PRIVATAR: ENABLING PRIVACY-PRESERVING REAL-TIME MULTI-USER VR THROUGH SECURE OFFLOADING

Jianming Tong [1 2 3] **Hanshen Xiao** [4] **Krishnakumar Nair** [3] **Hao Kang** [1] **Ziqi Zhang** [5] **Ashish Sirasao** [6] **G. Edward Suh** [7] **Tushar Krishna** [1]

ABSTRACT

Multi-user virtual reality (VR) applications such as football and concert experiences rely on real-time avatar reconstruction to enable immersive interaction. However, rendering avatars for numerous participants on each headset incurs prohibitive computational overhead, fundamentally limiting scalability. This work introduces a framework, Privatar, to offload avatar reconstruction from headset to untrusted devices within the same local network while safeguarding sensitive facial features against adversaries capable of intercepting offloaded data.

Privatar builds on the insight that "domain-specific knowledge of avatar reconstruction enables provably private offloading at minimal cost". (1) *System level*. We observe avatar reconstruction is frequency-domain decomposable via block-wise DCT with negligible quality drop, and propose Horizontal Partitioning (HP) to keep high-energy frequency components on-device and offloads only low-energy components. HP offloads local computation while reducing information leakage to low-energy subsets only. (2) *Privacy level*. For *individually* offloaded, *multi-dimensional* signals without aggregation, worst-case local Differential Privacy requires prohibitive noise, ruining utility. We observe users' expression statistical distribution are *slowly changing over time and trackable online*, and hence propose Distribution-Aware Minimal Perturbation (DAMP). DAMP minimizes noise based on each user's expression distribution to significantly reduce its effects on utility and accuracy, retaining formal privacy guarantee. Combined, HP provides empirical privacy protection against expression identification attack. And DAMP further augments it to offer a formal guarantee against arbitrary adversaries.

On a Meta Quest Pro, Privatar supports up to 2.37× more concurrent users at 5.7∼6.5% higher reconstruction loss and ∼9% energy overhead, providing a better throughout-loss Pareto frontier over SotA quantization, sparsity, and local reconstruction baseline. PRIVATAR further provides both provable privacy guarantee and stays robust against both empirical attack and NN-based Expression Identification Attack, proving its resilience in practice. Our code is open-sourced at https://github.com/georgia-tech-synergy-lab/Privatar.

1 INTRODUCTION

Virtual Reality (VR) is rapidly evolving to support shared, immersive 3D environments for applications ranging from concerts (Park et al., 2024), sports games (Huang et al., 2025), cinemas (Kim et al., 2024) to collaborative design(Zhou et al., 2025). Central to this evolution is the ability to render photorealistic avatars of *multiple users* in real-time, enhancing social interaction.

An ideal multi-user VR system must achieve four competing goals: i) high fidelity, ii) strong user privacy, iii) power and heat efficiency for better user experience, and iv) scalability to multiple users. In current paradigm, avatar reconstructions of all users happen locally within a single VR headset, which satisfies i)-iii) but fails to scale as the rendering workload from multiple users quickly overwhelms the limited computational resources of mobile devices. This paper explores an alternative of offloading local computation to a powerful but untrusted external device for throughput improvement. However, offloading introduces privacy leakage, as it exposes private user facial input individually to the local network, potentially leading to confidentiality and integrity risks. An adversary on the same local network (communication) or residing at other devices (compute) can eavesdrop on, infer from, or tamper with offloaded data.

[1]Georgia Institute of Technology, Atlanta, Georgia, USA [2]Massachusetts Institute of Technology, Cambridge, Massachusetts, USA [3]Google, Mountain View, California, USA [4]Purdue University/NVIDIA, West Lafayette, Indiana, USA [5]University of Illinois Urbana-Champaign, Champaign, Illinois, USA [6]AMD, Santa Clara, California, USA [7]Cornell University/NVIDIA, Westford, Massachusetts, USA. Correspondence to: Jianming Tong <jianming.tong@gatech.edu>.

Proceedings of the 7th MLSys Conference, Bellevue, WA, USA, 2026. Copyright 2026 by the author(s).

1.1 Technical Challenges

Broadly speaking, there are three lines of provable privacy protection for offloading in untrusted servers: cryptographically based encryption, trusted hardware based isolation, and information-theoretical based perturbation. But they all fall short in either efficiency or utility (quality of avatar).

Cryptographic solutions enable a user to encrypt their personal VR data without leakage, and the server to perform computation over ciphertexts without decryption, which perfectly satisfies i) accuracy and ii) privacy. However, SotA solutions (Tong et al., 2026a) cannot satisfy iii) efficiency and iv) scalability requirements listed above. For example, Homomorphic Encryption (HE) encrypts offloaded data and its computation, leading to 3 magnitude higher computation latency (Tong et al., 2026a), hardly supporting more users. Multiple Party Computation (MPC) leverages more than one non-colluding servers to collaboratively compute the private data, but requiring up-to Giga-Bytes of communication overhead to forbid real-time processing (Keller & Sun, 2022; Knott et al., 2021).

Confidential computing, using technologies like Trusted Execution Environments (TEEs) in CPU Intel SGX (Intel Corporation, 2014) and AMD SEV-SNP or SME (Advanced Micro Devices, Inc., 2020), offers sufficient privacy guarantee and somewhat throughput improvement, (1.79× in our experiments) because of slow throughput on CPU and extra encryption overhead.

Information-theoretical privacy preservation largely relies on adding noise to obfuscate release. Compared to cryptographic encryption or confidential computing, noise does not cause any efficiency loss but it hurts utility and introduces privacy risk. In this line, Differential Privacy (DP) (Dwork et al., 2006) is one of the most representative frameworks. DP aims to ensure that, in the worst case for an adversary with arbitrary prior belief, they cannot gain additional knowledge from the release to distinguish an individual's participation in a data processing. Operationally, DP requires first determining an *upper bound* of the *sensitivity* of the processing function (Xiao et al., 2023b), i.e., the worst-case change on the release when one arbitrarily replaces one datapoint. Afterwards, noise is calibrated according to both sensitivity and security parameters (Xiao et al., 2023a). However, depending on the application, finding a usable noise solution without significantly hurts utility is non-trivial (Xiong et al., 2020; Near et al., 2025).

1.2 Our Contributions

We introduce PRIVATAR, the first framework to enable privacy-preserving offloading without sever utility degradation, ensuring private, high-fidelity, and real-time multi-user avatar reconstruction at scale. We systemically protect privacy of individually released facial data in two levels.

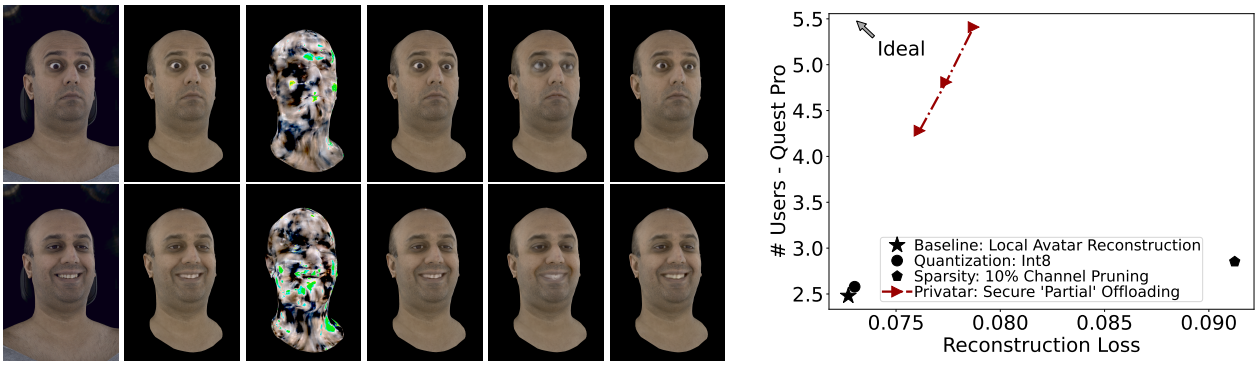
On the **System Level**, PRIVATAR's core innovation is a strategic partitioning of the avatar reconstruction pipeline. It separates the frequency representation of an avatar's texture, assigning different components to either the local VR headset or an untrusted external device based on variance. The facial mesh and the high-variance frequency components of the texture are processed securely on the local VR headset. Conversely, the reconstruction of the low-variance frequency components is offloaded to untrusted third-party devices to minimize the noise required by provable privacy protection. *This physical isolation reduces the sensitivity of offloaded data* by only exposing a subset of components, providing empirical protection against expression identification attack (Kopalidis et al., 2024; Zhang et al., 2023).

On the **Privacy Level**, PRIVATAR roots in using exploitation of data entropy for privacy protection. Compared to DP, which fully relies on algorithmic randomness to ensure indistinguishability, we further consider incorporating the data entropy for privacy enhancement. That is to say, we consider formalizing the adversarial inference hardness based on the randomness from both the data distribution and the additional noise perturbation instead of purely on noise perturbation in DP. Based on this, we propose Distribution-Aware Minimal Perturbation to reduce the amount of noise required by local DP by up-to 17.6× for the same level of provable privacy guarantee, reducing utility degradation.

Intuitively, PRIVATAR leverages the temporal invariant nature of users' facial expression to reduce the required noise, i.e. users turn to make the similar expressions as they did recently (analogue to user's quirk) and all expressions a user could make cannot change drastically over time. Using users' historical expression statistics and PAC Privacy (Xiao & Devadas, 2023; 2025; Xiao, 2024), we compute per-dimension, provably minimal perturbations that meet a target privacy guarantee. Our contributions are:

• Horizontal Partitioning (HP): Horizontally split of the avatar reconstruction into two independent paths for locally processing on device and offloading to untrusted devices, and a frequency-domain split of unwrapped facial textures that enables fine-grained data/compute relocation among two paths. This reduces local computation while only restricts the untrusted devices to see an incomplete view of the private data for privacy protection, improving scalability while maintaining privacy.

• Distribution-Aware Minimal Perturbation (DAMP): A dimension-wise noise determination mechanism that exploits the entropy of private data (statistical distributions) into minimizing perturbation (noise) needed to achieve provable privacy protection. DAMP calibrates multi-dimensional noises for individually released data via PAC privacy, yielding markedly better reconstruction quality (less reconstruction loss) than local DP at the same privacy level.



(a) Truth (b) Baseline (c) FO (d) Quantize (e) Sparsity (f) PRIVATAR (g) Latency–accuracy tradeoff in all settings.

Figure 1. **A visual and quantitative comparison among efficient avatar reconstruction techniques**. **PRIVATAR (Secure Partial Offloading, 1f)** achieves 2.37× throughput (#users per second) than SotAs (1b∼1e) under similar quality (loss). FO: Fully Offloading.

• PRIVATAR: Combined, HP reduce offloaded information to an incompleted subset, reducing noises needed by DAMP, and DAMP further augments the empirical privacy protection of HP to a formal guarantee against arbitrary adversaries. Our evaluation on commercial VR hardware (Meta Quest Pro) shows that PRIVATAR supports up-to 3 more users (2.37×) with negligible 5.7% accuracy degradation and only a minor (9%) increase in energy consumption, as shown in Fig. 1. PRIVATAR enables a better throughput-loss Pareto frontier over quantization, sparsity, and local processing baseline, and is empirically robust against NN-based expression identification attacks, and provable robust against arbitrary block-box attacks.

2 BACKGROUND

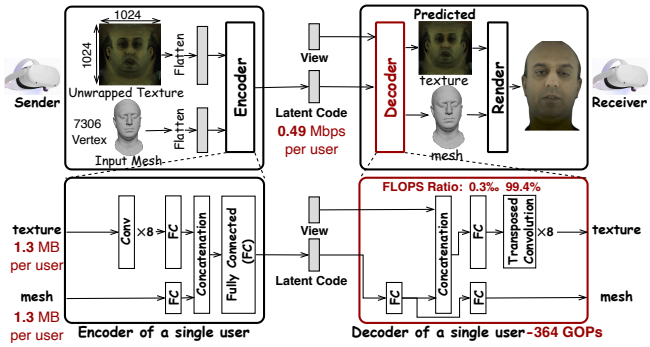


Figure 2. **Overview of the Variational Auto-Encoder (VAE) architecture used for avatar reconstruction.** ***Takeaway:*** The texture reconstruction stage (transposed convolution) accounts for 99.4% of the decoder's FLOPs, forming the primary computational bottleneck that PRIVATAR mitigates through partial offloading.

2.1 Facial Avatar Reconstruction Pipeline

An avatar needs two key data: a high-resolution unwrapped texture (the "facial skin", ∼1.3 MB) and a detailed facial mesh (the 3D model structure, ∼1.3 MB) in Fig. 2. Transmitting raw data for every user at a smooth 60 frames per second (FPS) is impractical. It would require ∼1.25 Gbps per user, which far exceeds the limited and unstable bandwidth of internet connections, which can be as low as 200 Mbps, as shown in Fig. 3a. To reduce bandwidth, modern VR systems use a Variational Auto-Encoder (VAE) (Lombardi et al., 2018; Ma et al., 2021).

• On the sender's end, an **encoder** compresses texture and mesh into a compact "*latent code*". This reduces required bandwidth by over 99%, to just 0.49 Mbps per user, making transmission over limited-bandwidth submarine cable feasible, as shown in Fig. 3a.

• On the receiver's end, a **decoder** receives this compact latent code and decompresses it to reconstruct the original high-resolution texture and mesh for avatar rendering.

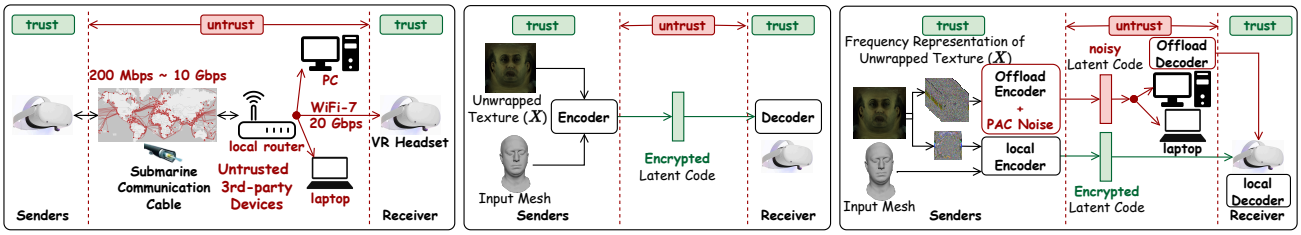
2.2 Compute Bottleneck in Multi-user Reconstruction

While VAE solves the bandwidth problem, it introduces a new one: a severe computational bottleneck on the receiver's headset. In a multi-user session, the headset must decode the latent code from all senders simultaneously.

The computation demand of decoder quickly overwhelms the limited computational budget of a standalone VR headset, restricting the number of users that can be supported. For instance, a commercial headset like the Meta Quest Pro has a computational capacity of 902 GFLOPS, which supports a maximum of *two* users at 60 FPS before losing FPS. The bottleneck is the decoder of the pipeline. Within the decoder, a series of eight transposed convolution layers responsible for reconstructing the unwrapped texture accounts for 99.4% of the total computation, as shown in Fig. 2. In contrast, reconstructing the facial mesh is computationally trivial, contributing only 0.6% overall FLOPS of a decoder.

2.3 Vulnerability of Facial Avatars

Facial avatars, built from textures and expression-driven meshes, introduce additional risks of identity leakage (via facial recognition or soft-biometric profiling) and expression leakage (revealing emotion or cognitive state).



(a) Setup and Threat Model (b) Baseline: Local Reconstruction (c) Privatar (Local + Offloading)

Figure 3. **Overview of PRIVATAR against baseline under realistic threat model. (3a) Setup and Threat Model:** In a multi-user VR session, each user only trust its own VR headset. **(3b) Baseline** processes **entire decoder** locally on receiver headset. It ensures privacy but the limited compute on device is not sufficient to support many users. **(3c) Privatar (This work)**: Our framework introduces Horizontal Partitioning, which splits the reconstruction into two paths and offloaded one path to untrusted PC to reduce local computation. Local Path (Green): The relatively privacy-sensitive data, such as the facial mesh and the highly identifying frequency components of the texture, are processed securely on the trusted receiver headset. Offloaded Path (Red): Less-sensitive, but computationally intensive, frequency components are perturbed with minimal, precisely calibrated noise and offloaded to untrusted devices. ***Key Innovation:*** Privatar prevents the untrusted device from ever accessing the complete private facial data. By only processing a partial and obfuscated frequency components, it cannot reconstruct user's sensitive expression, achieving both scalability and privacy without sacrificing quality.

Identity Leakage: The facial texture carries rich biometric information that uniquely identifies individuals, similar to a face image in traditional vision datasets. Leveraging facial textures, adversaries can recover identity through facial recognition models or soft-biometric profiling, linking avatars back to real-world users (Nair et al., 2023). In large-scale multi-user environments, this enables cross-session tracking and deanonymization attacks.

Expression and Emotion Leakage: Facial meshes encode temporally dynamic expression parameters, such as smiles, frowns, and subtle muscle movements, which can reveal a user's mood, cognitive state, or reactions to specific stimuli. Studies show that such motion telemetry can leak private emotional or health-related information, even when identity is obfuscated (Nguyen et al., 2024).

In summary, reducing local computational overhead of decoder is the critical challenge for enabling VR experiences with a higher number of simultaneous users. This work proposes a *secure partial offloading mechanism to achieve this without compromising user privacy or visual quality.*

3 CHALLENGES AND KEY INSIGHTS

This section presents multi-user VR setup in Fig. 3a, defines corresponding threat model, outlines two key challenges in secure offloading, and highlights two observations that guide Privatar's innovations on *what to offload* and *how to add noise to offloaded data for privacy protection*.

3.1 Threat Model of Reconstruction Offloading

Setup: In multi-user VR session, each participant wears a headset located at a different physical site (e.g., user's homes or offices). Headsets communicate with one another through the Internet via a local router, which represents the household or institutional network gateway connecting the headset to external infrastructure and nearby personal devices (e.g., PC, laptop). All traffic leaving the headset traverses this local router and wide-area links (e.g., submarine communication cables), exposing the data to untrusted local and network-side devices that can intercept or analyze offloaded information, as shown in Fig. 3a.

Trust Boundary: We aim to provide provable and strong privacy definitions similar to local DP (Xiong et al., 2020), where a user only trusts its own local headset, and may not trust any other devices even under the same local network, and will randomize his private data locally to ensure privacy before releasing, which leads to:

- **Untrusted network** and hence any third-party device connected to network, including the router, laptops and PCs. Only VR headsets involved are trusted.
- **Untrusted communication channel** between any device and receiver VR headsets, as shown in Fig. 3a.

Protected Asset: facial expressions are primary assets requiring protection. A leakage can reveal personal identification (Carr et al., 2023) and motion (Yang et al.).

Adversary: We consider a computationally-unbounded adversary who can observe users' release and has *full knowledge* on the following: 1) encoding and noise mechanism applied by the user, 2) the underlying distribution of each user's facial data. That is to say, the only thing the adversary does not know is the randomness in both user's facial data generation and encoding and perturbation mechanisms. For the formal, provable privacy analysis (§4.2.3), PRIVATAR ensures protection of any selected private asset (e.g., identity, expression etc.) from user's release against an arbitrary adversary attempting to extract that asset. To complement this formal guarantee with concrete empirical validations, we instantiate two adversaries as "expression identification

attacker", whose goal is to correctly infer the user's facial expression (Fig. 9, detailed in §D). This represents real-world attack to demonstrate the effectiveness of PRIVATAR.

3.2 Challenges in Reconstruction Offloading

Offloading reconstruction requires sending latent codes of users and decoder from the receiver VR headset to untrusted devices. This involves exposing sensitive user facial data and introduces privacy concerns. Privacy protection requires increasing the ambiguity of offloaded data, i.e. force different offloaded data to appear similar, thus making it difficult for adversaries to distinguish between them.

Arguably, noise is the most popular way of randomizing data to increase ambiguity. For individually released latent code, the SotA local Differential Privacy (DP)-based randomization solutions (Croft et al., 2021; Fan, 2019; Zhao & Chen, 2022) mostly apply uniform noise to all dimensions of offloaded data, also called isotropic noise. However, adding DP-based noise perturbation into offloaded latent codes either suffers from privacy leakage or ruins reconstruction quality. Our empirical results show:

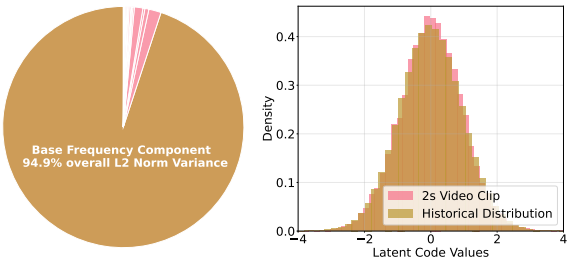
- With low noise, an ML attacker can identify expressions with 86.15% accuracy (Tab. 3), a clear privacy breach.
- With enough noise for privacy, the reconstruction quality is ruined, as shown in Fully Offloading (FO in Fig. 1), with a 105× increase in reconstruction loss compared to the baseline (local avatar reconstruction).

Such a flaw of imbalance between privacy and utility (quality) boils down to two key reasons.

• **Offloading the Entire Asset:** It is contradictory to make ***entire*** offloaded data to be ***ambiguous*** for high privacy protection and ***accurate*** for high reconstruction quality at the same time. High ambiguity requires high noises while high accuracy calls for low noises.

• **Prohibitive Isotropic Noise:** In avatar reconstruction, data are released individually without being aggregated. This invalidates typical DP for lack of a aggregation of users to hide the private data of a single one. The local-DP (Xiong et al., 2020) is the only viable solution, which typically uses isotropic noise, applying uniform level of randomization across all dimensions of the data. This noise is conservatively calibrated to worst-case sensitivity (the maximal possible change in data). As a result, dimensions with small values receive the same large amount of noise as the max-value dimension, causing the noise to overwhelm the actual signal and severely degrade reconstruction quality.

These two challenges make secure offloading a non-trivial open question, motivating PRIVATAR (Fig. 3c).



(a) L_2 Norm of Components (b) Distribution Comparison

Figure 4. **Illustration of two key observations**. ***Takeaway:*** (4a) The energy (L_2 norm) has extremely imbalanced distribution over frequency. Base frequency component contains 94.9% energy of entire data. (4b) Historical distribution (training dataset, §5) are similar to distribution of random 2 second clip of avatar trace.

3.3 Two Key PRIVATAR Observations and Insights

PRIVATAR balances the privacy and accuracy by addressing two failures with two core insights.

3.3.1 Insight 1: Imbalanced Frequency Distribution

We observe that entropy of private facial information has extreme imbalanced distribution over frequency, as illustrated in Fig. 4a. Specifically, after decomposing the facial unwrapped texture into a spectrum of sixteen frequency components, the L_2 norm[1] of the base frequency component contributes to 94.9% of that in the original unwrapped facial texture while these other 15 frequency components overall contribute to $\sim 5\%$, as shown in Fig. 4a. This indicates the base frequency component contains higher energy than the union of all remaining frequency components. Therefore, instead of offloading the entire unwrapped texture, Privatar only ***offloads a partial set of non-base components with less*** L_2 ***norm***, keeping the high-energy components exclusively on the trusted VR headset. Privacy is therefore achieved through the inherent difficulty of reconstructing a complete face from incomplete data, offering empirical privacy protection for the expression identification attack, detailed in §D.

3.3.2 Insight 2: User-Unique Slowly Drifting Distribution

To further provide a proved guarantee for arbitrary attacks, PRIVATAR adds noise to offloaded frequency components. Albeit the reduction of information from the offloaded *partial* frequency components, the noise required for provable privacy protection could still be prohibitive when following local DP-based noise determination. It is because noise of all dimensions has to be the same large to hide potential highest possible dimensional value to ensure privacy guarantee for arbitrary cases, i.e. *isotropic noise without consideration of the data distribution.* We observe that such

[1] L_2 norm (squared Euclidean length) often serves as a proxy for "energy" as it measures energy in an orthonormal basis.

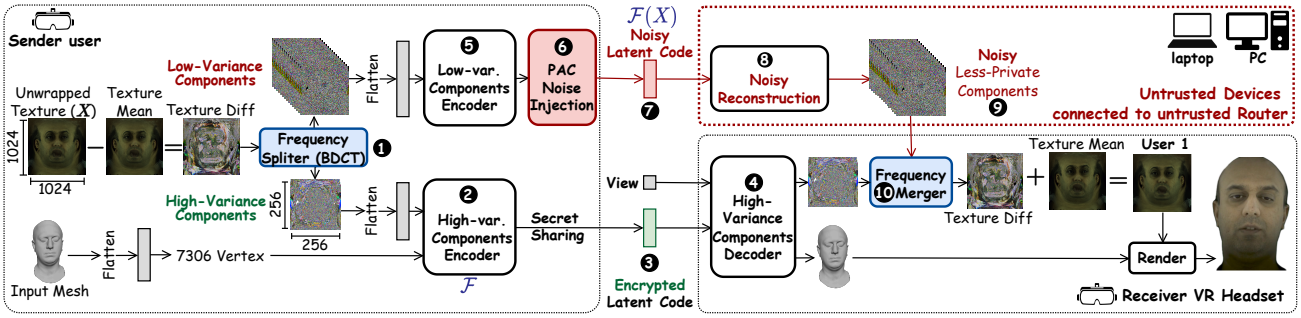


Figure 5. **Overview of PRIVATAR avatar reconstruction flow.** VR reconstruction is *horizontally partitioned* into two paths, running on the local device (bottom right) and untrusted devices (upper right red dashed box), separately. Noisy reconstruction is highlighted in red.

worst-case protection is over conservative for avatar reconstruction, as the distribution of each user's expression is slowly changing over time, i.e. being relatively static within a short duration, which could be used to reduce noises for dimensions with less entropy.

Intuitively, the extreme dynamite of expressions a person could make cannot drastically change from what he could make before, e.g. a person could not suddenly make his mouth to be 2× wider than the largest mouth that he could make before. Quantitatively, this is reflected by a slowly changing distribution. Specifically, for a randomly selected user, we show in Fig. 4b that the distribution of latent code for expressions in training dataset is *almost identical* to that for expressions of randomly selected 2 seconds (120 frames). This motivates the PRIVATAR's *Distribution-Aware Minimal Perturbation (DAMP)*, which ***leverages the latest statistical distribution of each user's expressions*** to apply ***non-uniform, minimum noise***, i.e. more for high-variance dimensions, less for low-variance ones. And DAMP keeps updating the statistical distribution with new expressions online to track distribution in the long run. This tailored approach provides a formal privacy guarantee while reducing required noise by up-to 17.6× compared to DP, preserving high reconstruction accuracy, as quantified later in Fig. 10c.

4 PRIVATAR

Goal. Enable secure offloading a subset of frequency components in private facial unwrapped texture to reduce computation overhead in the receiver VR headset, so that (i) the high-energy frequency components of texture and facial mesh never leave the trusted headset, (ii) low-energy components are safely offloaded with minimal noise perturbation.

4.1 Horizontal Partitioning (HP)

4.1.1 HP Overview

HP splits the encoder–decoder pair of VAE (Fig. 2) into two independent parallel paths: a *local path* runs on the trusted receiver headset and an *offloaded path* runs on untrusted devices under the same local network with receiver headset (Fig. 5). The private facial texture is split in frequency, reconstructed independently, then merged for rendering.

Local Path (❷→❸→❹) The encoder and decoder for selected frequency components run on trusted sender and receiver VR headset, separately, with the latent code to be encrypted during the communication for privacy protection. Encryption/decryption key are shared once ahead of actual transmission (Shamir, 1979), and runtime encryption and authentication lead to negligible overhead (Dworkin, 2007).

Offloaded Path (❺→❻→❼→❽→❾) The sender VR headset encodes the remaining components locally, injects distribution-aware noise (§4.2), and then offloads noisy latent codes for decoding to untrusted devices. Only *partial and obfuscated* view is ever exposed to reduce leakage.

4.1.2 Frequency Partitioning of Unwrapped Texture

Frequency Decomposition of Unwrapped Texture: Unwrapped facial texture is a 3 dimensional image in $\mathbb{R}^{H\times W\times 3}$. HP first subtracts it by the average texture in training dataset and applies a block Discrete Cosine Transform (DCT) (Ji et al., 2022; Strang, 1999) of size $B\times B$ to each non-overlapping block, yielding B^2 frequency components, each of shape $\frac{H}{B}\times\frac{W}{B}\times 3$. Each component is $\frac{1}{B^2}$ the size of unwrapped texture, so splitting across paths preserves the total size of data while enabling compute relocation.

Variance-Based Partitioning of Freq. Components: PRIVATAR uses the L_2 norm as a proxy of the energy contained in each frequency component to partition all components among local and offloaded paths.

Step 1: Given the goal of local computation reduction, PRIVATAR computes the number of components to offload.

Step 2: PRIVATAR selects required number of components with lower L_2 norm (variance) for offloading. Note that facial mesh is always kept local for privacy protection.

Differences to original VAE: Compared to Fig. 2, PRIVATAR reduces local compute and memory by: (1) down-

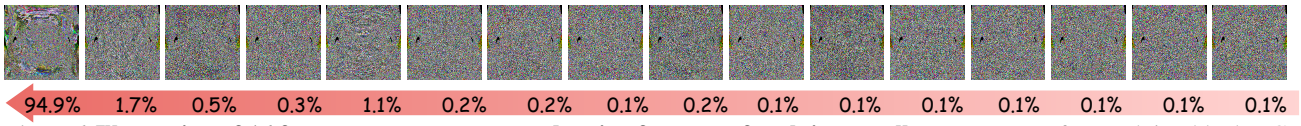


Figure 6. **Illustration of 16 frequency components and ratio of energy of each in overall components** after applying block DCT with $B = 4$ to the difference between input unwrapped texture and its average value in Fig. 5. Frequency increases from left to the right. Visually, all components look like random data, protecting the privacy of user expression information. The bottom bar shows profiled L_2 norm variance of each frequency component. The base frequency component has the highest L_2 norm (94.9% of overall L_2 norm), indicating a highest energy. Other frequency components have similarly small L_2 norm, much lower than that of the base frequency component. ***Takeaway:*** *the base frequency component carries the most energy. HP keeps it local and only offloads low-energy components to reduce noises needed for obfuscation, improving utility.* HP gives empirical protection as only subset instead of all components are offloaded, exposing a partial view of information, while formal privacy guarantee is obtained from DAMP through noise perturbation.

sampling the texture input to $\frac{1}{B}$ spatial resolution, which removes $log_2(B)$ layers from both the local encoder (❷) and decoder (❹). This lowers the per-component reconstruction cost to $\approx \frac{1}{B^2}$ of the original VAE, so reconstructing all B^2 components has comparable total cost; (2) offloading X of the B^2 components leaves only $\frac{B^2-X}{B^2}$ locally ($X! \in![2, B^2!-!2]$), reducing both local memory footprint and computation.

4.1.3 *Utility Effects of Horizontal Partitioning*

Partitioning has negligible utility (quality) effects. By partitioning B^2 components across two paths, PRIVATAR gives robust privacy protection against expression identification attack (Fig. 9). As a result, different choices of partitioning create the design space to trade off privacy, local compute latency, and communication latency.

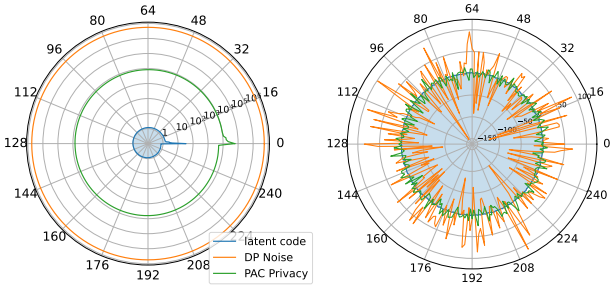
4.1.4 *Privacy Protection of Horizontal Partitioning*

HP only provides *empirical* protection. Achieving *provable* privacy for the *individually released user latent codes* still requires noise perturbation; however, HP reduces the required noise by exposing only a subset of texture information.

As shown in Fig. 6, each post-transform component appears noise-like, which suppresses direct expression cues from any single component or any incomplete subset. *Empirically*, HP splits private information across two paths so the offloaded path observes only an incomplete, low-energy view, mitigating expression identification attacks (Fig. 9). By limiting exposed content, HP also reduces identity leakage.

4.2 Distribution-Aware Min. Perturbation (DAMP)

Even with HP, providing *provable* privacy for *individually released user latent codes* can require prohibitive noise. We introduce DAMP, which reduces the required noise by leveraging the *statistical distribution* of the offloaded data, thereby minimizing utility loss while maintaining the same provable privacy level for any selected sensitive asset (e.g., identity, expression etc.).



(a) Statistical Visualization (b) Single Entry Visualization

Figure 7. **Statistical noise comparison between Differential Privacy based noise or PAC Privacy based noise in DAMP.** Each direction of the chart shows one dimension of the 256-dimensional latent code. ***Takeaway:*** Statistically, DAMP leverages the actual distribution of the latent code, blue in (7a), to reduce the trace of covariance (reflecting the amount of noise statistically) of DP based noise, orange in (7a), by 10^3 into PAC based noise, green in (7a). For each individual released latent code, PAC Privacy reduces the overall noisy value of DP (orange) by 17.6× into green curve in (7b), reflected by less fluctuation in the wave, significantly reducing accuracy degradation caused by noise.

4.2.1 *Key Idea*

Fig. 7 highlights the core problem: a fundamental mismatch between standard Differential Privacy (DP) and the data's distribution. DP typically injects *isotropic* noise (uniform variance in all dimensions), as shown by the orange ring in Fig. 7a. However, the actual distribution of the latent codes is highly *anisotropic* (non-uniform, concentrated in specific dimensions), shown in blue. This mismatch forces DP to add excessive, unnecessary noise in low-variance dimensions, inflating the total "noise energy" (covariance of generated noise) and severely degrading data utility. DAMP solves this by being distribution-aware. It is the first technique to leverage *PAC privacy* (Xiao & Devadas, 2023) for time-series facial data, enabling the generation of calibrated, ***anisotropic*** noise. As shown by the green ring in Fig. 7a, DAMP aligns the noise's dimensional distribution with the data's statistical distribution, allocating minimal perturbation to low-variance dimensions. This reduces the covariance of noise by three orders of magnitude. Consequently, in Fig. 7b, DAMP shrinks the per-sample perturbation error

by 17.6× (green vs. orange wave). DAMP thus preserves data utility for the same provable privacy guarantee.

4.2.2 *Initial Distribution and Runtime Update*

DAMP initializes with user's historical distribution, and then updates this distribution continuously as the user generates new expressions. During runtime, DAMP tracks the latest distribution, computes the minimum noise needed to meet the target privacy guarantee, and adds this noise to the offloaded latent code. By recalibrating noise as the distribution drifts slowly over time, DAMP preserves provable privacy protection throughout long-term use.

Crucially, latent code distribution is computed and stored entirely on user's local device (e.g., VR headset). This distribution is **never shared** and only used to calibrate the noise injected into the latent codes before they are offloaded, ensuring the distribution statistics themselves remain private.

4.2.3 *Provable Formal Privacy Guarantee*

To model privacy leakage and provide formal privacy guarantee, DAMP adopts the *PAC Privacy* framework (Xiao & Devadas, 2023; Xiao et al., 2024; Xiao, 2024; Xiao et al., 2025), which characterizes privacy through adversary's inference hardness. The privacy risk is quantified by *posterior success rate*, i.e. the probability that an adversary can correctly infer private information after observing the leakage.

Definition [PAC Privacy (Xiao & Devadas, 2023), (Xiao & Devadas, 2025)] A processing function $\mathscr{F}$ satisfies $(\delta_\rho, \rho, \mathsf{D})$-*PAC Privacy* (terminology listed in Tab. 1) if the following experiment is *impossible*:

A user samples X from the distribution D and sends $\mathscr{F}(X)$ to an adversary who knows D and $\mathscr{F}$. The adversary returns an estimate $\hat{X}$ such that

$$\Pr\left[\rho(\hat{X}, X) = 1\right] \geq 1 - \delta_\rho,$$

where $\rho(\cdot,\cdot)$ is a correctness criterion (e.g., correctly guessing the expression in expression identification attack).

A smaller δ_ρ, i.e., an impossibility of a stronger adversarial reconstruction, intuitively implies a higher privacy risk.

In multi-user avatar reconstruction, each user's private texture X (e.g., facial expression) is drawn from a distribution D. When the encoded latent code $\mathscr{F}(X)$ is offloaded to the cloud, we must ensure that any adversary can NOT reconstruct the exact sensitive expression $\hat{X}$ from $\mathscr{F}(X)$, i.e. $\rho(\hat{X}, X) \neq 1$. For instance, under an *Expression Identification Attack*, $\rho(\hat{X}, X) = 1$ if the adversary correctly classifies the user's expression (e.g. *laughing* or *surprised*).

The posterior success rate $(1 - \delta_\rho)$ quantifies this risk. To bound it below a desired threshold for arbitrary adversary, DAMP introduces minimal perturbation noise $\boldsymbol{e}$ to the latent code $\mathscr{F}(X)$, as illustrated in ❻ of Fig. 5.

Table 1. **Formal Privacy Analysis Terminology**

Term (Symbol)	Definition
Private Data (X)	Unwrapped Texture in avatar reconstruction.
Data Distribution (D)	Distribution of Private Data (X).
Processing Function ($\mathscr{F}$)	Encoder (❸) in avatar reconstruction.
Observation (O)	The offloaded noisy latent code.
Estimation of Secret ($\hat{X}$)	The guess of adversary after observing O.
Prior Successful Rate ($1-\delta_{\rho,o}$)	The Successful Rate of correct guess **before** observing O.
Posterior SR (PSR, $1-\delta_\rho$)	The Successful Rate of correct guess **after** observing O.
noise $\boldsymbol{e}$	The multi-dimensional noise $\boldsymbol{e}$, same shape as $\mathscr{F}(X)$.
mutual information bound v	The mutual information between X and $\mathscr{F}(x)+\boldsymbol{e}$.

To calibrate $\boldsymbol{e}$, we first define the *optimal prior success rate*, the adversary's best guess **before** seeing the leakage:

$$\delta_{\rho,o} = \min_{X' \in \mathscr{X}^*} \Pr_{X \sim \mathsf{D}}\left[\rho(X', X) \neq 1\right]$$

where $\mathscr{X}^*$ denotes a uniform distribution of private samples. Once D and ρ are fixed, $(1-\delta_{\rho,o})$ is determined. According to (Xiao & Devadas, 2023), the posterior success rate $(1-\delta_\rho)$ is bounded by the mutual information between X and its noisy representation $\mathscr{F}(X)+\boldsymbol{e}$:

$$\delta_\rho \ln\left(\frac{\delta_\rho}{\delta_{\rho,o}}\right) + (1-\delta_\rho)\ln\left(\frac{1-\delta_\rho}{1-\delta_{\rho,o}}\right) \leq \mathsf{MI}(X; \mathscr{F}(X)+\boldsymbol{e}) \quad (1)$$

where $\mathsf{MI}(\cdot;\cdot)$ denotes mutual information, noted as v.

This inequality directly links the added noise $\boldsymbol{e}$ to an upper bound on adversarial inference capability, forming the theoretical foundation for DAMP noise calibration. We provide detailed step-by-step noise calculation in §B.

5 EXPERIMENTS

In this section, we evaluate whether PRIVATAR maintains utility (quality of avatar) and privacy while improving throughput under realistic VR hardware constraints.

5.1 Setup

Dataset. We use Multiface (hsin Wuu et al., 2023). The training set contains 13 identities, 65 or 143 labeled expressions per identity, 40 views per expression from various angles, in total ∼ 1.7 million frames and ∼ 15 TB data. The synthetic test set has 1,172 consecutive frames.

Hardware We evaluate on a Meta Quest Pro (902 GFLOPS GPU). The untrusted offload host is a PC with RTX 5090 (non-TEE GPU) and AMD Threadripper 7985WX (secure memory encryption, SME) (Advanced Micro Devices, Inc., 2020) as CPU TEE. Additional results for Quest 3 are in §F. Headset and PC are connected by WiFi-7 (≤20 Gbps).

PRIVATAR configuration. We adopt a VAE (Lombardi et al., 2018; hsin Wuu et al., 2023) and set B=4, yielding 16 frequency components. We rank components by L_2 norm variance and offload the 14 lowest-variance components. The base component is always local. The offloaded branch

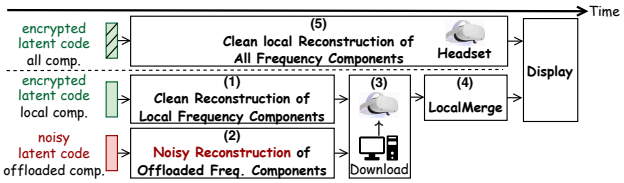
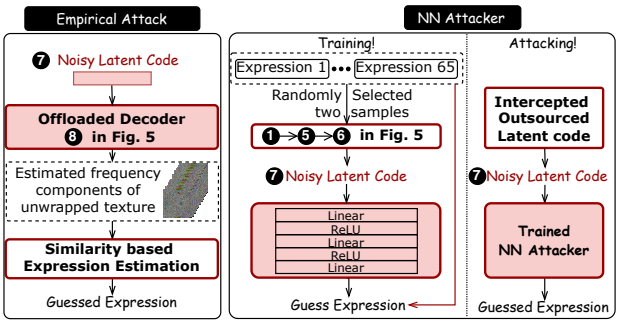


Figure 8. **Latency breakdown of avatar reconstruction in PRIVATAR.** (1–4) show collaborative headset–PC reconstruction. (5) represents full local reconstruction for highest throughput.



(a) Empirical Att. (b) NN-based Attacker

Figure 9. **Illustration of empirical attackers.** (9a) Empirical attacker takes in reconstructed offloaded components and estimate the expression based on its similarity to all frequency components of two random samples from all expressions. It achieves 86.5% PSR when offloading all components. (9b) NN attacker has three fully connected layers, and guesses expression from offloaded noisy latent code. Constructions are detailed in §D.

is trained for 2 epochs, batch size 10. Noise is calibrated by MI budget v following §B (default $v = 0.1$), and injected into the offloaded latent code. The choices of partitioning should be selected to maximize on-device compute utilization. For reconfigurable hardware (Tong et al., 2026b; 2024b) supporting diverse workload shapes with high compute utilization, finer-grained partitioning exposes a richer tradeoff space.

Baseline. The entire decoder runs locally on the headset (de-facto VR deployment). On Quest Pro at 60 FPS, the baseline supports 2.48 users with loss as $L_0 = 0.072$.

Metrics for Loss, Throughput, Privacy, Energy.
• ***Loss:*** Mean-squared error (MSE) between reconstructed texture and ground truth. We report normalized loss relative to $L_0 = 0.072$ of the unpartitioned VAE baseline.
• ***Latency:*** We breakdown latency as in Fig. 8: (1/4/5) Local compute (measured by roofline model (Williams et al., 2009)), (2) Offloaded compute (measured on RTX 5090 or AMD 7985WX), (3) Communication (download offloaded frequency components from PC to headset via WiFi-7).
• ***Throughput:*** We report max concurrent users at 60 FPS, i.e. throughput = #users/(critical-path-latency$\times$60).
• ***Privacy:*** the maximal Posterior Success Rate (PSR) of guessing an expression correctly. A PSR bound π means that for *any* adversary, the probability of correctly guessing the expression after observing the release is $\leq \pi$. We provide (a) t-PSR: a theoretical upper bound on posterior success rate following PAC Privacy (§B). (b) e-PSR: the empirical PSR, i.e., the maximum success rate across both (a) empirical attacker and (b) NN-based attacker in Fig. 9.
• ***Energy:*** compute and communication use 32 GOP/s/W (Guo et al., 2024) and 13.94 nJ/bit (Sun et al., 2014).

5.2 PRIVATAR vs SotAs

We compare normalized loss, #users and empirical PSR of PRIVATAR against SotAs in Tab. 2.

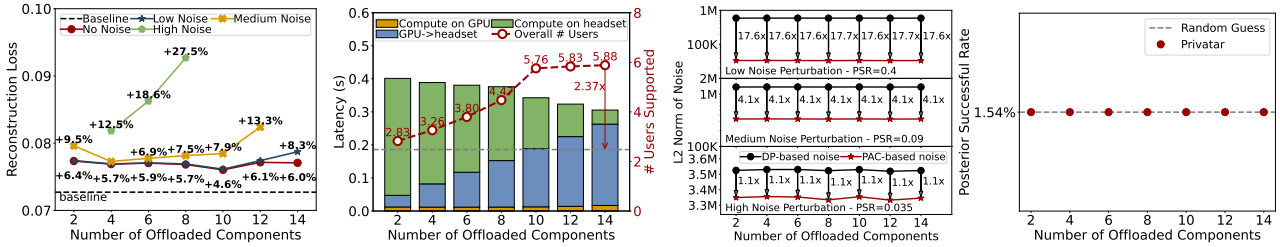
• **PRIVATAR vs Quantization:** Post-training *weight-only* 8-bit quantization of the decoder primarily reduces weight storage and memory loads of it. Total MAC count, i.e. local compute, is unchanged as activations (facial unwrapped texture) require floating-point data type to preserve fidelity. Hence PRIVATAR achieves 2.27$\times$ more throughput improvement over compute-bounded quantization at the same privacy protection level of random guess.
• **PRIVATAR vs Sparsity:** Channel pruning (He et al., 2017) (remove 10% channels and associated kernels with smallest L_2 norm). It does effectively reduce local computation but makes the shape of workload irregular, which requires dedicated hardware support that is not available on Qualcomm Snapdragon XR2+ Gen 1 on Quest Pro. Hence PRIVATAR provides 2.05$\times$ higher throughput over sparsity. Higher sparsity ratio introduces unbearable loss increase, and being impractical, and thus we did not compare against it.
• **PRIVATAR vs CPU TEE:** We offload the same component set as PRIVATAR to a CPU (AMD 9950X3D with SME). We use CPU because GPU TEEs are unavailable on RTX 5090. While secure memory encryption offers stronger privacy guarantee than PRIVATAR, the CPU with SME runs slower than GPU without TEE. Therefore PRIVATAR achieves 1.32$\times$ higher throughput at the same level of privacy guarantee (random guess).
• **Fully Offload (FO):** FO applies isotropic Gaussian DP noise to the latent code of entire facial texture, and offloads it to the GPU. This is distribution-agnostic and thus conservative for anisotropic latent code. Consequently, it introduces prohibitive 105$\times$ loss increase and becomes impractical.

In summary, with 14 offloaded components, PRIVATAR reaches 2.37$\times$ users and 2.17$\times$ users/watt, and achieves 2.06$\times$, 2.27$\times$, 1.32$\times$ higher throughput than sparsity (10% channel pruning), quantization (8-bit weights) and TEE-like (CPU with secure memory encryption) under the same privacy guarantee of e-PSR bound of random guess (1.54%).

Key Takeaway: At equal e-PSR (random-guess, 1.54%), PRIVATAR lies on the throughput–loss Pareto frontier: partial offloading reduces headset decode cost, while DAMP achieves the target e-PSR with minimal noise perturbation.

5.3 PRIVATAR Ablation Study

PRIVATAR has two design choices *(1) number (m) of fre-*



(a) Loss – m (b) Latency/Throughput – m (c) Noise L2 Norm – m (d) e-PSR – m

Figure 10. **Effects of number of offloaded components (m) on (a) Reconstruction Loss, (b) latency (left bar) and throughput (right red line) of avatar reconstruction, (c) the amount of noises needed for given privacy guarantee, and (d) Robustness against empirical attack. (10a)** Curves show normalized reconstruction loss relative to the unpartitioned VAE used in baseline (L_0=0.072) for PRIVATAR with no noise and with DAMP at low/medium/high noise. HP alone increases loss by only 5.7∼6.4% across m. Low and medium DAMP settings keep degradation modest (often visually indistinguishable), while high-noise settings trade accuracy for tighter PSR bounds. **(10b)** The latency breakdown of reconstructing a single user while the overall throughput (number of users, # user per second) are shown on left and right axis. **(10c) Noise efficiency**: For the same PSR bound, DAMP leverages per-user latent anisotropy to shrink the required L_2 norm of noise by up-to 17.6× (PSR= 0.4), 4.1× (0.09), and 1.1× (0.035) relative to isotropic DP-based noise. **(10d) Empirical PSR**: The e-PSR remains at random-guess (≈1.54%) for PRIVATAR across all m (with or without DAMP), since the base frequency never leaves the headset and the offloaded view is low information.

Table 2. **PRIVATAR vs. SotAs.** Note: Loss is MSE ratio to local baseline (↓: lower is better); #Users is a ratio to baseline (↑: higher is better). e-PSR: max PSR over two empirical attackers. ***Takeaway:*** PRIVATAR offers highest #users and invisible loss increase.

Technique	Loss (↓)	# Users (↑)	e-PSR
(***Baseline***) Local Reconstruction	1×	1×	1.54%
(***Sparsity***) 10% Channel Pruning	1.25×	1.15×	1.54%
(***Quantization***) 8-bit Weights	1.004×	1.04×	1.54%
(***TEE***) CPU w/ Secure Memory Encryption	1×	1.79×	1.54%
(***FO***) Fully Offloading + Isotropic Noise	105×	2.3×	85.6%
(**PRIVATAR**) Partial Offloading + DAMP Noise	**1.08**×	**2.37**×	**1.54**%

Table 3. **Ablation setup.** We sweep the number of offloaded components m and the MI budget v that calibrates DAMP. t-PSR is the certified PSR bound; e-PSR is the maximum observed empirical PSR bound across the empirical and NN attackers.

Privacy Level	# offloaded components (m)	e-PSR	t-PSR
Fully Offload + No Noise	16	85.6%	None
HP + No Noise	{2,4,6,8,10,12,14}	1.54%	None
HP + Low Noise ($v = 1$)	{2,4,6,8,10,12,14}	1.54%	40%
HP + Medium Noise ($v = 0.1$)	{2,4,6,8,10,12,14}	1.54%	9%
HP + High Noise ($v = 0.01$)	{2,4,6,8,10,12,14}	1.54%	3.5%

quency components to be offloaded, and *(2) the amount of noises to be added into the offloaded components*. We sweep $m \in \{2,4,6,8,10,12,14\}$ and three t-PSR targets {40%, 9%, 3.5%}, corresponding to mutual information bound as $v = \{1, 0.1, 0.01\}$, plus a no-noise reference and a fully-offloading all components reference in Tab. 3.

5.3.1 *Knob 1: Offloaded Frequency Components*

Loss: With *no noise*, partitioning in the frequency domain increases reconstruction loss by ≈ 6% because local/offload path only sees partial frequency components. Qualitatively, this increase is visually subtle in our examples (see Fig. 1). We further conduct perceptual study using LPIPS (Zhang et al., 2018), where various partitioning choice only lead to negligible 0.72% LPIPS increase. This proves the negligible utility effects of different partition choices. At *low noise* (e.g., $v = 1$, t-PSR = 40%), the loss is similar to the no-noise case since the *local merger* (❿ in Fig. 5) can suppress small perturbations of offloaded low-energy components using local high-energy components. As v decreases (tighter privacy, larger effective noise), loss grows and the loss increase is more pronounced at larger m because a higher fraction of components are reconstructed from the noisy latent code (yellow/green curves in Fig. 10a).

Latency / Throughput: Increasing m shifts more decoding work off the headset, reducing local compute latency and improving overall throughput (red curve in Fig. 10b). The latency breakdown for one user (bars in Fig. 10b) shows: (i) **local** latency decreases with m (green), (ii) **communication** latency increases with m due to downloading more noisy reconstructed frequency components from PC to the headset (blue), and (iii) **offloaded compute** on RTX 5090 is small relative to the (i) and (ii). Takeaway: higher m improves throughput until communication dominates.

Noise: Noise is injected *only* in the offloaded latent (❼ in Fig. 5). The PC decodes noisy frequency components which are sent back to the headset. Thus, changing m alters *how many* components are affected by noise, not the per-latent noise calibration. Total noise energy per frame scales with the count of offloaded components rather than changing the noise distribution itself.

5.3.2 *Knob 2: Amount of DAMP Noises*

For a fixed m, decreasing the MI budget v (stronger privacy) increases the required noise. Across all m, DAMP requires noise with up to 17.6× less L_2 norm than isotropic DP at t-PSR = 40%, 4.1× less at 9%, and 1.1× less at 3.5% (Fig. 10c). Empirically, as long as the *base component stays local*, the attacker's e-PSR remains at random-guess

$\approx 1.54\%$ for all tested m and v (Fig. 10d). DAMP preserves this empirical behavior while certifying tighter t-PSR. The noise reduction gets smaller with increase of the noise, because distribution matters less with larger noise.

In short, PRIVATAR picks configuration of v=0.1 (t-PSR= 9%) and m=14, and increases loss by 8.3% over the local baseline (in Fig. 3b) while ensuring e-PSR at random-guess.

6 RELATED WORK

6.1 Security and Privacy of Avatars

Multi-user avatar reconstruction in AR/VR exposes rich motion and interaction data that inherently reveal user identity and behavior. Prior work shows that even without visual access to the face, users can be re-identified from head and hand trajectories (Nair et al., 2023), private activities can be inferred from motion traces (Nguyen et al., 2024), and typed text can be reconstructed through subtle movements (Slocum et al., 2023; Luo et al., 2024; Ni et al., 2025). These findings highlight the privacy risks in motion-driven avatar reconstruction.

6.2 Secure Offloading Techniques

Secure offloading strategies fall into three categories: **encryption**, **partitioning**, and **perturbation**.

Encryption: Homomorphic Encryption (HE) (Chou et al., 2018; Reagen et al., 2021; Lloret-Talavera et al., 2022; Lou et al., 2020) offers strong privacy by encrypting both data and computation, but incurs orders-of-magnitude overhead in compute and memory (Tong et al., 2026a; 2024a). Multi-Party Computation (MPC) (Goldreich, 1998) reduces the computational cost but requires multiple trusted parties, making both impractical for real-time VR workloads.

Partitioning: Input partitioning separates sensitive and less-sensitive features for selective offloading. *Block partitioning* (Vishwamitra et al., 2017; Yuan et al., 2024) masks facial regions, while *frequency partitioning* (Wang et al., 2020; Ji et al., 2022) transmits high-frequency components. However, avatar reconstruction depends on complete facial recovery, lacking formal privacy guarantees.

Perturbation: Offloading noisy data and related computation to untrusted devices causes utility degradation. The minimal noise required for a given privacy level is preferred. Privacy protection of individually released data would require local Differential Privacy (LDP) (Xiong et al., 2020). LDP adds noise before offloading to protect user data without trusting the collector (Gadotti et al., 2022; Cheu et al., 2021). Yet, LDP noise grows with data dimensionality, significantly degrading utility for 256-dimensional latent code used in Variational Auto-Encoder. To address this, PRIVATAR optimizes multi-dimensional noise allocation and applies PAC Privacy (Xiao & Devadas, 2023) to achieve minimal noise with provable privacy bounds.

7 CONCLUSION

This work presents PRIVATAR, the first framework enabling secure partial offloading of avatar reconstruction to untrusted devices, reducing headset computation and scaling multi-user VR. PRIVATAR reframes privacy as a *representation* decision by introducing two key components: (1) *Horizontal Partitioning (HP)*, which splits the private data so untrusted devices never see a complete, identifying view, and (2) *Distribution-Aware Minimal Perturbation (DAMP)*, which minimizes injected noise while providing formal privacy guarantees. Combining, Privatar enables VR headset to support 2.37× users with 9% more energy, and being robust against both empirical attacker and NN-based attacker.

7.1 Acknowledgments

This work was supported in part by ACE, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. We thank reviewers for their feedbacks.

REFERENCES

Advanced Micro Devices, Inc. Sev-snp: Strengthening vm isolation with integrity protection and more. Technical Report White Paper, Advanced Micro Devices, Inc., 2020. URL https://www.amd.com/content/dam/amd/en/documents/epyc-business-docs/white-papers/SEV-SNP-strengthening-vm-isolation-with-integrity-protection-and-more.pdf.

Carr, T., Lu, A., and Xu, D. Linkage attack on skeleton-based motion visualization. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, pp. 3758–3762, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3615263.

Cheu, A., Smith, A. D., and Ullman, J. R. Manipulation attacks in local differential privacy. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 883–900. IEEE, May 2021. doi: 10.1109/SP40001.2021.00001.

Chou, E., Beal, J., Levy, D., Yeung, S., Haque, A., and Fei-Fei, L. Faster cryptonets: Leveraging sparsity for real-world encrypted inference, 2018.

Croft, W. L., Sack, J.-R., and Shi, W. Obfuscation of images via differential privacy: From facial images to general images. *Peer-to-Peer Networking and Applications*, 14: 1705–1733, 2021.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, pp. 265–284, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3540327312. doi: 10.1007/11681878_14.

Dworkin, M. J. Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac. 2007.

Fan, L. Practical image obfuscation with provable privacy. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 784–789, 2019. doi: 10.1109/ICME.2019.00140.

Gadotti, A., Houssiau, F., Annamalai, M. S. M. S., and de Montjoye, Y. Pool inference attacks on local differential privacy: Quantifying the privacy guarantees of apple's count mean sketch in practice. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 501–518, Boston, MA, August 2022. USENIX Association. ISBN 978-1-939133-31-1.

Goldreich, O. Secure multi-party computation. *Manuscript. Preliminary version*, 78(110):1–108, 1998.

Guo et al. Neural Network Accelerator Comparison, September 2024. URL https://nicsefc.ee.tsinghua.edu.cn/projects/neural-network-accelerator/.

He, Y., Zhang, X., and Sun, J. Channel pruning for accelerating very deep neural networks, 2017.

hsin Wuu, C., Zheng, N., Ardisson, S., Bali, R., Belko, D., Brockmeyer, E., Evans, L., Godisart, T., Ha, H., Huang, X., Hypes, A., Koska, T., Krenn, S., Lombardi, S., Luo, X., McPhail, K., Millerschoen, L., Perdoch, M., Pitts, M., Richard, A., Saragih, J., Saragih, J., Shiratori, T., Simon, T., Stewart, M., Trimble, A., Weng, X., Whitewolf, D., Wu, C., Yu, S.-I., and Sheikh, Y. Multiface: A dataset for neural face rendering, 2023.

Huang, J., Subhajyoti Mallick, S., Amat, A., Ruiz Olle, M., Mosella-Montoro, A., Kerbl, B., Vicente Carrasco, F., and De la Torre, F. Echoes of the coliseum: Towards 3d live streaming of sports events. *ACM Trans. Graph.*, 44 (4), July 2025. ISSN 0730-0301. doi: 10.1145/3731214.

Intel Corporation. Sgx programming reference. Programming Reference 329298-002, Intel Corporation, 2014. URL https://www.intel.com/content/dam/develop/external/us/en/documents/329298-002-629101.pdf. Accessed on April 19, 2026.

Ji, J., Wang, H., Huang, Y., Wu, J., Xu, X., Ding, S., Zhang, S., Cao, L., and Ji, R. Privacy-preserving face recognition with learnable privacy budgets in frequency domain, 2022.

Keller, M. and Sun, K. Secure quantized training for deep learning, 2022.

Kim, Y.-J., Sra, M., and Höllerer, T. Audience amplified: Virtual audiences in asynchronously performed ar theater. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 475–484. IEEE, October 2024. doi: 10.1109/ismar62088.2024.00062.

Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., Ibrahim, M., and van der Maaten, L. Crypten: Secure multi-party computation meets machine learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4961–4973. Curran Associates, Inc., 2021.

Kopalidis, T., Solachidis, V., Vretos, N., and Daras, P. Advances in facial expression recognition: a survey of methods, benchmarks, models, and datasets. *Information*, 15 (3):135, 2024.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lloret-Talavera, G., Jorda, M., Servat, H., Boemer, F., Chauhan, C., Tomishima, S., Shah, N. N., and Peña, A. J. Enabling homomorphically encrypted inference for large dnn models. *IEEE Transactions on Computers*, 71(5): 1145–1155, 2022. doi: 10.1109/TC.2021.3076123.

Lombardi, S., Saragih, J., Simon, T., and Sheikh, Y. Deep appearance models for face rendering. *ACM Transactions on Graphics*, 37(4):1–13, July 2018. ISSN 1557-7368. doi: 10.1145/3197517.3201401.

Lou, Q., Lu, W.-j., Hong, C., and Jiang, L. Falcon: fast spectral inference on encrypted data. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Luo, S., Nguyen, A., Farooq, H., Sun, K., and Yan, Z. Eavesdropping on controller acoustic emanation for keystroke inference attack in virtual reality. In *Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, 2024.

Ma, S., Simon, T., Saragih, J., Wang, D., Li, Y., Torre, F. D. L., and Sheikh, Y. Pixel codec avatars, 2021. URL https://arxiv.org/abs/2104.04638.

Nair, V., Guo, W., Mattern, J., Wang, R., O'Brien, J. F., Rosenberg, L., and Song, D. Unique identification of 50,000+ virtual reality users from head & hand motion data. In *32nd USENIX Security Symposium (USENIX Security 2023)*, Anaheim, CA, 2023.

Near, J. P., Darais, D., Lefkovitz, N., Howarth, G. S., et al. *Guidelines for evaluating differential privacy guarantees*. US Department of Commerce, National Institute of Standards and Technology, 2025.

Nguyen, A., Zhang, X., and Yan, Z. Penetration vision through virtual reality headsets: Identifying 360-degree videos from head movements. In *33rd USENIX Security Symposium (USENIX Security 2024)*, Philadelphia, PA, 2024.

Ni, T., Du, Y., Zhao, Q., and Wang, C. Non-intrusive and unconstrained keystroke inference in VR platforms via infrared side channel. In *Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, 2025.

Park, J., Choi, Y., and Lee, K. M. Research trends in virtual reality music concert technology: A systematic literature review. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

Reagen, B., Choi, W.-S., Ko, Y., Lee, V. T., Lee, H.-H. S., Wei, G.-Y., and Brooks, D. Cheetah: Optimizing and accelerating homomorphic encryption for private inference. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 26–39, 2021. doi: 10.1109/HPCA51647.2021.00013.

Shamir, A. How to share a secret. *Commun. ACM*, 22: 612–613, November 1979. doi: 10.1145/359168.359176.

Slocum, C., Zhang, Y., Abu-Ghazaleh, N., and Chen, J. Going through the motions: AR/VR keylogging from user head motions. In *32nd USENIX Security Symposium (USENIX Security 2023)*, Anaheim, CA, 2023.

Sridhar, M., Xiao, H., and Devadas, S. Pac-private algorithms. In *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2025.

Strang, G. The discrete cosine transform. *SIAM*, 1999.

Sun, L., Sheshadri, R. K., Zheng, W., and Koutsonikolas, D. Modeling wifi active power/energy consumption in smartphones. In *2014 IEEE 34th international conference on distributed computing systems*, pp. 41–51. IEEE, 2014.

Tong, J., Dang, J., Golder, A., Raychowdhury, A., Hao, C., and Krishna, T. Accurate low-degree polynomial approximation of non-polynomial operators for fast private inference in homomorphic encryption. In Gibbons, P., Pekhimenko, G., and Sa, C. D. (eds.), *Proceedings of Machine Learning and Systems*, volume 6, pp. 210–223, 2024a.

Tong, J., Itagi, A., Chatarasi, P., and Krishna, T. Feather: A reconfigurable accelerator with data reordering support for low-cost on-chip dataflow switching. In *Proceedings of the 51th Annual International Symposium on Computer Architecture*, ISCA '24, Argentina, 2024b. Association for Computing Machinery.

Tong, J., Huang, T., Dang, J., de Castro, L., Itagi, A., Golder, A., Ali, A., Jiang, J., Kun, J., Arvind, Suh, G. E., and Krishna, T. Leveraging asic ai chips for homomorphic encryption. HPCA'26, Australia, 2026a. 2026 IEEE International Symposium on High Performance Computer Architecture (HPCA).

Tong, J., Li, Y., Jain, D., Mendis, C., and Krishna, T. Minisa: Minimal instruction set architecture for next-gen reconfigurable inference accelerator. In *Proceedings of the 34th Annual International Symposium on Performance Analysis of Systems and Software*, ISPASS '26, 2026b.

Vishwamitra, N., Knijnenburg, B., Hu, H., Kelly Caine, Y. P., et al. Blur vs. block: Investigating the effectiveness of privacy-enhancing obfuscation for images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 39–47, 2017.

Wang, H., Wu, X., Huang, Z., and Xing, E. P. High frequency component helps explain the generalization of convolutional neural networks, 2020.

Williams, S., Waterman, A., and Patterson, D. Roofline: An insightful visual performance model for multicore architectures. *Commun. ACM*, apr 2009.

Xiao, H. *Automated and Provable Privatization for Black-Box Processing*. PhD thesis, Massachusetts Institute of Technology, 2024.

Xiao, H. and Devadas, S. Pac privacy: Automatic privacy measurement and control of data processing. In *Annual International Cryptology Conference*, pp. 611–644. Springer, 2023.

Xiao, H. and Devadas, S. Pac privacy and black-box privatization. *IEEE Security & Privacy*, 23(4):92–97, 2025.

Xiao, H., Wan, J., and Devadas, S. Geometry of sensitivity: Twice sampling and hybrid clipping in differential privacy with optimal gaussian noise and application to deep learning. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2636–2650, 2023a.

Xiao, H., Xiang, Z., Wang, D., and Devadas, S. A theory to instruct differentially-private learning via clipping bias reduction. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 2170–2189. IEEE, 2023b.

Xiao, H., Suh, G. E., and Devadas, S. Formal privacy proof of data encoding: The possibility and impossibility of learnable obfuscation. In *Proceedings of the 2024 ACM*

SIGSAC Conference on Computer and Communications Security, 2024.

Xiao, H., Wan, J., Shi, E., and Devadas, S. One-sided bounded noise: Theory, optimization algorithms and applications. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2025.

Xiong, X., Liu, S., Li, D., Cai, Z., and Niu, X. A comprehensive survey on local differential privacy. *Security and Communication Networks*, 2020(1):8829523, 2020.

Yang, Z., Sarwar, Z., Hwang, I., Bhaskar, R., Zhao, B. Y., and Zheng, H. Can virtual reality protect users from keystroke inference attacks? In *33rd USENIX Security Symposium (USENIX Security 2024)*, Philadelphia, PA.

Yuan, M.-J., Zou, Z., and Gao, W. Bi-cryptonets: Leveraging different-level privacy for encrypted inference, 2024.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Zhang, Z., Fort, J. M., and Giménez Mateu, L. Facial expression recognition in virtual reality environments: challenges and opportunities. *Frontiers in Psychology*, Volume 14 - 2023, 2023. ISSN 1664-1078. doi: 10.3389/fpsyg.2023.1280136.

Zhao, Y. and Chen, J. A survey on differential privacy for unstructured data content. *ACM Comput. Surv.*, 54(10s), sep 2022. ISSN 0360-0300. doi: 10.1145/3490237.

Zhou, H., Dong, Y., Inami, M., Sarsenbayeva, Z., and Withana, A. A survey on methodological approaches to collaborative embodiment in virtual reality, 2025.

A ARTIFACT APPENDIX

A.1 Abstract

This artifact contains the full implementation and evaluation pipeline for Privatar, a privacy-preserving real-time multi-user VR avatar reconstruction system. It horizontally partitions a frequency-decomposed VAE decoder, keeping privacy-sensitive low-frequency components on the local VR headset while offloading high-frequency components to an untrusted cloud with calibrated noise injection. The artifact includes: (1) training and testing scripts for the baseline VAE model and five design-choice variants (direct split, quantization, sparsity, frequency decomposition, and horizontally partitioned frequency decomposition), (2) latency profiling and FLOPs calculation, (3) differential privacy (DP) and PAC privacy noise generation, (4) noisy inference under various mutual information budgets, and (5) empirical and NN-based expression identification attacks. All experiments are reproducible within an NVIDIA GPU Docker container. Link to code: https://github.com/georgia-tech-synergy-lab/Privatar.

A.2 Artifact check-list (meta-information)

• **Algorithm:** Variational Autoencoder (VAE) with Block DCT frequency decomposition, horizontal partitioning, DP and PAC privacy noise injection

• **Data set:** Multiface dataset (subject 6795937), downloaded via provided script (∼1 TB)

• **Hardware:** NVIDIA GPU (validated on RTX 5090/4090/3090), 16 GB+ GPU memory, 52 GB+ system RAM.

• **Metrics:** MSE (screen, texture, vertex), LPIPS, latency (ms), FLOPs, posterior success rate (PSR)

• **Output:** Trained model weights (`best_model.pth`), test metrics, latent codes (`z_*.pth`, `z_offload_*.pth`), noise covariance matrices (`.npy`), attack accuracy (PSR)

• **Experiments:** Training (6 variants), testing (6 variants), latency profiling (5 variants + FLOPs), DP noise generation (80 files), PAC noise generation (75 files), noisy inference, empirical attack, NN-based attack, frequency covariance analysis

• **How much time is needed to prepare workflow?:** ∼30 minutes (Docker setup, dependency and data setup)

• **How much time is needed to complete experiments?:** Full training: ∼48 hours per variant (100K iterations); functional test (1K iterations): ∼16 minutes per variant. Testing, noise calculation, and attacks: ∼2 hours total.

• **DOI:** 10.5281/zenodo.19443137

A.3 Description

The artifact is delivered as a GitHub repository containing all source code, configuration files, and experiment scripts. The repository is structured into six model variant directories and a shared `experiment_scripts` directory:

- `multiface/` — Baseline VAE (DeepAppearanceVAE)
- `multiface_direct_split/` — Direct architecture split into local + cloud paths
- `multiface_quantization/` — quantized decoder (8-16 bit)
- `multiface_sparse/` — Channel-pruned decoder (20–80% sparsity)
- `multiface_frequency_decompose/` — BDCT frequency decomposition (no offloading)
- `multiface_partition_frequency_decompose/` — BDCT + horizontal partitioning (PRIVATAR)
- `experiment_scripts/` — DP/PAC noise analysis, empirical attack configs, BDCT visualization, figure drawing, rendering utilities, and dataset download scripts

Each variant directory contains

- core logic `train.py`/`test.py`.

- Training/Testing launchers with configurable parameters `launch_train_job_serial.py` and test script
- latency measurement on VR headset, CPU or GPU `latency_profiling_script*.py`.
- architecture definitions: `models.py`.

A.3.1 Hardware dependencies

An NVIDIA GPU with CUDA support is required. The artifact has been validated on:

- NVIDIA RTX 5090 (primary evaluation platform)
- NVIDIA RTX 3090/4090 (supported via existing profiling scripts). Other generations like GH200 prefers different launcher like `torchrun` instead of Python.

At least 16 GB GPU memory and 52 GB system RAM are recommended. The baseline latency profiling script runs on CPU (modeling a VR headset without a discrete GPU); when the device is not a VR headset, it defaults to CPU. All other variant profiling scripts run on GPU (modeling cloud execution).

A.3.2 Software dependencies

- NVIDIA Docker: use based on your system, e.g. `pytorch:24.01-py3`.
- OS packages: `mesa-utils`, `mesa-common-dev`, `libegl1-mesa-dev`, `libgles2-mesa-dev`.
- Python packages: `torch`, `Pillow`, `ninja`, `imageio`, `opencv-python`, `torchjpeg`, `lpips`
- `nvdiffrast`: cloned from GitHub and installed via `setup.py` inside Python virtual environment.

A.3.3 Data sets

The Multiface dataset is used, containing facial images, tracked meshes, and unwrapped UV textures across 65+ expressions and 40 camera views. The dataset is downloaded via the provided script at `experiment_scripts/dataset_config/download_dataset.py` (∼1 TB for identity 6795937). We also need pretrained model weights (`6795937_best_model.pth`, 97 MB).

A.4 Installation

Detailed step-by-step commands are provided in the repository's `README.md`. The installation involves five steps:

- Clone the repository and launch the Docker container (`nvcr.io/nvidia/pytorch:24.01-py3`) with GPU access, mounting the repository to `/work`.
- Install OS-level packages and Python dependencies.
- Install `nvdiffrast` from source. For RTX 5090, apply the provided `nvdiffrast_patch.sh` and install the nightly PyTorch build with CUDA 13.0 support.

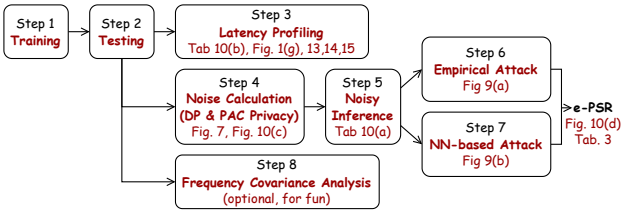


Figure 11. **Evaluation Flow**

- Download the Multiface dataset (∼1 TB).
- Download the pretrained model checkpoint (`6795937_best_model.pth`, 97 MB).
- All subsequent commands assume `/work` is the mount point inside Docker.

A.5 Experiment workflow

The experiments follow an eight-step pipeline. Each step depends on the outputs of previous steps. The overview of all steps and how to obtain results of the papers are illustrated in Fig. 11.

Functional test vs. full reproduction. For functional testing, set `val_num=30` and `max_iter=1000` (∼16 minutes per variant). For full reproduction, use `val_num=500` and `max_iter=100000` (∼48 hours per variant). Note: the baseline `multiface/launch_train_job_serial.py` ships with small defaults (`val_num=50`, `max_iter=100`); all other variants default to full reproduction values.

Step 1: Training.
Train all six model variants by running `launch_train_job_serial.py` in each variant directory. Each produces a `best_model.pth` checkpoint in `/work/training_results/`. Configurable parameters include: quantization bitwidths (`bitwidth_list`: 8–16), sparsity ratios (`sparsity_list`: 0.2–0.8), and number of offloaded frequency components (`num_freq_comp_offloaded_list`: 2–14).

Step 2: Testing.
Run `launch_test_job_serial.py` in each variant directory. This evaluates model quality (MSE, LPIPS) and saves latent codes (`z_<id>.pth` and `z_offload_<id>.pth`) to `testing_results/<project_name>/latent_code/`, which are required for noise calculation in Step 4.

Step 3: Latency Profiling.
Run `latency_profiling_script.py` in each variant directory. The baseline measures CPU latency (modeling VR headset); all others measure GPU latency (modeling cloud). For the partitioned variant, run both `latency_profiling_script_local_path.py` (local decoder) and `*_offload_path.py` (offloaded). Run `latency_flops_calculation.py` for FLOPs analysis across partition configurations.

Step 4: Noise Calculation.
Two noise mechanisms are supported: Differential Privacy (DP) based on L2 norm of latent codes, and PAC Privacy leveraging per-dimension covariance via SVD decomposition. Run DP noise scripts in `experiment_scripts/dp_analysis/` (`dp_noise_generation_for_multiface.py` for baseline; `dp_noise_generation_for_partition_multiface.py` for partitioned configs covering both local and offloaded branches) and PAC noise script in `experiment_scripts/pac_analysis/` (`pac_noise_generation_for_partition_multiface.py`). Both use mutual information bounds $[4, 3, 1, 0.1, 0.01]$ corresponding to posterior success rates $[98\%, 82.7\%, 40\%, 9\%, 3.5\%]$.

Step 5: Noisy Inference.
Run `launch_noisy_test_job_serial.py` in the partitioned frequency decomposition directory. Toggle `using_pac_noise` between `True` (PAC) and `False` (DP) to switch noise types. Results are saved to `/work/testing_results/`.

Step 6: Empirical Attack.
Run `launch_empirical_attack.py` in the partitioned directory. The attacker guesses expressions by matching predicted high-frequency texture components to precomputed reference components. Supports both PAC and DP noise via `using_pac_noise` toggle.

Step 7: NN-based Attack.
Train a 3-layer fully-connected classifier (256→128→66) via `launch_train_nn_attacker.py`, then evaluate under various noise levels via `launch_test_nn_attacker.py`. Training data consists of one sample per expression from `selected_expression_frame_list.txt`.

Step 8: Frequency Covariance Analysis.
Run `launch_l2norm_freq_cov_analysis.py` in the frequency decomposition directory to analyze the covariance trace of each of the 16 BDCT frequency components.

A.6 Evaluation and expected result

Step 1: Training (full reproduction should set `max_iter=100000`). All six variants train successfully and produce `best_model.pth` checkpoints.

Step 2: Testing (using trained models). Expected test metrics (minor run-to-run differences due to random sampling in validation). "no-noise" in Fig. 10a shows results.

- Baseline: screen≈0.076, LPIPS≈0.610
- Partition-14: screen≈0.077, LPIPS≈0.612

Step 3: Latency Profiling (RTX 5090). for Fig. 10b.

- Baseline decoder (CPU): 15.47 ms/inference
- Quantized decoder (GPU, 8-bit, traced): 0.685 ms
- Sparse decoder (GPU): 0.87 ms (10% pruned) to 0.42 ms (90% pruned)
- Partitioned local path: 0.24–0.32 ms across configs
- Partitioned offload path: 0.19–0.28 ms across 7 configs
- FLOPs: 1.48 G (14 offloaded) to 6.07 G (2 offloaded), a 75.6% reduction

Step 4: Noise Calculation. Noise trace values scale with mutual information budget as expected. Both DP and PAC noise generation complete successfully. One randomly selected noise results are used to generate Fig. 7 and Fig. 10c.

• DP noise: 80 files total — 5 (baseline complete offload) + 40 (partitioned offloaded branch) + 35 (partitioned local branch, 7 configs × 5 levels of mutual information). We use MI for mutual information in following contents.

• PAC noise: 75 files total — both local and offloaded branches across 8 partition configs × 5 MI levels (minus local files for the direct-split config)

Step 5: Noisy Inference. Noisy inference runs correctly with both PAC and DP noise. These contribute to "noisy loss" of Fig. 10a.

Step 6: Empirical Attack. The empirical attacker achieves PSR=3.18% or 1.54% for different identities when taking partition-14 with MI=1 PAC noise, close to the prior rate of $1/65 \approx 1.54\%$, confirming that the noise effectively prevents expression identification attack.

Step 7: NN-based Attack. The NN attacker trains successfully (10 epochs) and achieves PSR=1.54% on noisy latent codes, below the prior rate, confirming robustness against learned attacks. Results of both Step 6 and 7 are combined together to generate Tab. 3.

Step 8: Frequency Covariance Analysis. The covariance trace values for 16 frequency components match the expected output (e.g., component 0: ∼11308, component 15: ∼12.8). Low-frequency components carry ∼880× more variance than high-frequency components, confirming that high-frequency components are suitable for offloading.

A.7 Experiment customization

- **Training duration:** Adjust `val_num` and `max_iter` in each `launch_train_job_serial.py`. Functional test: `val_num=30`, `max_iter=1000`. Full reproduction: `val_num=500`, `max_iter=100000`.
- **Partition configurations:** To test different numbers of offloaded frequency components (2, 4, 6, 8, 10, 12, 14), modify `num_freq_comp_offloaded_list`.
- **Quantization bitwidth:** To use different precision, modify `bitwidth_list` (8–16 bits) in the quantization `launch_train_job_serial.py`.

- **Sparsity ratio:** Modify `sparsity_list` (0.2–0.8) in the sparsity `launch_train_job_serial.py`.
- **Noise type:** Toggle `using_pac_noise` between `True` (PAC) and `False` (DP) in noisy inference and attack launcher scripts.
- **Mutual information budget:** Adjust `mi_list` or `mutual_info_bound_list` to test different privacy levels (default: $[4,3,1,0.1,0.01]$).

B DAMP NOISE CALCULATION

B.1 Computing Minimal Distribution-Aware Noise

When we select noise $\boldsymbol{e}$ to be a multivariate Gaussian, i.e. $\boldsymbol{e} \sim \mathcal{N}(0,\Sigma_e)$, it suffices to optimize the noise covariance Σ_e for required mutual information bound v, i.e. $\mathsf{MI}(X;\mathscr{F}(X)+\boldsymbol{e}) \leq v$, as shown on the right hand side in (1), which consequently bounds our objective $\left(1-\delta_\rho\right)$. Following Algorithm 1 in (Sridhar et al., 2025), optimizations over covariance Σ_e are summarized below.

Step 1: Compute Covariance Matrix We first compute the covariance matrix $\Sigma_{\mathscr{F}(X)}$ of the profiled offloaded latent code $\mathscr{F}(X) \in \mathbb{R}^d$, a d-dimensional real vector ($d = 256$), where we assume the input X is uniformly selected from a data pool of expressions. And then we conduct the Singular Value Decomposition (SVD) of the covariance, denoted as

$$U \cdot s \cdot U^T = \Sigma_{\mathscr{F}(X)} \tag{2}$$

where $s = \text{Diag}\{\lambda_{0:d-1}\}$ represents the diagonal matrix of eigenvalues $\lambda_i, i \in [0,d)$ while the unitary matrix U corresponds to the eigenvectors.

Step 2: Obtain Minimal Covariance of Noise for a Given Posterior Successful Rate Given a posterior successful rate ρ requirement, Eq. 1 obtains the required mutual information v. Then, the i-th eigenvalue of minimal noise covariance, $\sigma = \text{Diag}\{\sigma_{0:d-1}\}$, is obtained through

$$\sigma_i = \frac{\sqrt{\lambda_i}\Sigma_{j=0}^{d-1}\sqrt{\lambda_j}}{2v} \tag{3}$$

Step 3: Inject Random Noise per Offloaded Latent Code PRIVATAR samples noise $\boldsymbol{e}$ from the Gaussian distribution $\mathcal{N}\left(0, U \cdot \sigma \cdot U^T\right)$ and release the noisy latent code $\mathscr{F}(X)+\boldsymbol{e}$. The noise intensity satisfies $\mathbb{E}[\|\boldsymbol{e}\|_2^2] = \sum_{i=0}^{d-1}\sigma_i$.

By tailoring noise to statistical distribution, DAMP significantly reduces the noise magnitude compared to conventional DP approaches, thereby mitigating reconstruction accuracy degradation.

C PRIVACY ANALYSIS OF PRIVATAR SETUP

C.1 Formal Privacy Guarantee (t-PSR)

For the Expression Identification Attack (EIA) in §4.2.3, we use $(\delta_\rho, \rho, \mathsf{D})$–PAC Privacy to certify a *posterior success rate* (PSR) bound, following §B.

For the identity (user) in the multiface dataset with 65 categories of expressions, the prior success rate will be $1-\delta_{\rho,o} = 1/65$ (uniform best guess), hence $\delta_{\rho,o} = 64/65$. PAC Privacy (Eq. 1) relates the target posterior successful rate δ_ρ to a mutual information (MI) budget v.

$$\delta_\rho \ln\frac{\delta_\rho}{\delta_{\rho,o}} + \left(1-\delta_\rho\right)\ln\frac{1-\delta_\rho}{1-\delta_{\rho,o}} \leq \mathsf{MI}(X;\mathscr{F}(X)+\boldsymbol{e}) \triangleq v.$$

This mapping is *monotone*: tighter privacy (lower bound for the t-PSR, δ_ρ) requires a smaller MI (v). For our setting ($\delta_{\rho,o} = 64/65$), the v for different t-PSR bound are:

t-PSR upper bound $\{0.40,\ 0.09,\ 0.035\} \Rightarrow v \approx \{1,\ 0.1,\ 0.01\}$

We then choose the *minimal* Gaussian noise covariance Σ_e that achieves $\mathsf{MI}(X;\mathscr{F}(X)+\boldsymbol{e}) \leq v$ following DAMP in §B.1. This per-user, distribution-aware calibration yields the minimum noise (Fig. 10c) necessary to certify the desired PSR against arbitrary attack under given t-PSR upper bound.

C.2 e-PSR for Expression Identification Attack

Under two empirical attack, PRIVATAR with loss noise level reduces its PSR from 86.2% down to random guess, 1.54%, emphasizing the efficacy of the provided horizontal partitioning and distribution based noise minimization.

D DETAILED ATTACKER SETUP

Empirical Attacker It randomly samples 1 angle from each expression in training dataset, and then partition it into frequency components to serve as frequency reference of a given expression. Then it will directly compare noisy reconstructed frequency components against all reference frequency components, making a guess of the expression whose reference frequency component has minimal difference to the reconstructed noisy frequency components. This attacker achieves 86.5% e-PSR when offloading all frequency components, emphasizing its efficacy.

NN-based attacker The attacker comprises three fully connected layers that map the intercepted noisy latent code to an estimate of expression directly. We pre-collect two samples per expressions to serve as training dataset of the NN-based attacker. And then mount attack by directly fed offloaded noisy latent code into it in attacking phase.

E TEE RESULTS ON QUEST PRO

We use the latency profiled on AMD 9950 X3D with Secure Memory Encryption (SME) to serve as the latency for CPU

TEE. In this setup, offloaded components without noises are directly processed in CPU with SME with stronger privacy guarantee. However, given the CPU SME being slower than the corresponding GPU without TEE setup, the maximal throughput improving for offloading $m = \{2,4,6,8,10,12\}$ frequency components is 1.79×, which is 30% less than PRIVATAR, as shown in Fig. 12. Among the collaborative local headset - CPU reconstruction path, the computation on both CPU and headset remain bottleneck. ***Takeaway:*** CPU with TEE provides stronger privacy guarantee than PRIVATAR but being slower.

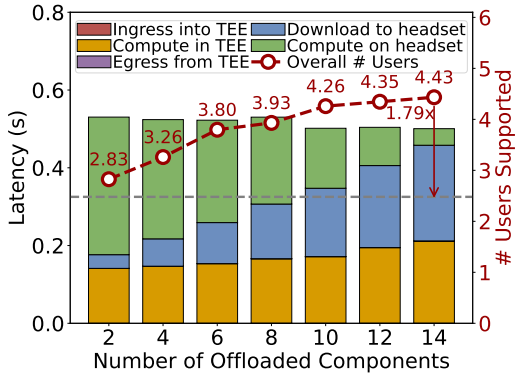


Figure 12. **Latency breakdown and overall throughput for CPU with secure memory encryption based secure partial offloading on Quest Pro.** Bars decompose end-to-end per-frame latency into *Ingress* (encrypt & copy into enclave), *Compute* (CPU inference inside the TEE), *Egress* (copy out from TEE), *Download* (return to headset), and *Compute on the headset* (compute latency of reconstructing local frequency components). The red curve shows the overall throughput (the number of concurrently supported users at 60 FPS). Despite strong confidentiality, limited CPU inference latency restricts it throughput improvement to ∼1.79×, which is less than 2.37× achieved by PRIVATAR.

F ADDITIONAL RESULTS FOR QUEST 3

Quest 3 provides 4.1× (3709 GOPS) higher overall local compute capability than Quest Pro, enabling it to reconstruct more users (10.6 # users in baseline for local reconstruction). We also provides **PRIVATAR vs SotAs** (Fig. 13), **PRIVATAR Latency Breakdown** (Fig. 13), and **TEE Latency Breakdown** (Fig. 15) of Quest 3. ***Takeaway:*** PRIVATAR provides better Pareto-frontier in throughput-loss curve on device with higher local computation capability, and communication is the system bottleneck for supporting more users from the collaborative local headset - GPU reconstruction.

Different partition choices induce different workload shapes, which map to the target hardware with different efficiencies. We therefore choose these choices of offloaded frequency components to maintain high on-device compute utilization for speedup. On reconfigurable hardware (Tong et al., 2026b; 2024b), finer-grained partitioning exposes a larger design space between local compute and communication.

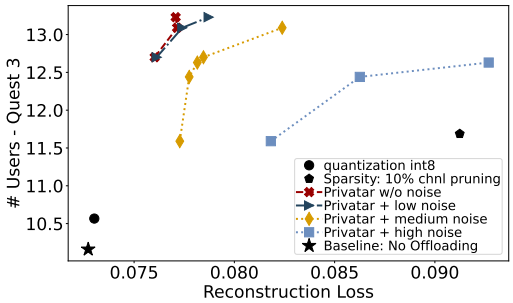


Figure 13. **Throughput-Loss comparison and ablation study with different number of offloaded frequency components for Quest 3.** PRIVATAR enables Quest 3 to achieve 1.34× more users with 4.6% loss, offering better pareto-frontier in throughput-loss curve when being compared against SotAs.

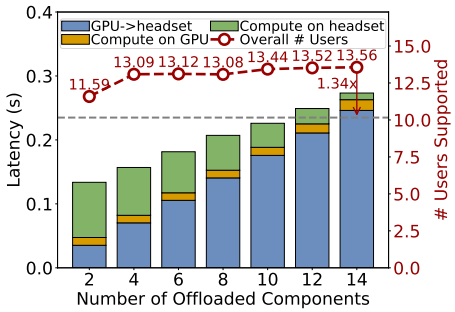


Figure 14. **Throughput Ablation Study on Quest 3.** PRIVATAR achieves 1.34× throughput improvement for Quest 3 because it offers 3.7 TFLOPs peak throughput which is 4.1× larger than Quest Pro. Communication bandwidth between PC and headsets is the bottleneck of throughput improvement from partial offloading.

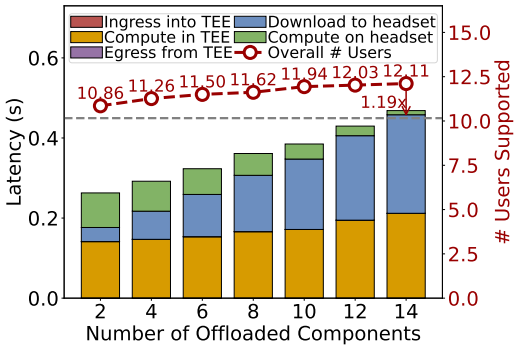


Figure 15. **Latency breakdown and overall throughput for CPU with secure memory encryption based secure partial offloading on Quest 3.** Quest 3 provides higher local compute hence a lower green bar, and stronger baseline with less throughput improvement of 1.19× under TEE. This aligns with the conclusion for Quest Pro: TEE provides stronger privacy guarantee but runs slower.