ reduces channel width, and $K_v \in \mathbb{R}^{N_v \times k_v}$ is the visual representation after question-gated channel reduction. The branch $\mathcal{B}_\theta$ applies the pooled-question gate defined below, while $\mathcal{G}_\theta$ denotes visual mixing and question-guided attention before the spatial map $S_e \in \mathbb{R}^{N_v \times B}$. Visual queries read visual and question states, while question queries read only question states; only the updated visual rows enter spatial compression. The text mixer $\mathcal{T}_\theta$ attends within the private contextual text, without interacting with the visual stream. For pooled public-question embeddings $\bar{e}_Q$, the question-gated branch is

$$g(Q) = \mathbf{1} + \tfrac{1}{2} \tanh(\mathrm{MLP}(\bar{e}_Q / s_Q)), \tag{9}$$

$$\mathcal{B}_\theta(X_v, Q) = s_v \big[((X_v U) \oslash r_U) \odot g(Q) \big] A_q, \tag{10}$$

where $U \in \mathbb{R}^{d \times r}$, $A_q \in \mathbb{R}^{r \times k_v}$, and s_Q, s_v, r_U are fixed public normalization statistics. The gate is broadcast over visual positions; U starts from public residual PCA directions and A_q starts at zero. Before question-guided visual attention, an initial mixer attends within each private modality.

Let $\widetilde{Z}_v$ and $\widetilde{Z}_t$ be the two blocks of $\widetilde{u}$, reshaped to match Z_v and Z_t. The base decoder expands the visual coefficients before modality-block mixing and channel reconstruction:

$$(U'_v, U'_t) = \mathrm{Mix}_d(S_d \widetilde{Z}_v, \widetilde{Z}_t; s), \qquad H_{0,g} = U'_g W_{g,d} + \mu_g, \tag{11}$$

with $S_d \in \mathbb{R}^{N_v \times B}$ and $W_{g,d} \in \mathbb{R}^{k_g \times d}$. It has no separate question or noise-scale input and restores all native visual positions.

A.2 QUESTION- AND NOISE-CONDITIONED REFINEMENT

For the five refinement streams introduced in Section 4.3, the question stream uses frozen embeddings E_Q. Each stream uses fixed RMS normalization estimated from public fitting views; native reconstructions also subtract public channel means. The streams are projected into a shared space with type and positional embeddings. Reconstruction and latent rows attend across modalities and to the question, while question rows attend only to question rows. A noise MLP embeds $\log(1 + \sigma \oslash (r_v, r_t))$, where r_g is the public latent RMS, and conditions both attention inputs and modality-specific channel modulation. Native-space output heads add residual corrections to H_0; zero initialization makes the initial complete decoder equal the base decoder.

The supplied standard deviation is $C_g\sqrt{2/\rho_0}$ for Joint, $C_{\max}\sqrt{2/\rho_0}$ for ISO, and $C_g\sqrt{2/\rho_g^*}$ for MOD. Clean and clip-only views use $\sigma = 0$ and still pass through refinement. ScienceQA client conditioning canonically sorts option texts and removes answer-letter labels, while the backbone receives the displayed order; WebQA uses its public question.

A.3 TRAINING OBJECTIVES

Base training uses task supervision with dataset-specific objectives and schedules. For ScienceQA/InternVL, clean base-decoder training uses full-vocabulary answer NLL, whereas fixed-axis base adaptation uses candidate-normalized NLL averaged over cyclic option orderings. Adaptation weights are proportional to $1 + 2\,\mathrm{mean}_k\, \mathbf{1}[\mathrm{PublicWrong}_{i,k} \wedge \mathrm{NativeCorrect}_{i,k}]$ and normalized to mean one, using generated Public-only answers and native candidate argmax on fitting data. This adaptation pairs a $\rho_0 = 3000$ view with clean, 750, or 10000 views, updating visual maps, question modules, and the base decoder while freezing the text encoder and encoder mixer. Qwen

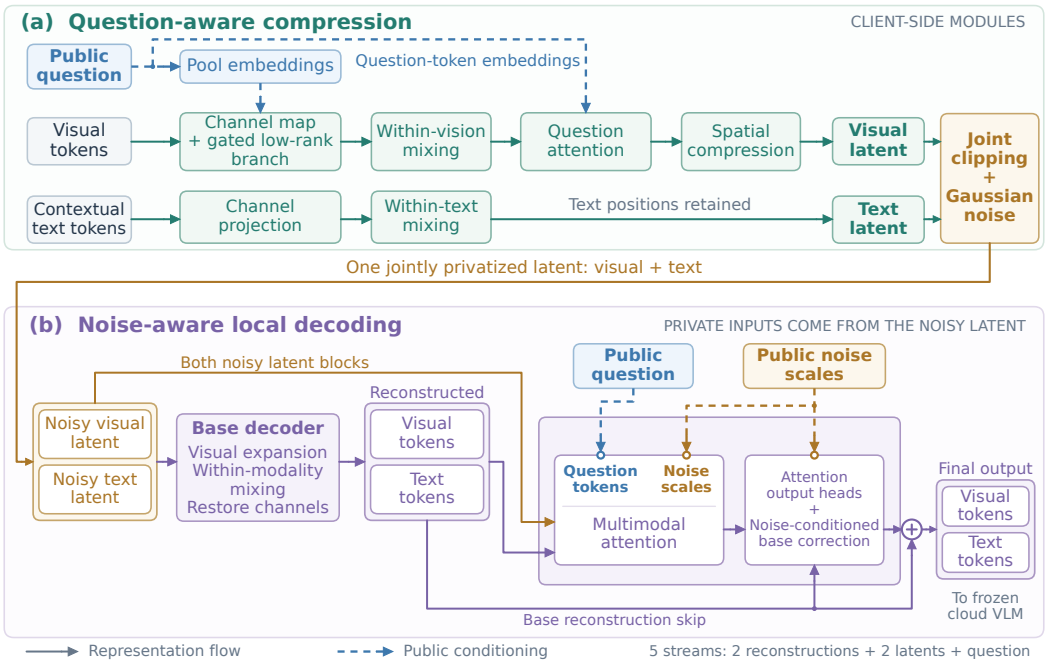


Figure 5: Client-side compression and decoding. (a) Public-question embeddings guide visual channel reduction and attention; private contextual-text processing remains separate. Spatial compression mixes visual positions rather than selecting image regions. Both latent blocks undergo one joint ellipsoidal clipping and Gaussian perturbation step. (b) The base decoder restores native token shapes. Refinement attends over five streams: two reconstructions, two noisy latent blocks, and public-question embeddings. A shared embedding of the public noise scales conditions attention inputs and channel gates. The right-hand box sums attention-head outputs from reconstruction-token states and a gated correction of the normalized base reconstruction; the resulting native-scale residual is added to the base tokens before upload. Fixed centering, normalization, and output rescaling are omitted. Solid arrows show representation flow; dashed arrows show public conditioning.

uses candidate-normalized NLL in separate clean and noisy base-training stages. WebQA uses full-vocabulary answer NLL including EOS, first on clean views and then with equal clean/noisy weights at $\rho_0 = 3000$, plus clean visual reconstruction MSE. Final refinement omits these example weights and reconstruction losses.

The final refinement objective combines answer supervision and preservation of correct base predictions:

$$\mathcal{L}_{\text{adapt}}(\psi) = \mathcal{L}_{\text{QA}}(\psi) + \mathcal{L}_{\text{pres}}(\psi). \tag{12}$$

Here, $\mathcal{L}_{\text{QA}}$ is full-vocabulary answer negative log-likelihood through the frozen backbone, and $\mathcal{L}_{\text{pres}}$ averages the preservation penalty defined below over the task-specific prediction units. Base and refined predictions are evaluated on the same clean or noisy view. With detached base distribution p_0 and refined distribution p_ψ, define

$$\begin{aligned} m_j(a) &= \log p_j(a) - \max_{b \neq a} \log p_j(b), \qquad j \in \{0, \psi\}, \\ \mathcal{P}(a; p_0, p_\psi) &= \mathbf{1}[\arg\max_b p_0(b) = a]\,[\min(2, \max(0, m_0(a))) - m_\psi(a)]_+. \end{aligned} \tag{13}$$

ScienceQA adds full-vocabulary NLL of the correct option letter to $\mathcal{P}$ computed over candidate letters, averaging over cyclic orderings of the same release. For WebQA, the base and refined models use full-vocabulary distributions under the same teacher-forced reference prefix; answer NLL is averaged over content tokens and EOS, while $\mathcal{P}$ is averaged over content positions only. Its preservation denominator includes all content positions, including those with zero penalty, and excludes EOS. Both datasets use preservation weight one.

Table 3: Final configurations: $N_v = 256$, $k_v = 128$, text latent $L \times 512$, and zero latent center. Axes are rounded to six decimal places.

Configuration	Native width d	Spatial coefficients B	C_v	C_t
ScienceQA / InternVL	4096	8	231.751346	31.017218
WebQA / InternVL	4096	32	355.538578	19.472061
ScienceQA / Qwen	3584	8	722.661610	12.567220

For the main configurations, final refinement updates only its residual parameters, following the repeating cycle clean, 3000, 750, 3000, 10000, 3000, where numbers denote ρ_0. Each main-study configuration uses one selected refinement checkpoint across all its evaluation budgets.

A.4 Public Calibration and Configurations

Public calibration fixes the release geometry before private inference, using zero latent center and positive global modality scales shared across public text lengths. Statistical initialization takes each block's 90th-percentile norm and multiplies both scales by the 90th percentile of the resulting normalized joint radius. ScienceQA/InternVL then task-calibrates the axes with frozen base modules using the bounded loss $1 - \exp(-\ell)$, where ℓ is full-vocabulary answer NLL, and the fixed fitting-example weights. WebQA uses statistical axes; Qwen recalibrates after clean base training. Dimensions and scales are selected through public development comparisons, including clean-capacity and privatized-utility checks. Table 3 gives the final dimensions and modality scales.

B Privacy Proofs and Mechanism Analysis

B.1 Complete-Upload Privacy Guarantee

Fix $h = (Q, s)$, $\rho_0 > 0$, and arbitrary $x, x' \in \mathcal{X}_h$, with identical compressor and decoder parameters, center c_h, and positive-definite axes A_h for both inputs. Let $\mu_h(x) = f_{E,h}(F_\theta(x; h))$ and $z_h(x) = A_h^{-1/2}(\mu_h(x) - c_h)$. The clipping rule in Equation (4) ensures $\|z_h(x)\|_2 \le 1$, hence

$$\|A_h^{-1/2}(\mu_h(x) - \mu_h(x'))\|_2 = \|z_h(x) - z_h(x')\|_2 \le 2. \tag{14}$$

For $\Delta = \mu_h(x) - \mu_h(x')$ and the common covariance $\Sigma_E = 2A_h/\rho_0$, the Gaussian Rényi divergence satisfies, for every $\alpha > 1$ (Bun & Steinke, 2016),

$$\begin{aligned} D_\alpha(\mathcal{L}(\widetilde{u} \mid x, h) \,\|\, \mathcal{L}(\widetilde{u} \mid x', h)) &= \frac{\alpha}{2}\Delta^\top \Sigma_E^{-1}\Delta \\ &= \frac{\alpha\rho_0}{4}\|A_h^{-1/2}\Delta\|_2^2 \le \alpha\rho_0. \end{aligned} \tag{15}$$

Thus, the complete latent release satisfies ρ_0-zCDP even when all private images and contextual text change simultaneously. For every $0 < \delta < 1$, the Rényi-to-DP conversion gives $(\alpha\rho_0 + \log(1/\delta)/(\alpha - 1), \delta)$-LDP (Bun & Steinke, 2016). Choosing $\alpha = 1 + \sqrt{\log(1/\delta)/\rho_0}$ yields $\epsilon = \rho_0 + 2\sqrt{\rho_0 \log(1/\delta)}$, as in Equation (7).

The actual upload $\mathcal{R}_h(x) = \mathcal{D}_\phi(\widetilde{u}; h, \sigma)$ preserves this bound by post-processing (Dwork & Roth, 2014). Its base reconstruction $H_0 = \mathcal{D}_{\phi_0}(\widetilde{u}; s)$ uses the same release, and refinement reads only that release, H_0, fixed parameters, and public conditions. In particular, σ is fixed by the public mechanism, dimensions, and budget; neither the original private latent nor the realized noise enters private-inference decoding. Attaching public h and applying the cloud backbone are also post-processing, proving Theorem 1.

Variable public lengths do not require a common output dimension across requests. For the representation space $\mathcal{Y}_s$ associated with layout s and any measurable event O for the tagged server view $(h, \mathcal{R}_h(x))$, define $O_h = \{y \in \mathcal{Y}_s : (h, y) \in O\}$. Applying the fixed-$h$ LDP inequality to O_h proves the guarantee for this complete view. The compared inputs must share h; public questions, token counts, and layout are not protected by this conditional guarantee.

B.2 Matched-Axis Noise Comparison

Fix the same public state, positive modality scales C_v, C_t, dimensions $D_v, D_t > 0$, and total budget ρ_0 for all three mechanisms. Joint and ISO share the clipped ellipsoid, whose Euclidean diameter is $2C_{\max}$, where $C_{\max} = \max(C_v, C_t)$. The Gaussian divergence formula therefore gives ISO's smallest variance certified against this full envelope as $2C_{\max}^2/\rho_0$ per coordinate. Writing $a_g = D_g C_g^2$, covariance traces give $J_E = 2(a_v + a_t)/\rho_0$ and $J_I = 2(D_v + D_t)C_{\max}^2/\rho_0$.

For $\rho_g > 0$, MOD clips block g to radius C_g and adds independent Gaussian noise with covariance $2C_g^2 I_{D_g}/\rho_g$. Each block has Rényi divergence at most $\alpha\rho_g$, and the conditional product Gaussian distributions add these bounds to $\alpha(\rho_v + \rho_t)$. Thus, $\rho_v + \rho_t \le \rho_0$ suffices for whole-request privacy, regardless of statistical dependence between modalities; parallel composition does not apply because both may change. Its noise cost is $J_M = 2(a_v/\rho_v + a_t/\rho_t)$, and Cauchy–Schwarz gives

$$\left(\frac{a_v}{\rho_v} + \frac{a_t}{\rho_t}\right)(\rho_v + \rho_t) \ge (\sqrt{a_v} + \sqrt{a_t})^2. \tag{16}$$

Equality uses the full budget with the allocation

$$\rho_g^* = \rho_0 \frac{C_g\sqrt{D_g}}{C_v\sqrt{D_v} + C_t\sqrt{D_t}}, \qquad g \in \{v, t\}. \tag{17}$$

Substitution gives the minimum noise energy $J_M^* = 2(\sqrt{a_v} + \sqrt{a_t})^2/\rho_0$. Consequently,

$$J_I - J_E = \frac{2}{\rho_0} \sum_{g\in\{v,t\}} D_g(C_{\max}^2 - C_g^2) \ge 0, \tag{18}$$

$$J_M^* - J_E = \frac{4C_vC_t\sqrt{D_vD_t}}{\rho_0} > 0. \tag{19}$$

The first inequality is strict for $C_v \ne C_t$, completing Proposition 1 and Table 1.

B.3 Clipping–Noise Tradeoff

Joint's smaller noise cost relative to MOD is accompanied by stronger clipping. For fixed u, let $e_g = u_g - c_g$, $r_g = \|e_g\|_2/C_g$, $\lambda_g = 1/\max\{1, r_g\}$, and $\lambda_E = 1/\max\{1, \sqrt{r_v^2 + r_t^2}\}$. Separate clipping retains $c_g + \lambda_g e_g$, whereas Joint retains $c_g + \lambda_E e_g$. Let $f_E = f_{E,h}$ and f_M denote these joint and separate clipping maps, with $B_k(u) = \|f_k(u) - u\|_2^2$ for $k \in \{E, M\}$. The ordering $\lambda_E \le \lambda_g \le 1$ implies

$$B_E(u) - B_M(u) = \sum_g \left[(1-\lambda_E)^2 - (1-\lambda_g)^2\right]\|e_g\|_2^2 \ge 0. \tag{20}$$

Zero-mean Gaussian noise gives the exact decomposition

$$\mathbb{E}\|f_k(u) + \eta_k - u\|_2^2 = B_k(u) + \mathrm{tr}(\Sigma_k). \tag{21}$$

Joint has lower total latent squared error than MOD when its noise saving exceeds the additional clipping cost; ISO shares Joint's clipping error. We evaluate the downstream effects through task accuracy.

B.4 Calibration and Release Scope

Public calibration fixes all learned parameters, centers, axes, and dimensions before private inference, or selects them by a fixed rule using only declared public conditions. Selecting the covariance from the current private latent is not covered, and the inference guarantee does not protect auxiliary training records. All three mechanisms use the sufficient zCDP conversion in Equation (7).

Reusing one release for public option orderings or additional downstream answers is post-processing. New randomized releases compose to $\rho_{\mathrm{tot}} = \sum_j \rho_j$, followed by the same conversion, provided every conditional mechanism satisfies its bound under the same private-input replacement

relation (Bun & Steinke, 2016). Adaptive public questions must disclose no additional unprivatized private information. Repeated noise realizations in evaluation estimate utility; uploading them would require composition.

The guarantee assumes ideal Gaussian sampling with fresh internal randomness. Seeded evaluations estimate utility; finite-precision deployment requires separate implementation validation.

C Experimental Protocols

C.1 Data and Public Conditions

ScienceQA uses 1,024 training and 256 development examples, plus 814 official-test questions with images and eligible hints after excluding known training-image overlap (Lu et al., 2022). Both backbones use this evaluation roster, containing 737 exact-RGB image clusters. The 256 PCA-calibration examples belong to the auxiliary-training set. The fixed 427-question subgroup comprises 403 clusters and was selected because historical Public-only *generated* answers were not correct under every cyclic option ordering. Selection uses historical generated answers rather than current candidate-answer scores.

WebQA uses 5,112 training, 128 development, and 755 official-validation examples from the color, shape, number, and YesNo categories (Chang et al., 2022). Each has one positive image and a complete caption of 1–128 InternVL tokens. Development is balanced across these four categories. Exact-source deduplication leaves 755 distinct evaluation images; evaluation retains natural category proportions. Neither primary evaluation roster is filtered by model correctness.

Complete images and original hints/captions are private; questions, options, instructions, token counts, and layouts are public. The replacement relation allows both private modalities to change simultaneously. Auxiliary data are treated as public, and local processing releases only final token representations and public conditions to the cloud. This inference-time guarantee does not protect auxiliary training records.

C.2 Baselines

Native-LDP applies Joint to centered native representations without compression; PCA-based release uses fixed public projections, Joint, and linear reconstruction. Each uses its own frozen scales, whereas controlled component swaps share ours' scales.

DP-BART uses the authors' open-source checkpoint (Igamberdiev & Habernal, 2023). It applies coordinate clipping and analytically calibrated Gaussian noise to a fixed padded representation, with length checks rather than silent text truncation. Decoding uses fixed public masks without original token IDs or private masks. DP-Prompt uses frozen InternVL with enforced public logit bounds and full-vocabulary sampling calibrated for a fixed maximum output length (Utpala et al., 2023). Early EOS does not reduce its declared privacy budget.

Caption+context privatizes an image-only caption together with the complete hint under the full request budget. Hybrids instead privatize the complete hint and image separately. Visual-token LDP (TOKEN in configuration labels) clips the entire centered visual-token block to a public ℓ_2 ball and adds Gaussian noise. DP-Pix applies whole-image-calibrated Laplace noise to RGB block means, clamps the result, and upsamples before visual encoding (Fan, 2018). These six adaptations are evaluated on ScienceQA/InternVL with transferred text mechanisms, while Ours is task-trained. Public-only omits private inputs; Raw input uses unmodified inputs, clean-codec bypasses clipping/noise, and clip-only omits noise. Raw input, clean-codec, and clip-only are unprotected references.

C.3 Whole-Request Budgets

The main representation curves use $\rho_0 \in \{30, 100, 300, 750, 1500, 3000, 10000, 30000, 100000\}$, converted by Equation (7) at $\delta = 10^{-6}$. Primary comparisons use $\rho_0 = 750, 3000, 10000$. In the main baseline comparisons, text-only uses full ϵ; hybrids compose $\epsilon_v = 0.25\epsilon$ and $\epsilon_t = 0.75\epsilon$, with development-selected choices fixed across budgets and shared by both text mechanisms. DP-Prompt text and DP-Pix vision are pure DP. DP-Prompt+TOKEN uses $\delta_v = 10^{-6}$; DP-BART+TOKEN uses

$\delta_t = \delta_v = 5\times10^{-7}$. DP-BART+DP-Pix retains $\delta_t = 5\times10^{-7}$; text-only DP-BART uses $\delta_t = 10^{-6}$. These meet a common $(\epsilon, 10^{-6})$ bound without identical privacy profiles. The low-budget extension sets $\epsilon \in \{0.1, 0.2, 0.5, 1, 2, 4, 8, 16, 32\}$ and obtains ρ_0 by inverting Equation (7). The separately evaluated expanded-budget hybrids instead allocate $\epsilon_v = 0.75\epsilon$ and $\epsilon_t = 0.25\epsilon$. Their results therefore do not extend the original hybrid curves under an unchanged allocation (Appendix D.4.2).

C.4 Scoring

ScienceQA selects the most probable candidate letter over all cyclic option orderings, reusing the same release with canonically ordered public option contents. We average orderings and three noise repetitions within each question, then weight questions equally; deterministic controls use one evaluation. WebQA uses greedy answers of at most 32 new tokens and frozen, reference-derived keyword exact match, not the official leaderboard metric. Annotations were model-assisted and reviewed without inspecting codec predictions; identical scoring rules apply to all methods. Its three repetitions are also averaged within questions before equal-question aggregation.

C.5 Statistical Protocol

Pointwise 95% percentile bootstrap intervals use 5,000 exact-RGB cluster resamples for ScienceQA and 10,000 question resamples for WebQA. Cluster draws retain all member questions and normalize by sampled question count. Paired contrasts use shared draws and intervals of score differences. Intervals exclude training-seed variation and are unadjusted for multiple comparisons or configuration selection.

C.6 Client-Side Resource Costs

We profile the ScienceQA/InternVL3.5-8B client, including the pretrained front end, learned compression, joint perturbation, and local decoding. The cloud transformer and language-model head are excluded. The pretrained front end includes the visual encoder, projector, and full input embedding table. The added modules contain 18.52M parameters, equivalent to 1.93% of this front end, or 5.49% of the visual encoder and projector alone.

Table 4: Client-side parameter counts and resource measurements for the profiled ScienceQA/InternVL configuration. Memory and latency refer to the complete client; latency is measured on a shared GPU.

Metric	Value
Pretrained front-end parameters (M)	959.913
Additional module parameters (M)	18.517
Complete-client parameters (M)	978.430
Peak PyTorch-allocated GPU memory (GiB)	1.903
Mean latency (ms)	76.75
Median latency (ms)	62.13
95th-percentile latency (ms)	159.14

Measurements use an NVIDIA RTX PRO 6000 Blackwell Max-Q Workstation Edition GPU with PyTorch 2.9.1 and CUDA 12.8. The pretrained modules use BF16 and the learned compression and decoding modules use FP32. We use batch size one and a development example selected to represent the median input length, containing one 448×448 image tile, 33 private contextual-text tokens, and 41 public question-and-option tokens. The configuration uses eight visual spatial coefficients, 512 text latent channels, and $(\epsilon, \delta) = (16, 10^{-6})$.

After 20 warm-up calls, we collect three groups of 100 synchronized wall-clock measurements. Inputs already reside on the GPU, so timing excludes model loading, disk access, CPU preprocessing, tokenization, host-to-device transfer, network communication, and cloud answer generation. Peak memory includes resident client tensors and temporary activations tracked by the PyTorch allocator, excluding the CUDA context and other processes. Because the GPU is shared with other workloads, the latency values are reference measurements for this configuration.

D Additional Utility and Component Results

D.1 Adapted Text-Private Baselines

Table 5 reports all six multimodal adaptations under the original matched whole-request budgets. The best adaptation remains close to Public-only, while the full framework retains higher task utility at each budget. Neither hybrid route consistently improves over Caption+context.

Table 5: **Accuracy (%) of adapted text-private baselines on ScienceQA with InternVL3.5-8B.** Columns give a common whole-request (ϵ, δ)-LDP upper bound with $\delta = 10^{-6}$.

Method	$\epsilon \approx 954$	$\epsilon \approx 3407$	$\epsilon \approx 10743$
DP-Prompt (caption+context)	64.22	64.39	63.93
DP-Prompt + Visual-token LDP	64.07	61.62	59.65
DP-Prompt + DP-Pix	63.68	63.85	64.11
DP-BART (caption+context)	65.01	64.26	63.90
DP-BART + Visual-token LDP	64.26	61.90	57.85
DP-BART + DP-Pix	64.73	63.90	64.01
Ours	**73.01**	**77.23**	**79.36**

The hybrids allocate 25% of ϵ to vision and 75% to text. Public-only accuracy is 63.51%. Implementation details appear in Appendix C.2.

D.2 Component Substitutions in the Main Checkpoint

Table 6 reports four compressor–decoder combinations on ScienceQA with InternVL3.5-8B, for both the full evaluation and the fixed 427-question subset defined in Appendix C.1. PCA denotes a fixed projection or reconstruction map, Learned denotes the task-trained question-aware compressor, and Refined denotes the complete learned decoder with residual refinement.

All four substitutions retain the main checkpoint's dimensions, centers, scales, and Joint mechanism, without retraining or recalibration. PCA/Refined uses the decoder trained with learned latents; PCA/PCA differs from the independently calibrated PCA-based main-curve baseline.

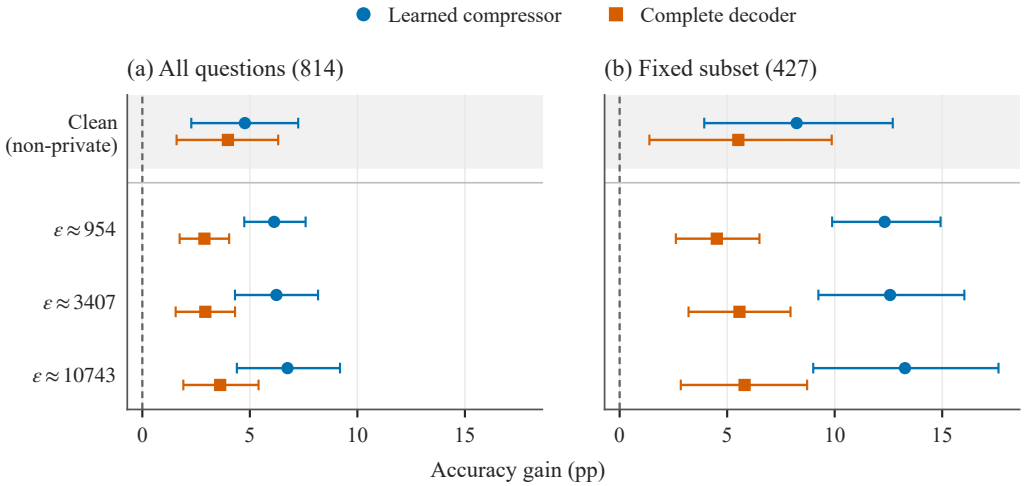


Figure 6: Paired component gains on ScienceQA/InternVL3.5-8B. Learned compressor compares Learned/Refined with PCA/Refined; complete decoder compares Learned/Refined with Learned/PCA. Bars show pointwise paired 95% image-cluster bootstrap intervals. Shading separates Clean, which bypasses clipping and noise, from private releases with $\delta = 10^{-6}$. The fixed subset follows Appendix C.1.

Table 6: Accuracy (%) of compressor/decoder combinations on ScienceQA with InternVL3.5-8B. The full set and fixed subset use identical component substitutions and release parameters. Clean bypasses clipping and noise; private columns use $\delta = 10^{-6}$.

Compressor / decoder	Clean	$\epsilon \approx 954$	$\epsilon \approx 3407$	$\epsilon \approx 10743$
(a) Full evaluation: 814 questions, 737 image clusters				
PCA / PCA	74.86	65.16	66.66	65.88
PCA / Refined	80.84	66.89	71.00	72.61
Learned / PCA	81.63	70.13	74.30	75.74
Learned / Refined	85.61	73.01	77.23	79.36
(b) Fixed subset: 427 questions, 403 image clusters				
PCA / PCA	56.25	38.72	42.03	41.30
PCA / Refined	67.21	41.19	49.69	52.65
Learned / PCA	69.92	48.99	56.69	60.11
Learned / Refined	75.44	53.51	62.26	65.92

All substitutions use the full framework's fixed axes without retraining.

Release geometry. Figure 7 adds the fixed 427-question subset to the main mechanism comparison. Joint improves over ISO by 2.30 points and MOD by 0.84 points at $\epsilon \approx 3{,}407$; the fixed-subset gains are 5.68 and 1.74 points. At $\epsilon \approx 10{,}743$, the overall paired interval for Joint versus ISO includes zero. ISO shares Joint's clipped mean, whereas MOD changes clipping and budget allocation. All mechanisms use the Joint-trained decoder with their respective noise scales.

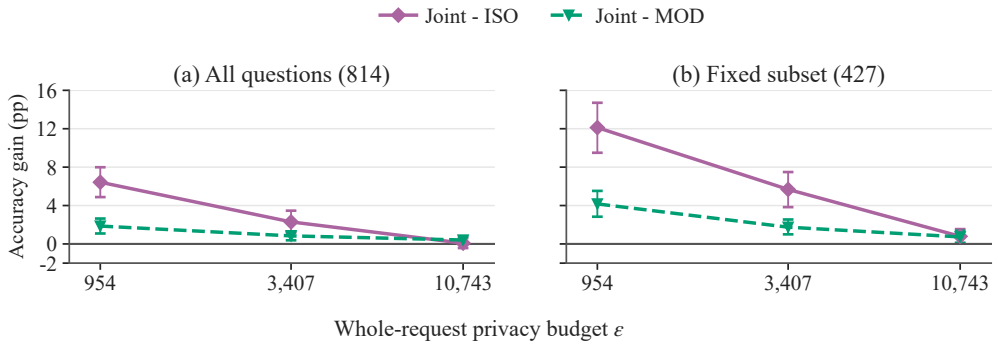


Figure 7: Paired Joint accuracy gains on ScienceQA/InternVL ($\delta = 10^{-6}$), with frozen Joint-trained modules, centers, and scales. Panel (a) reproduces Figure 3(b); panel (b) adds the fixed subset from Appendix C.1. Whiskers show pointwise 95% paired bootstrap intervals and the horizontal line marks zero gain.

D.3 Matched Training for Question Conditioning and Refinement

This independently retrained study compares encoder question conditioning and subsequent refinement on the full evaluation and fixed 427-question subset (Figure 3a); its checkpoints differ from the main study's.

Shared initialization and matched training. We initialize a common question-independent compressor and base decoder from public PCA and native representation statistics, without loading a previously task-trained checkpoint. We jointly train the compressor and base decoder for 2,048 clean updates, including the text compression maps and compressor mixer. The question-conditioned branches remain frozen at zero output during this stage. Training uses equally weighted candidate-answer negative log-likelihood, without reconstruction, distillation, or attention auxiliary losses.

We fork this common checkpoint into Q-on and Q-off arms with the same architecture and initial weights, including identically initialized conditioning branches. Q-on receives the public question and canonically ordered, unlabeled options; Q-off receives a fixed single-token embedding, with

neither current-question content nor question-length information. Each arm receives 2,048 clean updates followed by 2,048 noisy updates under matched losses, trainable modules, release scales, training budgets, and checkpoint-selection rules. The frozen VLM receives the actual question in both arms, so Q-off is also task-trained.

We then freeze the selected Q-on compressor and base decoder and train residual refinement for 2,048 updates. Thus the Q-on versus Q-off comparison controls the conditioning input under matched training, while the refinement comparison evaluates adding a subsequent adaptation stage.

Table 7: Mean accuracy (%) in the independently retrained staged ablation. Private columns give rounded whole-request ϵ at $\delta = 10^{-6}$; Clean bypasses clipping and noise. These checkpoints differ from those used for the main results and Table 6. Bold indicates the highest mean within each evaluation set.

Variant	Clean	$\epsilon \approx 954$	$\epsilon \approx 3407$	$\epsilon \approx 10743$
(a) All questions (814)				
Q-off, no refinement	84.98	69.28	73.57	76.59
Q-on, no refinement	85.54	72.75	76.64	78.70
Q-on + refinement	**87.18**	**75.62**	**79.55**	**82.05**
(b) Fixed subset (427)				
Q-off, no refinement	75.83	49.48	57.90	63.39
Q-on, no refinement	76.51	55.15	62.38	65.81
Q-on + refinement	**79.20**	**59.99**	**68.66**	**71.86**

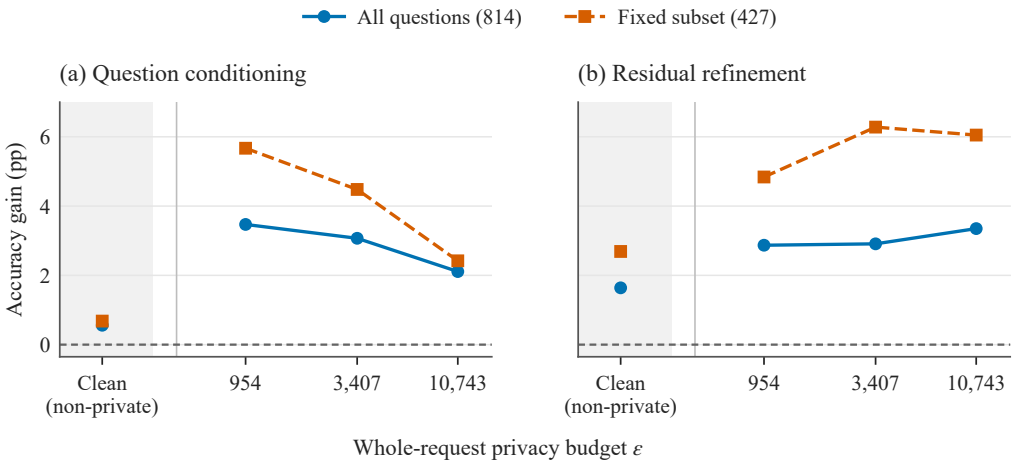


Figure 8: Question-conditioning and refinement gains in the independently retrained study, expanding Figure 3(a) to both evaluation sets. Panel (a) compares Q-on with Q-off, both without refinement; panel (b) adds refinement to the same Q-on base. Points are descriptive mean accuracy differences without paired confidence intervals. Clean bypasses clipping and noise; private releases use $\delta = 10^{-6}$.

Results. Without refinement, question conditioning adds 3.47, 3.07, and 2.11 percentage points across the three private budgets on all 814 questions, with corresponding gains of 5.67, 4.48, and 2.42 points on the fixed subset. Adding refinement to Q-on yields further gains of 2.87–3.35 points overall and 4.84–6.28 points on the subset. Both components provide larger gains under privatization than in the clean controls: clean gains are 0.56 and 0.68 points for question conditioning, and 1.64 and 2.69 points for refinement, on the full set and subset, respectively.

D.4 Utility beyond the Main Budget Grid

These additional evaluations examine both stronger privacy constraints and the budgets at which baseline pipelines recover higher utility on ScienceQA/InternVL. They retain the main study's 814 questions, 737 exact-RGB image clusters, three noise realizations, and cyclic-option scoring. The fixed-checkpoint extension retains the main study's pipelines, and the expanded-budget baseline comparisons use the main Ours checkpoint as their reference. The low-budget adaptation study additionally evaluates a separately fine-tuned checkpoint with public-statistics-based shrinkage.

D.4.1 Low-Budget Extension

Fixed-checkpoint extrapolation. We add nine budgets from $\epsilon = 0.1$ to 32 to the original nine-point curves, with $\delta = 10^{-6}$ throughout. All learned modules, centers, scales, and baseline reconstruction maps remain fixed; the added budgets lie outside Ours' training noise range. For the Gaussian releases, the numerically stable inverse of Equation (7) is

$$\rho_0 = \left(\frac{\epsilon}{\sqrt{\log(1/\delta) + \epsilon} + \sqrt{\log(1/\delta)}} \right)^2 . \tag{22}$$

Table 8 retains the reported pointwise 95% intervals from 5,000 RGB-cluster bootstrap resamples.

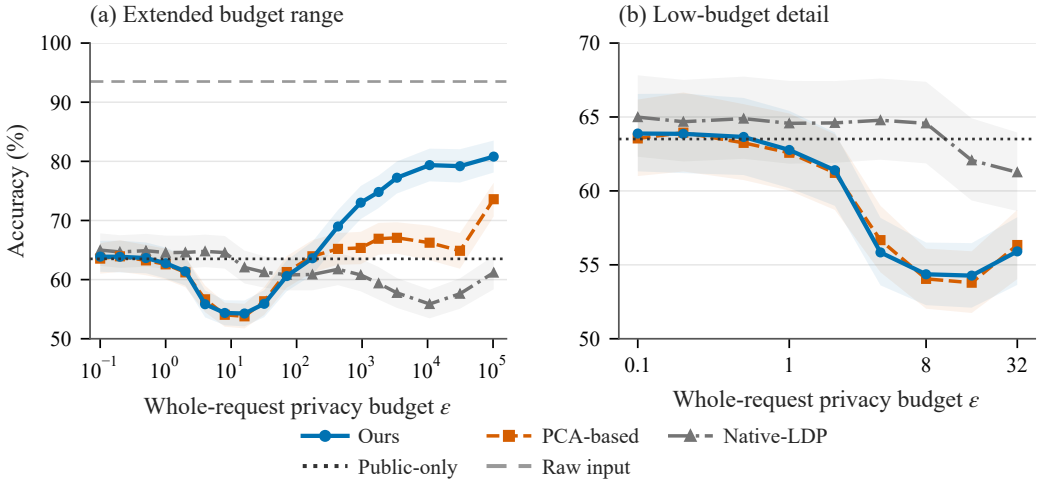


Figure 9: Extended privacy–utility curves of the original fixed pipelines on ScienceQA/InternVL. Panel (a) combines nine additional low-budget points with the original nine-point curves; panel (b) enlarges the low-budget region. Shading gives pointwise 95% image-cluster bootstrap intervals; lines connect measured points. The panels use different vertical ranges; Raw input (93.50%) lies outside the range of panel (b). Ours denotes the original main checkpoint; all plotted pipelines retain their main-study parameters across budgets, with $\delta = 10^{-6}$. The separately adapted low-budget variant is reported in Table 9.

At $\epsilon \leq 0.5$, the three pipelines attain 63.25–64.99% accuracy, near the Public-only reference. Ours and PCA-based release then exhibit similar troughs: at $\epsilon = 4$–32, their accuracies range from 53.79% to 56.67%, whereas Native-LDP attains 61.26–64.78%. Thus, the non-monotonicity is not unique to the learned decoder. Accuracy near Public-only does not establish that the pipelines produce the same answers or that private evidence contributes no information. These comparisons do not identify the cause of this behavior; LDP itself does not require fixed-model accuracy to vary monotonically with the budget.

Low-budget adaptation. We additionally evaluate a low-budget variant obtained by further training a previously trained codec with public-statistics-based shrinkage of its noisy latent. The learned

Table 8: Fixed-checkpoint accuracy (%) at the nine additional whole-request budgets, with pointwise 95% image-cluster bootstrap intervals. Ours denotes the original main checkpoint; the adapted variant is reported separately in Table 9. Public-only is 63.51%; Raw input is 93.50%.

ϵ	Ours	PCA-based	Native-LDP
0.1	63.88 [61.32, 66.56]	63.55 [61.00, 66.18]	64.99 [62.31, 67.82]
0.2	63.86 [61.20, 66.59]	63.94 [61.31, 66.66]	64.68 [61.99, 67.51]
0.5	63.65 [61.07, 66.29]	63.25 [60.73, 65.87]	64.89 [62.17, 67.73]
1	62.77 [60.19, 65.42]	62.58 [60.02, 65.24]	64.57 [61.84, 67.44]
2	61.40 [58.99, 63.95]	61.22 [58.71, 63.83]	64.60 [61.90, 67.43]
4	55.84 [53.61, 58.18]	56.67 [54.49, 59.00]	64.78 [62.11, 67.60]
8	54.36 [52.27, 56.51]	54.05 [52.03, 56.10]	64.57 [61.86, 67.38]
16	54.27 [52.10, 56.47]	53.79 [51.74, 55.92]	62.07 [59.35, 64.91]
32	55.91 [53.64, 58.22]	56.34 [54.03, 58.72]	61.26 [58.64, 63.95]

compressor, base decoder, and residual refiner are updated jointly at a single training budget, $\epsilon = 16$ and $\delta = 10^{-6}$, corresponding to $\rho_0 \approx 3.03957$. For modality $g \in \{v, t\}$, let $\widetilde{u}_g$ be the corresponding block of the Joint Gaussian release and σ_g its Gaussian noise standard deviation. We apply Wiener-style scalar shrinkage before decoding:

$$u_g^{\mathrm{sh}} = a_g \widetilde{u}_g, \qquad a_g = \frac{r_g^2}{r_g^2 + \sigma_g^2}. \tag{23}$$

Here, r_g^2 is the fixed coordinate-weighted second moment of clipped, pre-noise latents from historical public auxiliary data, computed as the total squared norm divided by the total number of coordinates. These statistics remain fixed throughout adaptation, and each modality uses one scalar coefficient across all its positions and channels. Writing $u^{\mathrm{sh}} = (u_v^{\mathrm{sh}}, u_t^{\mathrm{sh}})$, the base decoder reconstructs tokens from u^{sh}, and the refiner reads both these reconstructions and u^{sh} together with the public question. The refiner retains the original σ_g as its noise condition, while the effective noise standard deviation in u_g^{sh} is $a_g \sigma_g$. Because shrinkage uses only the private release and fixed public information, it preserves the original LDP guarantee by post-processing.

All adapted results in Table 9 use one fixed checkpoint from a single training seed. The same checkpoint is evaluated at both lower and higher budgets without further parameter updates. Evaluation retains the 814 ScienceQA questions, 737 exact-RGB image clusters, three noise realizations, and cyclic-option scoring of the main study.

Table 9: Accuracy (%) of the main checkpoint and the separately adapted low-budget variant on ScienceQA/InternVL3.5-8B, with $\delta = 10^{-6}$. Brackets report pointwise 95% image-cluster bootstrap intervals for the adapted variant; the main-checkpoint means are included for comparison. Public-only accuracy is 63.51%.

Whole-request ϵ	Main checkpoint	Low-budget adaptation
8	54.36	67.54 [64.58, 70.44]
16	54.27	67.37 [64.42, 70.25]
32	55.91	67.44 [64.48, 70.34]
≈ 954	73.01	68.17 [65.32, 71.08]
≈ 3407	77.23	73.76 [71.10, 76.37]

At $\epsilon = 8, 16, 32$, paired gains over Public-only are 4.03, 3.86, and 3.93 percentage points, with 95% intervals $[2.67, 5.40]$, $[2.52, 5.22]$, and $[2.59, 5.26]$, respectively, obtained from 5,000 image-cluster bootstrap resamples. At $\epsilon \approx 954$ and 3407, the adapted variant instead falls below the main checkpoint by 4.84 and 3.47 points in mean accuracy. Together, these results support low-budget adaptation as a complementary configuration with a different utility profile across budgets.

D.4.2 Expanded-Budget Baseline Comparisons

We evaluate four larger-budget candidates against the main checkpoint's 77.23% accuracy at $\epsilon \approx 3{,}407$. Candidates were explored on 128 previously inspected test questions and then evaluated on all 814 questions; this expanded-budget comparison is exploratory. All four candidates are retained,

including those whose mean accuracy remains below the reference. Native-LDP and PCA-based release retain their original Joint mechanisms and scales. The hybrid candidates use the same visual-token and text mechanism families with a different allocation: 75% of the total ϵ to vision and 25% to text, compared with 25% and 75% in the main comparisons.

Table 10: Observed utility at expanded baseline budgets on ScienceQA/InternVL. All budgets are reported whole-request ϵ bounds at $\delta = 10^{-6}$. Accuracy entries give means; Δ is baseline minus Ours in percentage points, with paired 95% intervals. Budget ratios describe these evaluated points and are not minimum-budget estimates.

Method	Evaluated ϵ	Accuracy (%)	Budget ratio	Δ (pp)
Ours	3,407.17	77.23	1.00×	–
Native-LDP	567,915.92	76.02	166.68×	$-1.21\,[-4.46, +2.00]$
PCA-based	1.000235×10^9	74.96	293,567.84×	$-2.27\,[-5.86, +1.18]$
DP-BART + Visual-token LDP†	8,021,026.09	79.86	2,354.16×	$+2.63\,[-0.36, +5.57]$
DP-Prompt + Visual-token LDP†	3,175,497.11	76.10	932.00×	$-1.13\,[-4.22, +1.94]$

† Expanded-study allocation: $\epsilon_v = 0.75\epsilon$, $\epsilon_t = 0.25\epsilon$. These hybrid configurations are distinct from those in Table 5. Intervals are retained from this comparison's analysis and do not replace the original main-curve intervals.

Native-LDP and DP-Prompt with visual-token LDP reach 76.02% and 76.10% at approximately $167\times$ and $932\times$ our reference budget, while DP-BART with visual-token LDP exceeds the reference mean at approximately $2{,}354\times$ the budget. PCA-based release remains at 74.96% even at $\epsilon \approx 10^9$, close to its 74.80% clip-only accuracy; this is consistent with limitations of the fixed representation, rather than noise alone.

E Empirical Inversion Evaluation

Evaluation and observations. We evaluate fixed text and image recovery procedures on 814 ScienceQA questions with InternVL3.5-8B at $\epsilon \approx 954$, 3407, and 10743. For representation releases, each procedure reads one modality block of the uploaded tokens, without question conditioning or joint image–text recovery. Privacy matching and aggregation follow Appendices C.3 and C.5.

Text recovery. For representation methods, we recover each token by maximum cosine similarity to the public embedding table and decode the resulting sequence. For DP-BART and DP-Prompt, we instead score their released rewritten strings directly. Word F1 measures lexical recovery against the original hint after case folding and whitespace normalization, using multiset overlap of \w+ tokens and assigning one when both strings are empty. The two recovery routes use different observation interfaces.

Fixed image inverse. ImageInverse uses public auxiliary native-feature/image pairs, with 960 training and 64 validation images and normalization fitted only on training features. It projects visual-token channels onto a spatial grid and reconstructs RGB through convolutional upsampling blocks. The model selected by validation loss and its normalization are fixed across defenses, without adaptation to defended representations. Its 224×224 output is bilinearly resized to the 448×448 single-patch model view for scoring. SSIM uses RGB in $[0, 1]$ with data range one; SqueezeNet LPIPS uses inputs rescaled to $[-1, 1]$. Lower SSIM and higher LPIPS indicate less recovery. The attack is trained with an ℓ_1 and LPIPS objective, so LPIPS is interpreted alongside SSIM.

Interpretation. At $\epsilon \approx 3{,}407$, Ours increases image LPIPS from 0.2998 for unprotected native tokens to 0.6267 while retaining 77.23% QA accuracy (Table 11). The text-rewrite hybrids yield lower lexical overlap but also lower task accuracy. At the two smaller budgets, Ours has lower LPIPS than PCA-based despite lower SSIM (Figure 10), so the recovery ranking depends on the metric. Visual-token LDP hybrids retain their distinct visual δ allocations; the two DP-Pix hybrids share the same private visual release. These results characterize the specified fixed procedures;

Table 11: Task utility and reconstruction on ScienceQA with InternVL3.5-8B. Private methods use matched whole-request bounds ($\epsilon \approx 3407$, $\delta \leq 10^{-6}$); the unprotected reference has no finite privacy guarantee.

Method	QA ↑	Word F1 ↓	SSIM ↓	LPIPS ↑
Non-private				
Unprotected native	93.50	100.00	0.6428	0.2998
Private releases				
Native-LDP	57.76	6.37	0.2377	0.7879
PCA-based	67.07	46.18	0.5303	0.6318
DP-Prompt + Visual-token LDP	61.62	0.24	0.2349	0.7668
DP-BART + Visual-token LDP	61.90	2.12	0.2349	0.7666
DP-Prompt + DP-Pix	63.85	0.24	0.5261	0.6995
DP-BART + DP-Pix	63.90	2.12	0.5261	0.6995
Ours	77.23	14.85	0.5041	0.6267

QA and Word F1 are percentages; arrows denote the defense perspective. Unprotected native releases native encoded tokens without compression, clipping, or noise. Image scores use the fixed token-input inverse. Text scores compare cosine vocabulary recovery with directly observed baseline rewrites, whose observation interfaces differ as described in the text.

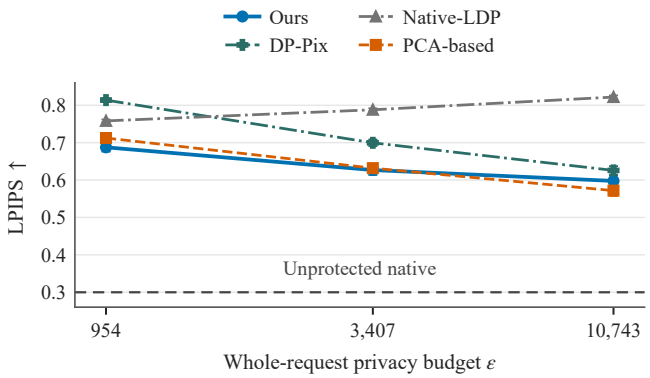


Figure 10: Image-inversion LPIPS on ScienceQA/InternVL3.5-8B, complementing the SSIM results in Figure 4. Higher LPIPS indicates less recovery under the fixed token-input inverse. Bars show pointwise 95% image-cluster bootstrap intervals; the dashed line is unprotected native tokens. DP-Pix retains its original hybrid budget allocation; the horizontal axis is the complete-request bound with $\delta \leq 10^{-6}$.

compression and decoding may also introduce distribution mismatch for the native-trained image inverse.