PPIDM: Privacy-Preserving Inference for Diffusion Model in the Cloud

Zhangdong Wang, Zhihuang Liu, Yuanjing Luo, Tongqing Zhou, Jiaohua Qin, and Zhiping Cai, *Member, IEEE*

***Abstract*—Cloud environments enhance diffusion model efficiency but introduce privacy risks, including intellectual property theft and data breaches. As AI-generated images gain recognition as copyright-protected works, ensuring their security and intellectual property protection in cloud environments has become a pressing challenge. This paper addresses privacy protection in diffusion model inference under cloud environments, identifying two key characteristics—denoising-encryption antagonism and stepwise generative nature—that create challenges such as incompatibility with traditional encryption, incomplete input parameter representation, and inseparability of the generative process. We propose PPIDM (Privacy-Preserving Inference for Diffusion Models), a framework that balances efficiency and privacy by retaining lightweight text encoding and image decoding on the client while offloading computationally intensive U-Net layers to multiple non-colluding cloud servers. Client-side aggregation reduces computational overhead and enhances security. Experiments show PPIDM offloads 67% of Stable Diffusion computations to the cloud, reduces image leakage by 75%, and maintains high output quality (PSNR = 36.9, FID = 4.56), comparable to standard outputs. PPIDM offers a secure and efficient solution for cloud-based diffusion model inference.**

***Index Terms*—Privacy-preserving, diffusion model, cloud environments, generate artistic images.**

I. Introduction

TEXT-TO-IMAGE diffusion models, such as Stable Diffusion [1], have demonstrated significant commercial value by generating high-quality and creative images from textual inputs. These models are widely used in applications such as artistic creation and advertising [2], with over 150 million downloads reported [3]. Despite their success, diffusion models are computationally intensive, often requiring high-end GPUs to produce high-resolution images [4]. Meanwhile, there is increasing demand for generating such images on resource-constrained devices, such as smartphones, to enhance productivity. For instance, users often seek to create presentation-ready images on the go, eliminating reliance on stationary high-performance computing devices [5].

To address these computational limitations, many users turn to cloud servers (e.g., Google Cloud) for model inference tasks [6]. Cloud-based diffusion models enable devices with limited resources to efficiently generate high-quality images [7]. However, this reliance on the cloud introduces new concerns, as the generated image data—often considered valuable user assets—may include personal creativity, sensitive information (e.g., user interests [8], healthcare data [9]), or corporate trade secrets (e.g., design drafts [10], animated character concepts [11]). These concerns are further amplified by the growing recognition of the copyright value of AI-generated images. In a landmark ruling, the Beijing Internet Court determined that AI-generated images can be recognized as works and are protected under copyright law [12].

Despite this legal recognition, significant privacy and security risks persist. Cloud servers are vulnerable to external attacks and misconfigurations, which can lead to data breaches. For example, the 2024 cyberattack on Snowflake exposed sensitive data from over 160 companies, affecting hundreds of millions of personal records [13]. Such incidents highlight the urgent need for privacy protection mechanisms tailored to diffusion model-generated images in cloud environments to safeguard user interests and foster trust in cloud services.

Existing research on privacy protection for diffusion models has primarily focused on the training phase, employing techniques such as differential privacy [14] and federated learning [15]. While effective for protecting training data, these approaches do not address the unique privacy risks posed during the inference phase in cloud environments. Additionally, traditional cloud-based encryption techniques, such as homomorphic encryption used in non-diffusion tasks like clustering [16] and retrieval [17], [18], are incompatible with diffusion models due to their distinct generative characteristics. Specifically, when applied to diffusion models, these conventional encryption methods face two critical challenges:

- **Denoising-Encryption Antagonism**: The denoising process of diffusion models inherently conflicts with the

Received 3 December 2024; revised 9 February 2025; accepted 18 March 2025. Date of publication 21 March 2025; date of current version 8 September 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62172155, Grant 62472434, and Grant 62102425; in part by the National Key Research and Development Program of China under Grant 2022YFF1203001; and in part by the Science and Technology Innovation Program of Hunan Province under Grant 2022RC3061 and Grant 2023RC3027. This article was recommended by Associate Editor B. Xiao. *(Corresponding authors: Zhiping Cai; Tongqing Zhou.)*

Zhangdong Wang, Zhihuang Liu, Tongqing Zhou, and Zhiping Cai are with the College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: wangzd@nudt.edu.cn; lzhliu@nudt.edu.cn; zhoutongqing@nudt.edu.cn; zpcai@nudt.edu.cn).

Yuanjing Luo is with the College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China, and also with the College of Computer Science and Information Technology, Central South University of Forestry and Technology, Changsha 410004, China (e-mail: yjluo@nudt.edu.cn).

Jiaohua Qin is with the College of Computer Science and Information Technology, Central South University of Forestry and Technology, Changsha 410004, China (e-mail: qinjiaohua@163.com).

Digital Object Identifier 10.1109/TCSVT.2025.3553514

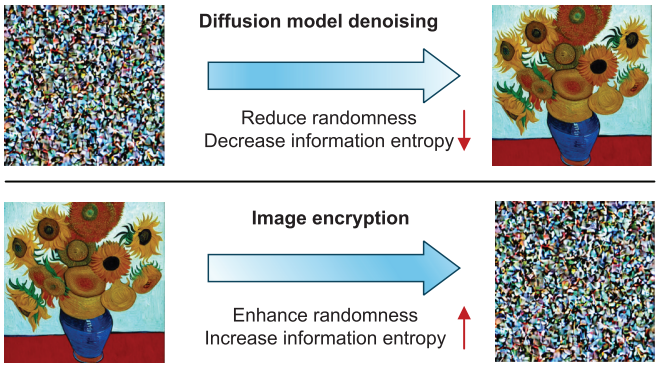


Fig. 1. Confronting image encryption noise addition and denoising diffusion model noise reduction.

noise-adding process of image encryption. As illustrated in Fig. 1, image encryption techniques transform a clear image into a noise-like representation without information leakage by increasing its entropy through added noise. Conversely, diffusion models generate clear content images from Gaussian noise by reducing entropy through denoising. This antagonistic relationship can result in decryption distortion during denoising, while the denoised encrypted images may inadvertently reveal plaintext information. Consequently, **traditional encryption techniques are unsuitable for diffusion model inference**.

- **Stepwise Generative Nature**: Diffusion models operate through a Markov chain-based generative process, where each iteration strongly depends on the previous step's output as the input for the next. If complete initial input parameters and features are provided, cloud servers could iteratively reconstruct the final private image. Thus, **it is imperative to avoid sharing full initial features and parameters with the cloud.** Furthermore, due to the structural constraints of the Markov chain, **altering the generation order would significantly degrade the quality of the generated images**.

To address the aforementioned challenges, we propose **PPIDM (Privacy-Preserving Inference for Diffusion Models)**, a novel framework for diffusion model inference in cloud environments that mitigates both resource constraints and privacy risks. Specifically, our approach leverages multiple non-colluding cloud servers to collaboratively process image generation. The client device is only responsible for lightweight tasks, such as data transmission and text encoding, while the computationally intensive denoising operations are offloaded to the cloud, thereby alleviating the client's computational burden. Each cloud server independently handles the denoising of local features, reducing the risk of feature leakage.

The main contributions of this work are summarized as follows:

- This work pioneers the definition of privacy protection in diffusion model inference within cloud environments, identifying key challenges: incompatibility with conventional encryption, incomplete representation of input parameters, and inseparability of the model's sequential structure. These challenges make conventional privacy methods unsuitable.
- We propose PPIDM, a privacy-preserving inference framework that avoids encryption and sequence modification. By distributing computationally intensive layers, including initial features, across multiple non-colluding cloud servers, PPIDM ensures each server only processes partial data, preventing full image reconstruction while meeting mobile users' resource and privacy needs.
- Experiments on regular and artistic datasets show that with four cloud devices, PPIDM offloads 67% of SD model computation to the cloud, maintains high output fidelity (PSNR = 36.9, FID = 4.56), and reduces image information leakage by 75%.

Roadmap The organization of this paper is as follows. Section II discusses the development of diffusion models and the field of image privacy protection in cloud environments, and basic knowledge of related technologies. Section III provides a detailed definition of the problem. Section IV describes the proposed PPIDM in detail. Section V conducts an experimental evaluation of PPIDM. Section VI analyzes the theoretical safety of PPIDM. Section VII discusses the future development of diffusion model inference for image generation. Finally, Section VIII summarizes the work presented in the paper.

II. Background

A. Related Work

We review related works on privacy protection for diffusion model inference in cloud environments, focusing on two areas: privacy protection for diffusion models and privacy protection for images in cloud environments.

1) Privacy Protection for Diffusion Models: Existing research on privacy protection for diffusion models focuses primarily on protecting training data sets, which can be categorized into two aspects: protecting training sets' source data and protecting sensitive concepts within training data.

a) Protecting the Source Data of Training Sets: In specific domains such as medical imaging, data privacy and decentralized data distribution make centralized training challenging. Medical institutions often cannot share data, and uploading data to the cloud for training poses significant privacy risks [22]. To address these issues, researchers have explored techniques such as federated learning [23], which enables distributed training of diffusion models while preserving data privacy. Federated learning avoids collecting raw data by aggregating model parameters, ensuring the security of private training data.

Tun et al. [24] was the first to investigate the application of federated learning to diffusion model training, demonstrating its potential for preserving training data privacy. FedDISC [15] integrates pre-trained diffusion models into a semi-federated learning framework, improving performance, generation quality, and stability under single communication rounds without client-side training. FedDiff [20] proposed a dual-branch multimodal learning network and federated framework based on

diffusion models to tackle the privacy challenges of heterogeneous multi-source data. Goede et al. [25] utilized the FedAvg algorithm in federated learning to train denoising diffusion models, achieving generation quality comparable to centralized training. Vora et al. [26] introduced the FedDM framework, which enhances communication efficiency through quantization and addresses data heterogeneity issues, enabling efficient privacy-preserving image generation in distributed environments.

While these studies focus on the privacy of training data in diffusion models, their methods are not suitable for addressing privacy issues in cloud-based inference and image generation, given the distinct requirements and processes of the inference phase.

b) Protecting Sensitive Concepts in Training Data: Diffusion models may memorize specific training samples (e.g., faces or trademarks) [27]. To mitigate the leakage of sensitive concepts, researchers have developed methods based on differential privacy (DP), focusing on optimizers, datasets, and training approaches to protect sensitive information.

DPDM [28] was the first to use DP-SGD (Differentially Private Stochastic Gradient Descent) for training diffusion models, identifying the pressing need for advancing privacy-preserving generative modeling with diffusion models. Ghalebikesabi et al. [29] proposed pretraining on public datasets followed by fine-tuning with DP-SGD on private datasets, achieving better results than previous methods. DP-LDMs [30] explored fine-tuning attention modules in latent diffusion models (LDMs) using DP-SGD, effectively balancing efficiency and privacy. PRIVIMAGE [14] selected more compact subsets of public data for pretraining through semantic queries, saving computational resources while maintaining image quality. DP-Promise [21] introduced a two-phase diffusion model training approach to reduce overall noise injection, achieving a superior balance between privacy and utility.

These studies primarily address the issue of sensitive concept leakage in training data but do not solve the privacy challenges of cloud-based diffusion model inference and image generation.

2) Privacy Protection for Images in Cloud Environments: In cloud environments, research on image privacy protection primarily focuses on discriminative tasks such as clustering [31] and retrieval [32]. Most of these approaches rely on homomorphic encryption or traditional encryption techniques to ensure image privacy.

a) Privacy Protection for Image Clustering Tasks: For image clustering, Bunn and Ostrovsky [33] proposed a privacy-preserving two-party k-Means clustering protocol based on Paillier homomorphic encryption, though its security significantly decreases with more than two participants. Liu et al. [34] achieved privacy-preserving clustering using fully homomorphic encryption (FHE), introducing a method to compare encrypted data distances with trapdoor information. Wu et al. [35] designed an outsourced k-Means clustering scheme combining fully homomorphic encryption and ciphertext packing techniques, enabling parallel computation without additional cost. Zhang et al. [16] further developed a multi-key FHE scheme to handle ciphertext conversion under different keys, enhancing data privacy protection.

b) Privacy Protection for Image Retrieval Tasks: For image retrieval, Bellafqira et al. [36] proposed a content-based image retrieval method using homomorphic encryption to protect image privacy, extracting wavelet-based features for similarity measurement. Li et al. [19] employed convolutional neural networks (CNNs) to extract plaintext image descriptors, followed by secure encryption and clustering-based indexing trees for retrieval. Wang et al. [18] designed a block-based image encryption scheme paired with a Transformer-based feature extractor to securely retrieve features from encrypted images. Yu et al. [17] proposed an encryption method compatible with JPEG compression, extracting local Markov features from encrypted images and constructing feature vectors via a bag-of-words model for image retrieval.

These studies show that privacy-preserving methods for clustering and retrieval tasks primarily depend on encryption techniques. Furthermore, these discriminative tasks do not modify the original encrypted image during the inference process. However, diffusion models involve complex denoising and reconstruction processes, making encryption-based approaches unsuitable for privacy protection in generative tasks.

Thus, new privacy-preserving methods must be developed to address the privacy risks inherent in diffusion model inference. Tab. I provides a comprehensive comparison of existing methods. Unlike prior approaches, our work is the first to integrate computational efficiency, generation quality, and privacy requirements in cloud-based diffusion model inference, meeting the demand for secure, high-quality image generation for mobile users.

B. Preliminaries

This section introduces relevant basic knowledge, including the relevant theoretical knowledge of text-to-image diffusion models and U-Net.

1) Text-to-Image Diffusion Models: The function of text-to-image diffusion models is to generate images matching the description from the input text prompt. The model consists of three key components: the CLIP (Contrastive Language-Image Pre-training) text encoder, U-Net, and the VAE decoder (Variational Auto-Decoder). Specifically, CLIP encodes the text prompt into a feature vector aligned with the semantics of the image; U-Net predicts the residual noise and converts Gaussian noise into latent space features of the image; the VAE decoder maps these latent features to the pixel space of the image.

The inference generation process is as follows: Given an input text prompt p, the CLIP text encoder E_{CLIP} first extracts the text's embedded features F_p. Then, Gaussian noise matrices $X_t \sim \mathcal{N}(0, I)$ are randomly initialized. Subsequently, at each time step, the noise matrix X_t, text embedded features F_p, and the time step t (where $0 < t \leq T$) are simultaneously fed into the U-Net model U. The U-Net uses these three inputs to calculate the noise matrix X_{t-1} for the next time step:

$$X_{t-1} = U(X_t, F_p, t) \tag{1}$$

TABLE I
THE COMPARISON BETWEEN PPIDM AND EXISTING RESEARCH

Scheme	Scenario	Task	Protection Target	Key Technology	F1	F2
Zhang et al. [16]	Single cloud	Image clustering	Inference content	Homomorphic encryption	✓	✓
Li et al. [19]	Single cloud	Image retrieval	Inference content	Conventional encryption	✓	×
Yu et al. [17]	Single cloud	Image retrieval	Inference content	Searchable encryption	✓	✓
Li et al. [20]	Multi-cloud	Diffusion model	Training source data	Federated learning	×	×
Yang et al. [15]	Multi-cloud	Diffusion model	Training source data	Semi-supervised federated learning	×	×
Wang et al. [21]	User end	Diffusion model	Training specific concepts	Differential privacy	×	×
PPIDM (Our)	Multi-cloud	Diffusion model	Inference content	Feature decomposition	✓	✓

Note: F1: Inference Content Protection, F2: Support Mobile User End

TABLE II
PARAMETER DISTRIBUTION OF SD AND SDXL MODELS

Model	SD1.4/1.5	SDXL
Text Encoder	123M	817M
VAE Decoder	50M	50M
U-Net Total Parameters	860M	2.6B
U-Net Convolution Parameters	589M	334M
U-Net Attention Mechanism Parameters	94M	956M

Through t iterative denoising, the latent features X_0 of the final denoised image are obtained. Finally, the VAE decoder D_{VAE} maps the denoised latent features X_0 to the final pixel-level image I.

2) U-Net: The U-Net network is a core component of the text-to-image diffusion model, which iteratively denoises Gaussian noise by integrating text features and time-step guidance, ultimately generating a latent representation of image features. In Stable Diffusion, the U-Net architecture enhances the traditional U-Net by incorporating ResNetBlock (with temporal embedding), Spatial Transformer (including attention layers, cross-attention, and self-attention modules), and CrossAttnDownBlock, CrossAttnUpBlock, and CrossAttnMidBlock modules. All modules consist of convolution (Conv), attention, normalization (GroupNorm), linear layers, SiLU activation, and Dropout, forming the basic architecture.

Table II compares the parameter distribution of SD and SDXL models, providing insight into their computational bottlenecks and opportunities for optimization in text-to-image diffusion models.

As shown in Table II, the U-Net accounts for 75–84% of the total model parameters in SD and SDXL. Among these, convolution (Conv) and attention layers make up 50–80% of the U-Net parameters. The main limiting factor for user devices is storage, as they must accommodate the parameters and memory requirements within a given hardware budget. At the same time, for diffusion model inference, addressing privacy concerns primarily involves protecting both input text and output images.

The convolution layers act as the backbone of U-Net, directly linking input and output image latent features, while attention layers directly associate with the embedded text features. Therefore, optimizing the convolution and attention mechanisms is key to resolving user resource constraints while addressing image and text privacy challenges.

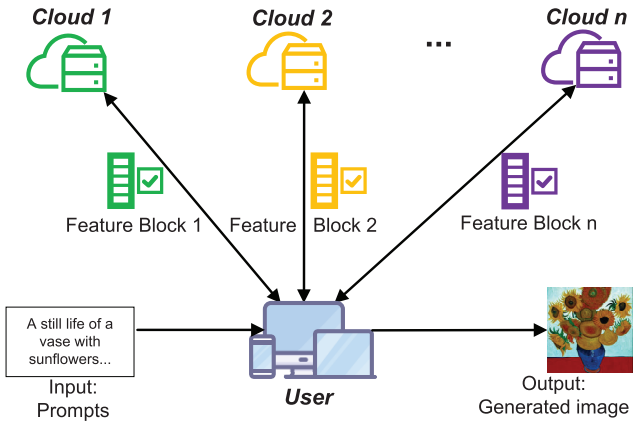


Fig. 2. System model.

III. PROBLEM FORMULATION

The problem of privacy-preserving inference for diffusion models in the cloud aims to address privacy and security challenges faced by resource-constrained users relying on cloud services for diffusion model inference. Specifically, users input text prompts and leverage the computational power of the cloud to perform inference tasks using text-to-image diffusion models, generating images corresponding to the input text. During this process, users aim to minimize the leakage of input text and output image content to protect personal creativity and privacy. Below, we introduce the system model, threat model, and design objectives of this work.

A. System Model

The system model of this work comprises two primary entities: users and cloud service providers. Users collaborate with multiple non-colluding clouds to complete the inference tasks of text-to-image diffusion models. The system model is illustrated in Fig. 2.

The definitions of the entities are as follows:

User: The owner of the personalized trained text-to-image diffusion model and the image generated. User devices are typically mobile devices with limited computational resources but high communication bandwidth (e.g., smartphones, and tablets). Users input text prompts and expect to receive corresponding images. They primarily handle lightweight computational tasks, including text feature extraction,

distribution and collection of denoising process features, and image decoding.

Cloud Service (CS): Cloud service providers are the primary computational entities for text-to-image diffusion model inference, equipped with ample computational resources and high communication bandwidth. This work employs multiple non-colluding cloud providers $\mathrm{CS} = \{\mathrm{CS}_1, \mathrm{CS}_2, \ldots, \mathrm{CS}_n\}$ to build the cloud service system, avoiding complete data control by a single cloud provider and reducing the risk of image privacy leakage. The cloud receives features and parameters from the user, performs the high-load computations of the denoising process, and returns the results to the user.

B. Threat Model

Based on the above system model, we define the potential threats faced by the system. The user, as the intellectual property owner of the images generated by the diffusion model inference, is considered fully trusted and secure.

The cloud service provider participates in the computations for image generation and, similar to existing image privacy protection schemes in cloud environments [17], [19], [23], is considered an "honest-but-curious" semi-trusted party. While adhering to the protocol, the CS may attempt to infer the content of the generated images illegally. Cloud providers, being organizations with significant reputational and legal considerations, have little incentive to tamper with communication data. Furthermore, to protect their commercial interests, cloud providers have a strong motivation to prevent external attackers from stealing cloud data.

While cloud providers are not allowed to view or use the content of generated images, they may be attacked, leading to data leakage. Considering the diverse security configurations of different cloud providers, we assume attackers can compromise a single cloud provider but cannot simultaneously compromise multiple providers. Additionally, due to conflicting interests and independent operations, cloud providers are assumed not to collude.

Malicious attackers, such as hackers, may extract feature data and computational layer parameters from a single cloud server to perform inference analysis on the data. The privacy protection in this work aims to safeguard the content of the generated images, which is closely related to the user input and highly personalized. Since attackers cannot reconstruct unique inference content by analyzing model parameters, we do not consider the security of model parameters.

C. Design Objectives

This work focuses on the following three design objectives:

Lightweight User-Side Computation: Since user devices in cloud environments typically have limited computational resources, users expect to use devices such as smartphones, tablets, or lightweight laptops for image generation. Thus, the primary design objective is to offload complex computational tasks to the cloud, minimizing the computational burden on the user side.

High-Quality Generated Images: While reducing user-side computational costs through cloud services, users also expect the quality of cloud-generated images to be comparable to locally generated ones. Hence, the second design objective is to ensure that the quality of generated images does not significantly degrade when offloading computations to the cloud.

Privacy-Preserving Generated Images: Users do not want the content of generated images, which belong to personal intellectual property and privacy, to be leaked. Through observations, we find that traditional encryption methods are unsuitable for diffusion models. Therefore, this work aims to minimize the proportion of leaked information in generated images to protect user privacy from an information leakage perspective.

IV. Privacy-Preserving Inference for Diffusion Model Scheme

A. Scheme Framework

To achieve the three design objectives of lightweight user-side computation, high-quality generated images, and privacy-preserving generated images, we propose a privacy-preserving inference for the diffusion model scheme (PPIDM). Its design consists of three core components:

1) **Text Privacy Protection**: Lightweight CLIP text encoding is performed on the user side to ensure text privacy, safeguarding the user's text data from being exposed to the cloud.
2) **Cloud Computation Offloading**: To reduce the computational burden on the user side, the high-load computational layers of the U-Net model are offloaded to the cloud for execution. This significantly reduces the resource consumption on the user side, especially for generating high-resolution images.
3) **Privacy-Preserving Feature Partitioning**: During the denoising process of U-Net, features closely related to the text and image are divided and distributed to multiple non-colluding cloud servers for processing. Each cloud server processes only part of the feature data, reducing the likelihood of any single cloud server acquiring a complete view of the data and thereby lowering the risk of data privacy leakage.

Tab. III presents the main notations and descriptions in this paper.

We define the core functional modules of PPIDM as follows:

- $E_{\text{CLIP}}(P) \rightarrow F_p$: Text encoder, input as text p, output as text embedding features F_p.
- $\mathbb{U} = \{\mathbb{U}_{\text{center}}, \mathbb{U}_{\text{local}}\}(F_p, X_t) \rightarrow X_{t-1}$: The proposed PPIU-Net, consisting of $\mathbb{U}_{\text{center}}$ on the user side and $\mathbb{U}_{\text{local}}$ on the cloud. Input as text embedding features F_p, timestamp t, and the current image features X_t. Output as denoised image features X_{t-1}.
- $D_{\text{VAE}}(X_t) \rightarrow I$: Image decoder, input as latent space image features X_t, output as generated image I.

The inference process of PPIDM involves interactions between the user and cloud servers, as illustrated in Fig. 3. The workflow is described as follows:

Step 1: The user inputs p on a mobile or similar lightweight device. The E_{CLIP} extracts the F_p. Simultaneously, the user

TABLE III
NOTATIONS AND DESCRIPTION

Notation	Description
p	Input text
E_{CLIP}	CLIP text encoder
F_p	Embedded features of the text
$t \in \{0 < t < T\}$	Time step
X_t	Latent space features of the image after t step iterative denoising
User	User - end
$CS = \{CS_1, CS_2, \ldots, CS_n\}$	n cloud service providers
U	U-Net network
$\mathbb{U} = \{\mathbb{U}_{center}, \mathbb{U}_{local}\}$	PPIU-Net network consists of the user-center and the cloud-local part.
$\{X_t^j, j \in (0, 1, \ldots, l)\}$	Feature data of the j layer of the U-Net in the t step denoising iteration.
$\{X_t^{(i,j)}, i \in (0, 1, \ldots, n), j \in (0, 1, \ldots, l)\}$	Feature data of the j layer in the i cloud's U-Net during the t-step denoising iteration.
C	Convolution layer calculation
$\{C_1, C_2, \ldots, C_n\}$	Sub-block data after partitioning the input data X_t^j for convolution.
$\{C_{\text{pad1}}, C_{\text{pad2}}, \ldots, C_{\text{padn}}\}$	Padded sub-block data for edge preservation during convolution.
$\{s_c^1, s_c^2, \ldots, s_c^n\}$	Output of convolution computations returned from local servers to the central server.
S_C	Reconstructed complete output data after aggregation, representing X_t^{j+1}.
A	Attention mechanism calculation
Q, K, V	Query, Key, and Value matrices for attention mechanism.
$\{Q^1, Q^2, \ldots, Q^n\}$	Sub-blocks of query matrix Q split according to the number of cloud servers.
S_A^i	Output of attention computation for sub-block i, represented as $A(Q^i, K^i, V^i)$.
S_A	Aggregated output of the attention mechanism, representing X_t^{j+1}.
G	Group normalization layer calculation
$\{G^1, G^2, \ldots, G^n\}$	Sub-blocks of the input data X_t^j after partitioning for group normalization.
$\{\mu_1, \mu_2, \ldots, \mu_n\}$	Mean of each sub-block for group normalization.
$\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2\}$	Variance of each sub-block for group normalization.
$\{\hat{G}^1, \hat{G}^2, \ldots, \hat{G}^n\}$	Normalized sub-blocks after group normalization, returned to the central server.
S_G	Reconstructed final output after global aggregation of the normalized sub-blocks, representing X_t^{j+1}.
Φ	Sub-block feature aggregation
D_{VAE}	VAE image decoder
I	Pixel-level generated image

generates $X_t \sim \mathcal{N}(0, I)$, and sends the F_p along with $X_t \sim \mathcal{N}(0, I)$ and other parameters to PPIU-Net, which consists of the user-side $\mathbb{U}_{\text{center}}$ and the cloud-side $\mathbb{U}_{\text{local}}$. These modules are composed of convolution layers $\text{layer}_{\text{conv}}$, attention layers $\text{layer}_{\text{Att}}$, Group normalization layer_{GN}, and other layers $\text{layer}_{\text{other}}$.

Step 2: For each timestep, $\mathbb{U}_{\text{net}}$ iteratively executes computations according to the network structure. First, $\mathbb{U}_{\text{center}}$ processes the input feature X_t^j, where $j \in \{0, 1, \ldots, L\}$, based on the current layer type. If the layer type is $\text{layer}_{\text{other}}$, the feature is normally processed. For $\text{layer}_{\text{conv}}$, $\text{layer}_{\text{Att}}$, and layer_{GN}, the corresponding features X_t^j are assigned for local computation.

Step 3: Each cloud server in $\{CS_1, CS_2, \ldots, CS_n\}$ independently computes the assigned local feature $X_t^{(i,j)}$, obtaining updated local features $X_t^{(i,j+1)}$, which are then returned to $\mathbb{U}_{\text{center}}$.

Step 4: $\mathbb{U}_{\text{local}}$ aggregates all results $X_t^{(i,j+1)}$ from the cloud servers to update the global feature for the current layer and produces the complete output X_t^{j+1}.

Step 5: Steps 2–4 are repeated until all layers of PPIU-Net have completed computation for the current timestep t, yielding the synchronized output $X_{t-1} = X_t^L$.

Step 6: Steps 2–5 are repeated for all timesteps to complete the iterative denoising process, ultimately generating the final latent spatial feature X_0.

Step 7: The user decodes the latent spatial feature X_0 through the image decoder D_{VAE}, generating the final high-quality image I.

PPIDM does not encrypt feature data, does not input complete initial parameters and features, and does not adjust the model's generation order, ensuring the quality of generated images. The core idea of PPIDM lies in the design of PPIU-Net, which adopts a network architecture combining a user-side aggregation center and multiple cloud-side computing centers. The complex computation layers and latent privacy risks in the U-Net network (such as convolution layers, attention layers, etc.) are distributed across different cloud servers. Each cloud server processes only part of the data, communicates results back to the user, and aggregates and updates the output. Even if a single cloud server is compromised, attackers would find it challenging to infer complete latent spatial features and obtain the full image-level information. This effectively protects user privacy. The overall PPIDM workflow is presented in Algorithm 1.

B. Lightweight User-Side Text Encoding

The model complexity and computational cost of the text encoder E_{CLIP} are significantly smaller than those of the U-Net network, making its computational requirements generally manageable on conventional devices. Moreover, this component can be directly executed locally to align with the decentralized computation design of diffusion models. For instance, the model size of the text encoder in SD1.5 is approximately 123MB. Encryption of the input text prompt is likely to exert a negative influence on the generated quality.

Algorithm 1 PPIDM Algorithm

Input: Prompt p, Step t.
Output: Generated image I.
User: Generates Gaussian noise matrix $X_t \sim \mathcal{N}(0, I)$.
User: Enters p into E_{CLIP} to get F_p.
for $t = T$ **to** 0 **do**
  for $j = 0$ **to** l **do**
    if layer $=$ Conv, Att, Gn **then**
      User: Segments $\{X_t^{(j)}, j \in (0, 1, \ldots, l)\}$ to $\{X_t^{(i,j)}, i \in (0, 1, \ldots, n), j \in (0, 1, \ldots, l)\}$ in $\mathbb{U}_{\text{center}}$, distributed to $\{CS_i, i \in (0, 1, \ldots, n)\}$.
      CS: Performs $U_{\text{local}}(X_t^{(i,j)}, F_p, t) \rightarrow X_t^{(i,j+1)}$, returned to the **User**.
      User: Aggregates all $X_t^{(i,j+1)}$ to get global results X_t^{j+1}.
    else
      User: Performs $\mathbb{U}_{\text{center}}(X_t^j, F_p, t) \rightarrow X_t^{j+1}$.
    end if
  end for
end for
User: Inputs X_0 to decoder D_{VAE} to generate image I.

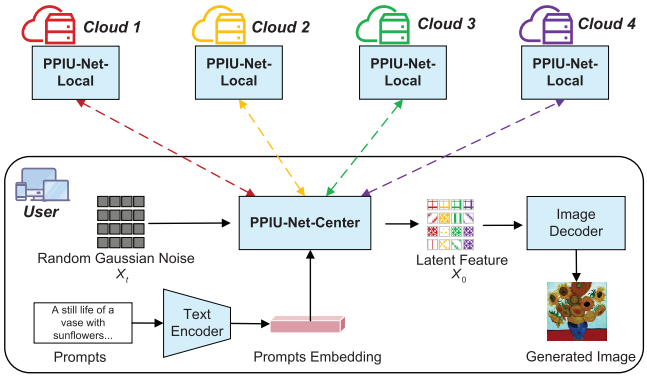


Fig. 3. Overview diagram of the PPIDM. For ease of understanding, we take four non-colluding clouds as an example. Details of the key components, including the Text Encoder, PPIU-Net, and Image Decoder, can be found in Sections IV-B, IV-C, and IV-D.

Consequently, within the PPIDM scheme, both the text and its corresponding encoder are processed by the user.

At this stage, users input a text prompt p into the text encoder E_{CLIP} to obtain a high-dimensional text embedding feature F_p. Only F_p is uploaded to the cloud server *CS*, thus ensuring privacy protection and optimizing the quality of downstream images.

C. High-Burden Multi-Cloud Collaboration: Privacy-Preserving Inference U-Net

In SD models, U-Net requires handling most parameters. Convolutional layers, attention layers, and normalization layers of U-Net consume significant computational resources and directly correlate with image quality features. Therefore, we design PPIU-Net, which includes PPIU-Net-Center $\mathbb{U}_{\text{center}}$ and multiple cloud-based PPIU-Net-Local $\mathbb{U}_{\text{local}}$ modules. $\mathbb{U}_{\text{center}}$ extracts and decomposes high-dimensional features into sub-features, distributing them to various cloud servers. Each cloud-local $\mathbb{U}_{\text{local}}$ processes a portion of the feature information, and returns the results to $\mathbb{U}_{\text{center}}$, which aggregates and updates the final complete feature representation. The structure of PPIU-Net is illustrated in Figure 4.

Procedure: For Gaussian noise and latent spatial feature X_t, $\mathbb{U}_{\text{center}}$ processes each layer based on the computational layer type. When the layer is computationally complex (*e.g.*, convolutional layer $layer_{\text{conv}}$, attention layer $layer_{\text{att}}$, or normalization layer $layer_{\text{gn}}$), $\mathbb{U}_{\text{center}}$ segments input features X_t^j into smaller subfeatures and distributes them across multiple servers $\{CS_i\}$ for parallel computation: $\{X_t^{(i,j)}, i \in (0, 1, \ldots, n), j \in (0, 1, \ldots, l)\}$, where $X_t^{(i,j)}$ is processed locally at each CS_i, using specific mechanisms for convolution *C*, attention *A*, and group normalization *G*. The output feature is represented as:

$$X_t^{(i,j+1)} = \begin{cases} C(X_t^{(i,j)}), & \text{layer} = \text{Conv}, \\ A(X_t^{(i,j)}), & \text{layer} = \text{Att}, \\ G(X_t^{(i,j)}), & \text{layer} = \text{GN}. \end{cases} \tag{2}$$

After receiving the computation results $X_t^{(i,j)}$ from all cloud servers, $\mathbb{U}_{\text{center}}$ performs a global aggregation operation Φ to obtain the computed results for the layer:

$$X_t^{j+1} = \Phi(X_t^{(1,j+1)}, X_t^{(2,j+1)}, \ldots, X_t^{(n,j+1)}), \tag{3}$$

where the outputs of all network layers are computed to X_t^l. The latent space feature X_0 is obtained after $\mathbb{U}_{\text{center}}$ and $\mathbb{U}_{\text{local}}$ cooperate to complete T de-noising iterations.

Specifically, the processes for privacy-preserving convolution, attention mechanism, and group normalization are as follows:

1) Privacy-Preserving Convolution (PPIConv): The computation process of PPIConv is divided into three steps.

Step 1: User-side data partitioning and padding. $\mathbb{U}_{\text{center}}$ partitions the input data X_t^j based on the number of cloud servers and creates sub-block data $\{C_1, C_2, \ldots, C_n\}$. For edge preservation during the convolution, padding operations are applied. The padded output data are denoted as $\{C_{\text{pad1}}, C_{\text{pad2}}, \ldots, C_{\text{padn}}\}$.

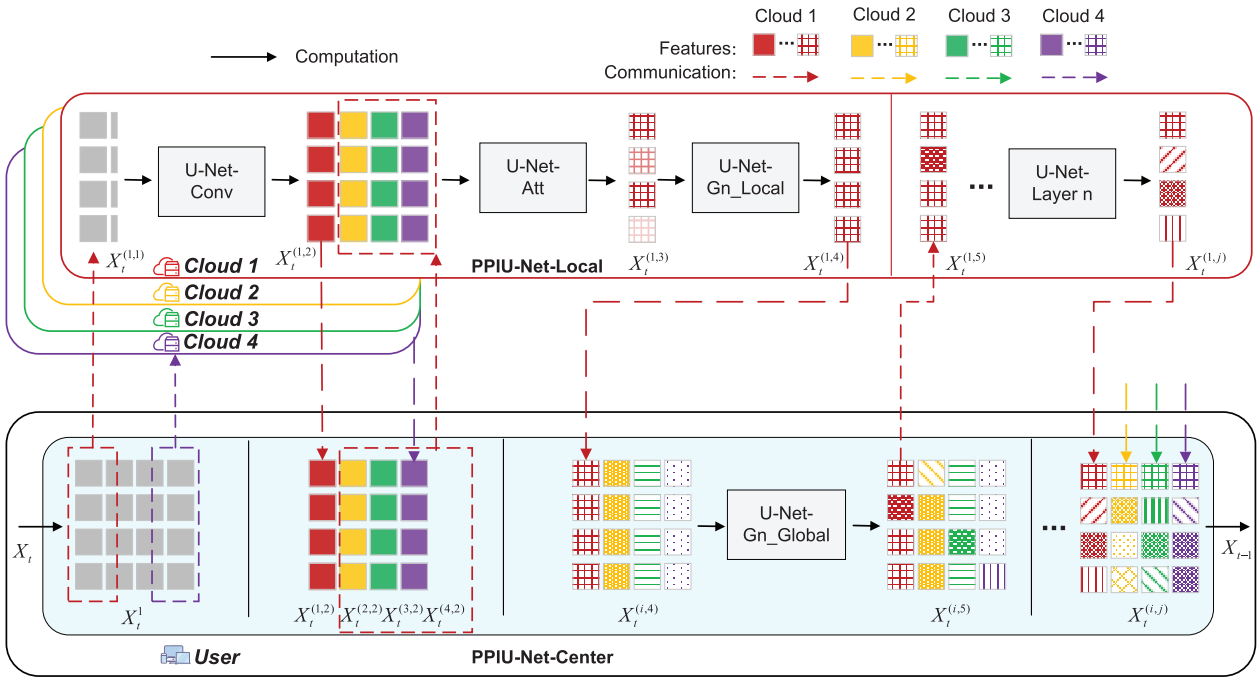


Fig. 4. Collaborative inference process of PPIU-Net with four cloud servers.

Step 2: Server-side block convolution computation. $\mathbb{U}_{\text{center}}$ distributes the padded sub-block data $\{C_{\text{pad1}}, C_{\text{pad2}}, \ldots, C_{\text{padn}}\}$ to multiple cloud servers $\mathbb{U}_{\text{local}}$, which perform convolution computations. The outputs $\{s_c^1, s_c^2, \ldots, s_c^n\}$ are then returned to $\mathbb{U}_{\text{center}}$.

Step 3: Global aggregation and updating. $\mathbb{U}_{\text{center}}$ aggregates all returned convolution results $\{s_c^1, s_c^2, \ldots, s_c^n\}$ to reconstruct the complete output data S_C, which is X_t^{j+1}.

2) Privacy-Preserving Attention Mechanism (PPIAtt): PPIAtt focuses on matrix computations involving query, key, and value matrices (QKV).

Step 1: User-side matrix splitting. $\mathbb{U}_{\text{center}}$ generates query matrix Q, key matrix K, and value matrix V based on the input data X_t^j. Then, $\mathbb{U}_{\text{center}}$ splits Q into sub-blocks $\{Q^1, Q^2, \ldots, Q^n\}$ according to the number of cloud servers.

Step 2: Server-side attention computation. $\mathbb{U}_{\text{center}}$ distributes the corresponding sub-blocks of key-value pairs (K, V) to the cloud servers CS_i, which execute the attention computation and return the results $S_A^i = \{A(Q^i, K^i, V^i), i = 1, 2, \ldots, n\}$.

Step 3: Global aggregation. The aggregated output S_A^i of the attention mechanism is represented as S_A, which is X_t^{j+1}.

3) Privacy-Preserving Group Normalization (Ppign): The core of PPIGN lies in the local computations of group statistics by the cloud servers.

Step 1: User-side data partitioning. $\mathbb{U}_{\text{center}}$ splits the input data X_t^j into sub-blocks $\{G^1, G^2, \ldots, G^n\}$ and distributes them to the servers CS_i.

Step 2: Server-side group normalization. Each $\mathbb{U}_{\text{local}}$ computes the mean $\{\mu_1, \mu_2, \ldots, \mu_n\}$ and variance $\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2\}$ of its sub-block, then normalizes the data and returns the outputs $\{\hat{G}^1, \hat{G}^2, \ldots, \hat{G}^n\}$ to $\mathbb{U}_{\text{center}}$.

Step 3: Global aggregation. $\mathbb{U}_{\text{center}}$ aggregates all returned normalized sub-blocks to compute the global mean and variance and reconstruct the final outputS_G, which is X_t^{j+1}.

D. Image Decoder

The decoder D_{VAE} of the VAE can similarly function as a component independent of the U-Net. Compared to the text encoder and U-Net network, it has fewer parameters (approximately 50M) and lower computational requirements. In this proposed solution, we choose to deploy the VAE on the client side to enhance the protection of latent features and image privacy. The client inputs the latent features X_0 into the VAE decoder D_{VAE} for decoding, where $D_{\text{VAE}}(X_t) > I$, generating pixel-level images I.

V. Experiments and Analysis

A. Experimental Setup

1) Experimental Platform: We conducted experiments on a machine equipped with an Intel(R) Core(TM) i7-14700KF processor, 64GB of memory, and an NVIDIA GeForce RTX 4090 GPU running the Windows operating system. The experiments were implemented in the PyTorch environment to validate the proposed PPIDM method.

2) Datasets: Considering the differences in diffusion models across artistic and general-purpose domains, experiments were conducted on two types of datasets: artistic images and general images. The detailed parameters of the datasets are as follows:

- **Art [37]:** Includes two categories of works, with a total of 1,100 prompts. The first category includes works by five renowned artists: Pablo Picasso, Van Gogh, Rembrandt, Andy Warhol, and Caravaggio, with 100 prompts in total. The second category includes works generated using

six publicly authorized SD models and similar styles, with artists including Kelly McKernan, Thomas Kinkade, Tyler Edlin, Kilian Eng, and the *"Demi-Human"* series. This category contains 1,000 prompts, with content primarily in painting styles.

- **Coco30k [38]:** Includes a variety of daily scenes and objects such as people, animals, and vehicles, with a total of 30,000 prompts. The content generated primarily resembles realistic styles.

3) Baseline: Since this paper focuses on evaluating privacy protection during the inference phase of diffusion models, the baselines are set as the regular-resolution Stable Diffusion (SD) model and the high-resolution Stable Diffusion XL (SDXL) [39] model.

4) Evaluation Metrics: The evaluation metrics used in this paper align with those of SDXL to assess the quality of the generated images. Comparisons are made against images generated by SD or SDXL under the same parameter settings. Specific evaluation metrics include:

- **PSNR (Peak Signal-to-Noise Ratio):** Measures the similarity between the generated image and the reference image. Higher values indicate better image quality.
- **SSIM (Structural Similarity):** Evaluates the structural similarity of images based on pixel values. Higher values indicate better structural similarity.
- **LPIPS (Learned Perception of Image Patch Similarity):** Uses deep learning models to assess the structural and perceptual similarity between the generated image and the reference image. Lower values indicate smaller perceptual errors.
- **FID (Fréchet Inception Distance):** Measures the distance between the distributions of generated images and real images. Lower values indicate better quality of the generated images.

B. Experimental Details

This solution primarily focuses on privacy protection during the inference phase and does not involve fine-tuning or training the model. The standard parameters during inference are as follows: SD Model Resolution: 512. SDXL Model Resolution: 1024. Number of Inference Steps: 50. Guidance Scale: 5. 0. Scheduler Method: DDIM [40]. Number of PPIDM Cloud Devices: 4. Any changes to the experimental setup will be detailed in the relevant sections.

C. Image Generation Quality Evaluation

In this section, we evaluate the quality of images generated by PPIDM, including Similarity evaluation between the images generated by PPIDM and the original model. Ablation experiments on parameters related to the quality of the generated images.

As shown in Table IV, we evaluated the proposed scheme PPIDM from the perspectives of datasets, the number of cloud devices, and models to assess the similarity between the generated images and those of the original model. Overall, PPIDM achieves high consistency with the original model across all metrics: the PSNR of the generated images exceeds 36.7, LPIPS is less than 0.017, FID is less than 4.8, and SSIM is greater than 0.96. These results indicate that PPIDM effectively preserves the image generation quality of the original model. Examples of the generated images are shown in Figure 5, where differences are nearly imperceptible to the naked eye.

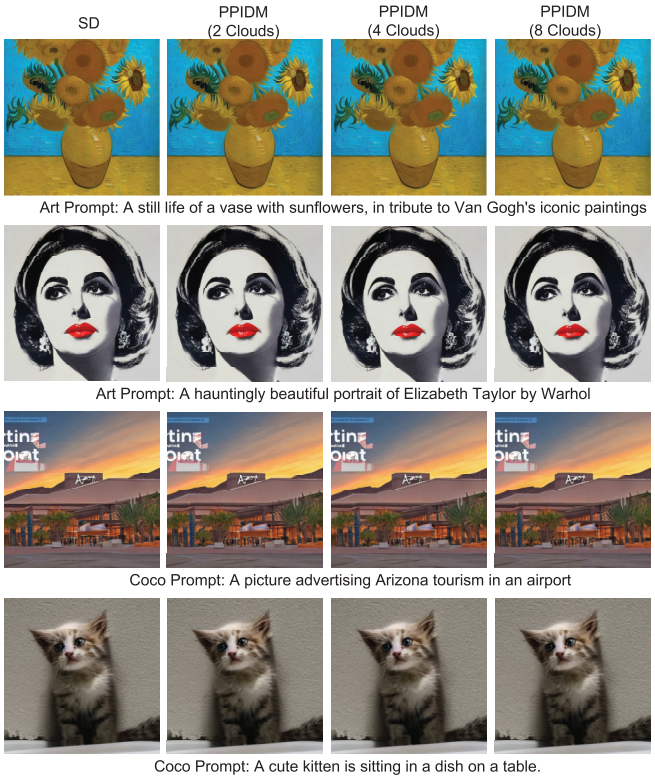


Fig. 5. Comparison of the images generated by SD and PPIDM on the Art and COCO datasets.

TABLE IV
IMAGE QUALITY EVALUATION OF PPIDM - GENERATED IMAGES

Dataset	Resolution	Method	Metrics			
			PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Art	512	SD	-	-	-	-
		Ours-2	36.80	0.968	0.018	4.73
		Ours-4	**36.90**	**0.969**	**0.017**	**4.56**
		Ours-8	36.76	0.968	0.018	4.74
	1024	SDXL	-	-	-	-
		Ours-2	39.48	0.974	0.014	2.64
		Ours-4	39.57	**0.975**	**0.013**	**2.42**
		Ours-8	**39.77**	**0.975**	**0.013**	2.53
Coco	512	SD	-	-	-	-
		Ours-2	37.27	0.979	0.014	0.159
		Ours-4	**37.31**	**0.979**	**0.014**	**0.159**
		Ours-8	37.12	0.978	0.014	0.164
	1024	SDXL	-	-	-	-
		Ours-2	38.90	0.977	0.014	0.102
		Ours-4	39.15	0.978	0.014	0.097
		Ours-8	**39.18**	**0.978**	**0.014**	**0.097**

Analyzing from the dataset perspective, PPIDM performs better on the standard dataset COCO compared to the artistic dataset Art across all metrics. This demonstrates that PPIDM achieves higher image generation quality for conventional content and realistic style images than for artistic oil painting images. This also suggests that PPIDM is more suitable for generation tasks in real-world scenarios. Notably, the FID

TABLE V

IMPACT OF PPIU-NET'S COMPUTATIONAL LAYERS ON GENERATED IMAGE QUALITY

Model	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
SD				
SD + Ppiconv	47.09	0.993	0.003	1.09
SD + Ppiatt	47.19	0.993	0.003	1.10
SD + Ppign	36.91	0.968	0.017	4.62
SD + Ppiconv + Ppiatt	**47.31**	**0.994**	**0.003**	**0.88**
SD + Ppiconv + Ppign	36.87	0.968	0.018	4.69
SD + Ppiatt + Ppign	36.87	0.968	0.018	4.62
PPIDM	36.90	0.969	0.017	4.56

metric shows a significant difference between the COCO and Art datasets, with COCO achieving a much better FID. This is likely because the COCO dataset has higher inter-class distinguishability, resulting in smaller distribution differences in the generated images.

In terms of the number of cloud devices, the overall results indicate that the number of devices has minimal impact on the quality of the generated images, with only slight changes in similarity. For 512×512 resolution images generated by the SD model, image quality improves as the number of devices increases from 2 to 4, achieving the best results with four devices, before declining when the number of devices increases to 8. This trend can be attributed to the fact that as the number of devices increases from 2 to 4, feature sub-blocks become smaller, increasing computational granularity and optimizing image quality. However, when the number of devices increases further to 8, the feature sub-blocks become too small, leading to insufficient information and a decline in quality. In contrast, for 1024×1024 resolution images generated by the SDXL model, the best quality is achieved with eight devices, and a decline point has not yet been observed.

From the model perspective, the standard resolution of the SD model is 512, while the standard resolution of the SDXL model is 1024. The results show that the SDXL model performs better at 1024 resolution than the SD model does at 512 resolution. This is attributable to both the superior network structure and parameters of the SDXL model, as well as the smaller proportion of segmentation boundaries at higher resolutions, which further improves overall similarity.

D. Ablation Study

First, we conducted ablation experiments on the complex computational layers in PPI-Unet to evaluate the impact of different computational layers on the performance of the proposed scheme. The experimental results are shown in Table V, with parameters set as follows: the dataset is Art, the model is SD, the resolution is 512, the number of iterations is 50, and the number of cloud devices is 4.

As shown in Table V, when the PPIConv and PPIAtt layers are modified independently in the SD model, the metrics of the generated images are superior to those of PPIDM. This indicates that the convolutional and attention layers have a relatively small impact on generation quality. However, after modifying the PPI GN layer, the quality metrics of the generated images deteriorated, indicating that the feature

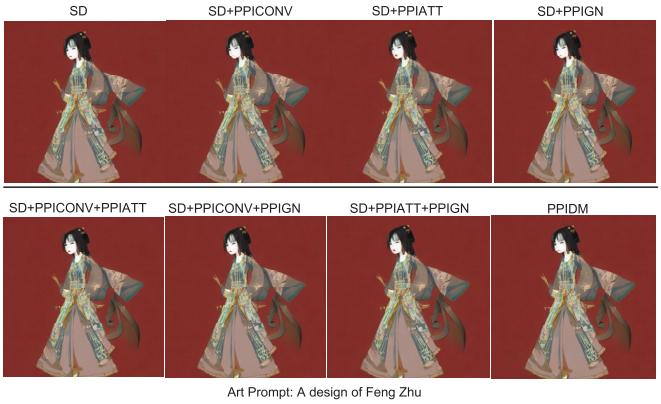


Fig. 6. Impact of PPIU-Net's computational layers on generated image quality.

TABLE VI

THE IMPACT OF TIME-STEP PARAMETERS ON GENERATED IMAGE QUALITY

Time-step	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
30	**37.06**	**0.97**	**0.017**	**4.67**
40	37.04	0.969	0.017	4.46
50	36.90	0.969	0.017	4.56
60	36.45	0.967	0.019	4.74
70	36.48	0.966	0.019	4.82

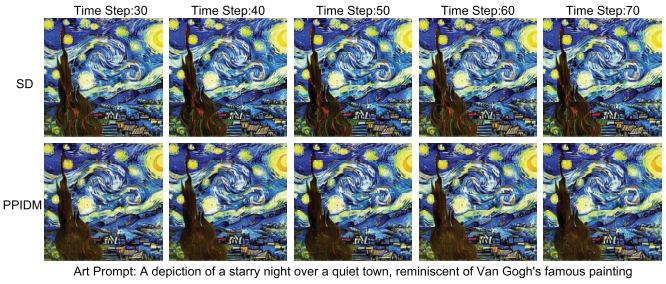


Fig. 7. Impact of time steps on generated image quality.

partitioning and computation in the PPI GN layer introduced significant losses. This degradation occurs because, unlike the conventional GN in SD, PPIDM reduces computational costs on the user side by decomposing global GN into multiple local GN computations, which are later aggregated. Consequently, this process inevitably results in some precision loss. It's a trade-off between precision and computational cost. Overall, PPIDM, which integrates all three computational layers, shows no major shortcomings in similarity metrics, indicating a balanced similarity between the generated images and the original images across various aspects. No significant differences are discernible to the naked eye. Examples of the generated images are shown in Figure 6.

Next, we evaluated the similarity between the images generated by PPIDM and SD under different numbers of inference steps. The experimental results are shown in Table VI.

From Table VI, it can be observed that as the number of inference steps increases, the PSNR, LPIPS, FID, and SSIM metrics between PPI and SD-generated images degrade. This is because the cumulative effect of feature differences

TABLE VII
THE IMPACT OF RESOLUTION ON THE QUALITY OF GENERATED IMAGES

Resolution	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
256	31.42	0.925	0.045	12.66
512	36.90	0.969	**0.017**	4.56
768	38.06	0.970	0.019	**4.49**
1024	**39.08**	**0.972**	0.020	4.54

Fig. 8. Impact of resolution on generated image quality.

becomes more pronounced with more iterations, leading to a gradual decrease in the similarity between the generated images and those of the original model. Image instances generated by different Time steps are shown in Figure 7. As the number of Time steps increases, the difference between the images generated by PPIDM and SD remains minimal, with no perceptible distinction observable in Figure 7.

Finally, we evaluated the quality of images generated by PPIDM under different resolutions. The results are shown in Table VII.

From Table VII, it can be seen that as the resolution increases, the PSNR and SSIM metrics between PPISD and SD-generated images gradually improve. This is because, at higher resolutions, the proportion of feature data slicing to the overall features becomes smaller, reducing the impact of slicing on the pixel quality of the images. On the other hand, the LPIPS and FID metrics do not exhibit a monotonic trend with resolution changes. This is due to the SD model's standard resolution being 512, and images generated at resolutions of 256 or 1024 may have missing or repetitive content, introducing some disturbance to the deep perception metrics. Example effects of generated images with different resolutions are shown in Figure 8.

As shown in Figure 8, only the original training resolution of 512 effectively generates images that align well with the prompt semantics, while resolutions of 256, 768, and 1024 exhibit suboptimal performance. The similarity between images generated by PPIDM and SD remains high.

E. Image Generation Time Overhead

In this section, we discuss the time overhead for image generation in the proposed scheme, including both communication and computation costs.

Currently, cloud servers and user-end devices are typically equipped with high-bandwidth communication capabilities.

TABLE VIII
CALCULATION OF COMMUNICATION DATA SIZES FOR CONVOLUTIONAL LAYERS

Number of Devices	Conv Up(MB)			Conv Down(MB)		
	Min	Max	Total	Min	Max	Total
1	-	-	-	-	-	-
2	0.23	7.96	85.23	0.031	5	55.18
4	**0.15**	**4.21**	**47.03**	**0.016**	**2.5**	**27.59**
8	0.11	2.34	27.92	0.008	1.25	13.79

Assuming the user's mobile device supports 5G communication, the average downlink speed of China's 5G network in Q4 2023 is 340.56 Mbps, and the uplink speed is 81.14 Mbps [41]. Therefore, we use this as the communication bandwidth assumption for the user device. Each cloud server is assumed to have a high-speed fiber connection, far exceeding the bandwidth of user devices. Hence, the primary focus is on the communication performance of the user device. First, we calculate the size of the data to be communicated and then estimate the required transmission time based on the communication bandwidth. In the PPIDM scheme, communication and computation costs mainly occur in the convolutional layers, attention layers, and normalization layers. Below, we analyze the time consumption of these computational layers.

1) Time Overhead for Convolutional Layers: For convolutional layers, we calculated the input and output data sizes of each convolutional layer per iteration for generating a single 512×512 resolution image in each cloud device. Additionally, we computed the maximum and minimum communication costs for a single convolutional layer upload/download and the total communication costs for all convolutional layers, as shown in Table VIII.

From Table VIII, it can be observed that as the number of cloud devices increases, the convolutional data size processed by each cloud device decreases linearly. In a scenario with 4 cloud devices, the user's communication overhead per cloud is as follows: the minimum upload data size for a single convolutional layer is 0.15 MB, and the maximum upload data size is 4.21 MB. After completing one iteration of denoising, the total upload data size for all convolutional layers is 47.03 MB. Similarly, the minimum download data size for a single convolutional layer is 0.016 MB, and the maximum download data size is 2.5 MB, with a total download data size of 27.59 MB for all layers.

The total communication cost between the user and 4 cloud devices is as follows: a total upload size of 188.12 MB and a total download size of 110.37 MB. In a 5G network environment, the theoretical upload time is 10.88 seconds, and the download time is 2.59 seconds. Since upload and download can occur simultaneously, the theoretical time overhead for convolutional layers in one round of denoising is approximately 10.88 seconds. Regarding computation time, our device (NVIDIA 4090) achieves far lower computation times than the aforementioned communication times, and the multi-cloud device setup further reduces the required time. Additionally, the computation cost can be pipelined into the communication cost, making computation time negligible.

TABLE IX
CALCULATION OF COMMUNICATION DATA SIZES FOR ATTENTION LAYERS

Number of Devices	Att Up(MB)			Att Down(MB)		
	Min	Max	Total	Min	Max	Total
1	-	-	-	-	-	-
2	0.90	2.68	29.36	0.15	2.5	22.03
4	**0.83**	**1.43**	**18.34**	**0.07**	**1.25**	**11.01**
8	0.68	0.81	12.83	0.03	0.625	5.50

TABLE X
CALCULATION OF COMMUNICATION DATA SIZES FOR GROUPNORM LAYERS

Number of Devices	GN Up(MB)			GN Down(MB)		
	Min	Max	Total	Min	Max	Total
1	-	-	-	-	-	-
2	0.15	7.5	171.87	0.0002	0.0002	0.059
4	**0.07**	**3.75**	**85.93**	**0.0002**	**0.0002**	**0.059**
8	0.03	1.87	42.96	0.0002	0.0002	0.059

Thus, the total image generation time can be considered equal to the communication time, which is 10.88 seconds.

2) Time Overhead for Attention Layers: For attention layers, we calculated the communication data size in the same manner as for convolutional layers. The results are shown in Table IX.

From Table IX, as the number of cloud devices increases, the attention layer data size processed by each device decreases, but the reduction is less significant compared to convolutional layers because the key-value (K-V) components are not split; only the query (Q) component is divided. In the standard setup with 4 devices, the minimum upload data size for a single attention layer is 0.830 MB, and the maximum upload data size is 1.437 MB. After completing one iteration of denoising, the total upload size for all attention layers is 18.347 MB. Similarly, the minimum download data size for a single attention layer is 0.078 MB, and the maximum download data size is 1.25 MB, with a total download size of 11.01 MB for all layers.

The total communication cost between the user and 4 cloud devices is as follows: a total upload size of 73.38 MB and a total download size of 44.06 MB. In a 5G network environment, the theoretical upload time is 7.23 seconds, and the download time is 0.13 seconds. As with convolutional layers, the computation time for attention layers is negligible compared to the communication time. Thus, the theoretical time overhead for attention layers in one round of denoising is approximately 7.23 seconds.

3) Time Overhead for Group Normalization Layers: For group normalization layers, the communication data statistics and computation results are shown in Table X.

From Table X, as the number of cloud devices increases, the data uploaded by the user to each cloud device for group normalization proportionally decreases, while the download data size remains constant. In the standard setup with 4 devices, the minimum upload data size for a single group normalization layer is 0.07 MB, and the maximum upload data size is 3.75 MB, with a total upload size of 85.93 MB. The download data size for a single layer is 0.0002 MB, and the total download size is 0.059 MB.

The total communication cost between the user and 4 cloud devices is as follows: a total upload size of 343.74 MB and a total download size of 0.23 MB. In a 5G network environment, the theoretical total upload time is 33.89 seconds, and the total download time is 0.005 seconds. As with convolutional and attention layers, the computation time for group normalization layers is negligible compared to the communication time. Thus, the theoretical time overhead for group normalization layers in one round of denoising is 33.89 seconds.

4) Overall Time Overhead: On an NVIDIA 4090 device under a 5G communication environment, the total communication and computation time overhead for convolutional, attention, and group normalization layers between the user and the cloud servers is 52 seconds. Our scheme balances the user's computational resources, the privacy protection of cloud data, and the quality of the generated images, which inevitably results in high communication costs.

VI. SECURITY ANALYSIS

In this section, we analyze the security of the proposed PPIDM from the perspectives of input text and generated images.

A. Input Text Privacy Protection

Input text represents the semantic intent of the user's desired generated image and is part of the user's intellectual property. Therefore, its privacy protection is crucial. During the inference process, the input text prompt is transformed into high-dimensional text embedding features through a text encoder. The reverse parsing of high-dimensional text embeddings presents significant challenges [42], making it difficult to reconstruct the original text. Key challenges in reversing high-dimensional text embeddings include:

- Lossy Compression: The text embedding process discards fine-grained details, making exact text recovery mathematically infeasible.
- High Dimensionality: The vast and sparse nature of the embedding space adds computational and theoretical barriers to reverse parsing.
- Semantic Abstraction: The embeddings encode high-level semantics rather than specific lexical or syntactic details, further complicating recovery.

Moreover, various defense mechanisms have been developed to enhance the security of text features, such as embedding perturbation, dimensionality expansion, and differential privacy. Given the maturity of these techniques, our work primarily focuses on images security challenges in the inference phase of diffusion models.

B. Generated Image Security

The security of the generated images in PPIDM is enhanced through techniques such as feature partitioning and the denoising process of diffusion models, which significantly reduce the effective information entropy available to attackers. Even if an attacker obtains some feature sub-blocks, the recovered image information remains vague and fragmented, making it difficult to reconstruct the original plaintext image. Below, we conduct a detailed security analysis of the scheme.

1) Entropy Limitation of Single Cloud Servers: The image features are divided into $N \times P$ sub-blocks, distributed across N independent cloud servers. The information entropy (denoted by $H(\cdot)$) of each subblock is $H(Block) = \frac{H_{\text{Image}}}{N \times P}$.

The information entropy accessible to attackers from a single cloud server is:

$$H_{\text{Attacker}} = H_{\text{Single}} = \sum_{i=1}^{P} H(Block_i) \tag{4}$$

The information entropy required to reconstruct a complete image is:

$$H_{\text{Reconstruction}} = \sum_{i=1}^{N \times P} H(B_i) + H(Arr) \tag{5}$$

where *Arr* denotes the arrangement of the sub-blocks, $H(Arr) \neq 0$. The single-cloud server has only part of the feature block information and cannot provide $H(Arr)$, which makes the attacker unable to achieve global recovery.

2) Joint Effect of Noise Protection: In the T-step iteration of the diffusion model, the features of the first $T-1$ steps are dominated by noise σ. Even if attackers obtain complete features, their effective information is weakened. The entropy of sub-block features stored on a single cloud server is:

$$\begin{aligned} H_{\text{Single}} &= \frac{H(Feature_t)}{N} \\ &= \frac{H(Feature_T) - (T-1-t)H(\sigma)}{N} \end{aligned} \tag{6}$$

where $H(\sigma)$ represents the noise entropy. When $t \neq T-1$, i.e., the denoising step is not fully completed, the intermediate features always contain noise. However, the attacker can only get the intermediate features because the final denoising is done by the client. This noise significantly reduces the quality of plaintext image reconstruction through standard decoders.

3) Complexity of Sub-Block Arrangement: The completeness of the image depends on the spatial arrangement of $N \times P$ sub-blocks. The arrangement information entropy is:

$$H(Arr) = \log((N \times P)!) \tag{7}$$

From the attacker's perspective, no arrangement information entropy can be obtained, i.e., $H(Arr) = 0$. The lack of global sub-block arrangement information further limits the possibility of image reconstruction.

Therefore, the possession of only partial feature blocks, along with noise interference and the absence of sub-block arrangement information, prevents a single cloud or attacker from reconstructing the complete image, thereby ensuring the security of the generated images in PPIDM.

VII. Discussion

In this section, we discuss the future development of privacy protection for image generation using diffusion models in cloud environments. As diffusion models are applied to various tasks and scenarios, privacy protection techniques need continuous development and optimization. Below, we categorize and summarize potential future directions from different perspectives:

Reducing communication costs is a key challenge for optimizing the efficiency of privacy protection methods. For security purposes, our proposed scheme requires frequent data transmission, resulting in high communication costs. In the future, methods such as optimizing data transmission protocols, compressing feature data, and reducing redundancy can be employed to decrease communication costs, thereby improving the efficiency of collaborative computing in the cloud. These improvements will enhance the real-time performance of generation tasks, reduce the reliance of cloud-based generation tasks on user bandwidth, and make privacy protection methods in cloud computing more efficient and practical.

Multi-user collaborative generation tasks represent another privacy protection challenge that needs to be addressed. In scenarios where multiple users simultaneously use diffusion models, it is crucial to protect the data privacy of each user during the collaborative computing process. This will further enhance the productivity of diffusion models and increase users' trust in the system. A suitable approach may involve combining other advanced privacy protection techniques, such as Secure Multi-Party Computation (MPC) and Differential Privacy. By incorporating these technologies, it is possible to ensure that data from all parties remains secure during collaborative computation and to add noise to the generation process for further privacy protection. Additionally, integrating watermarking techniques can embed invisible watermarks into the generated images, enabling tracking and identification of the source of generated images, thereby enhancing the level of privacy protection [43].

Beyond text-to-image generation, other diffusion model tasks, such as image-to-image generation and image editing, face similar privacy challenges. These tasks often involve processing user input images or editing content, which may contain sensitive data. Extending our method to cover these tasks will help effectively protect user privacy during image transformation or editing processes. Furthermore, as diffusion models are increasingly applied to multimodal tasks (e.g., text-to-video generation) [44], [45], privacy protection issues will become more complex. In the future, privacy protection mechanisms must address the safeguarding of multimodal data to ensure the security of all types of data privacy.

VIII. Conclusion

In this paper, we study the privacy protection problem of image generation during the inference phase of diffusion models in a cloud environment for the first time. We find that this task is characterized by the denoising-encryption adversarial nature and the stepwise generation property of diffusion models, which makes existing inference privacy protection schemes unsuitable for diffusion models. To address this, we propose a privacy-preserving diffusion model inference framework (PPIDM). On the one hand, this framework offloads complex computational layers from the user side to the cloud, significantly reducing the computational overhead on the user side. On the other hand, by partitioning feature data and distributing it to multiple non-colluding cloud servers for independent computation, the system's security is enhanced. We conducted a comprehensive evaluation of PPIDM on both

regular and artistic datasets. The results demonstrate that PPIDM achieves a good balance among user-side overhead, generation quality, and data security. We hope that PPIDM will drive advancements in the field of privacy protection for diffusion models.

REFERENCES

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.

[2] M. A. Shabani, Z. Wang, D. Liu, N. Zhao, J. Yang, and Y. Furukawa, "Visual layout composer: Image-vector dual diffusion model for design layout generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 9222–9231.

[3] Stability AI. (2024). *Stability AI Secures Significant New Investment From World-Class Investor Group and Appoints Prem Akkaraju as CEO*. Accessed: Nov. 2024. [Online]. Available: https://stability.ai/news/stability-ai-secures-significant-new-investment

[4] M. Li et al., "DistriFusion: Distributed parallel inference for high-resolution diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 7183–7193.

[5] Y. Li et al., "SnapFusion: Text-to-image diffusion model on mobile devices within two seconds," in *Proc. 37th Int. Conf. Neural Inf. Process. Syst.*, 2024, pp. 20662–20678.

[6] Stability AI. (2024). *Our Partners*. Accessed: Nov. 2024. [Online]. Available: https://stability.ai/partners

[7] C. Wang et al., "PAI-Diffusion: Constructing and serving a family of open Chinese diffusion models for text-to-image synthesis on the cloud," 2023, *arXiv:2309.05534*.

[8] Y. Que, L. Xiong, W. Wan, X. Xia, and Z. Liu, "Denoising diffusion probabilistic model for face sketch-to-photo synthesis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 10, pp. 10424–10436, Oct. 2024.

[9] Z. Liu, L. Hu, T. Zhou, Y. Tang, and Z. Cai, "Prevalence overshadows concerns? Understanding Chinese users' privacy awareness and expectations towards LLM-based healthcare consultation," in *Proc. IEEE Symp. Secur. Privacy*, Dec. 2025, p. 92.

[10] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 3813–3824.

[11] Z. Xu et al., "MagicAnimate: Temporally consistent human image animation using diffusion model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 1481–1490.

[12] Chinacourt. (2024). *IceBreaker: The First AI Vincennes Pattern to Work*. Accessed: Nov. 2024. [Online]. Available: https://www.chinacourt.org/article/detail/2024/02/id/7796864.shtml

[13] Google Cloud. (2024). *UNC5537 Targets Snowflake Customer Instances for Data Theft and Extortion*. Accessed: Nov. 2024. [Online]. Available: https://cloud.google.com/blog/topics/threat-intelligence/unc5537-snowflake-data-theft-extortion

[14] K. Li, C. Gong, Z. Li, Y. Zhao, X. Hou, and T. Wang, "PrivImage: Differentially private synthetic image generation using diffusion models with semantic-aware pretraining," in *Proc. 33rd USENIX Secur. Symp. (USENIX Secur.)*, 2024, pp. 4837–4854.

[15] M. Yang, S. Su, B. Li, and X. Xue, "Exploring one-shot semi-supervised federated learning with a pre-trained diffusion model," 2023, *arXiv:2305.04063*.

[16] P. Zhang et al., "Privacy-preserving and outsourced multi-party K-means clustering based on multi-key fully homomorphic encryption," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 3, pp. 2348–2359, May 2023.

[17] P. Yu, J. Tang, Z. Xia, Z. Li, and J. Weng, "A privacy-preserving JPEG image retrieval scheme using the local Markov feature and bag-of-words model in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 11, no. 3, pp. 2885–2896, Mar. 2023.

[18] Z. Wang, J. Qin, X. Xiang, and Y. Tan, "Privacy-preserving image retrieval based on disordered local histograms and vision transformer in cloud computing," *Int. J. Intell. Syst.*, vol. 2023, no. 1, pp. 1–16, Jan. 2023.

[19] Y. Li, J. Ma, Y. Miao, Y. Wang, X. Liu, and K.-K. R. Choo, "Similarity search for encrypted images in secure cloud computing," *IEEE Trans. Cloud Comput.*, vol. 10, no. 2, pp. 1142–1155, Feb. 2022.

[20] D. Li, W. Xie, Z. Wang, Y. Lu, Y. Li, and L. Fang, "FedDiff: Diffusion model driven federated learning for multi-modal and multi-clients," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 10, pp. 10353–10367, Oct. 2024.

[21] H. Wang, S. Pang, Z. Lu, Y. Rao, Y. Zhou, and M. Xue, "DP-Promise: Differentially private diffusion probabilistic models for image synthesis," in *Proc. 33rd USENIX Secur. Symp. (USENIX Secur.)*, Philadelphia, PA, USA, 2024, pp. 1063–1080.

[22] Z. Liu, L. Hu, Z. Cai, X. Liu, and Y. Liu, "SeCoSe: Toward searchable and communicable healthcare service seeking in flexible and secure EHR sharing," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 4999–5014, 2024.

[23] Y. Guo, F. Liu, T. Zhou, Z. Cai, and N. Xiao, "Privacy vs. efficiency: Achieving both through adaptive hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 4, pp. 1331–1342, Apr. 2023.

[24] Y. L. Tun, C. M. Thwal, J. S. Yoon, S. M. Kang, C. Zhang, and C. S. Hong, "Federated learning with diffusion models for privacy-sensitive vision tasks," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Oct. 2023, pp. 305–310.

[25] M. de Goede, B. Cox, and J. Decouchant, "Training diffusion models with federated learning," 2024, *arXiv:2406.12575*.

[26] J. Vora, N. Bouacida, A. Krishnan, and P. Mohapatra, "FedDM: Enhancing communication efficiency and handling data heterogeneity in federated diffusion models," 2024, *arXiv:2407.14730*.

[27] N. Carlini et al., "Extracting training data from diffusion models," in *Proc. 32nd USENIX Conf. Secur. Symp.*, Jan. 2023, pp. 5253–5270.

[28] T. Dockhorn, T. Cao, A. Vahdat, and K. Kreis, "Differentially private diffusion models," in *Proc. Trans. Mach. Learn. Res.*, Jan. 2023, pp. 1–44. [Online]. Available: https://openreview.net/forum?id=ZPpQk7FJXF

[29] S. Ghalebikesabi et al., "Differentially private diffusion models generate useful synthetic images," 2023, *arXiv:2302.13861*.

[30] M. F. Liu, S. Lyu, M. Vinaroz, and M. Park, "DP-LDMs: Differentially private latent diffusion models," 2023, *arXiv:2305.15759*.

[31] S. Yu et al., "How to construct corresponding anchors for incomplete multiview clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 4, pp. 2845–2860, Apr. 2024.

[32] Z. Wang, J. Qin, X. Xiang, Y. Tan, and J. Peng, "A privacy-preserving cross-media retrieval on encrypted data in cloud computing," *J. Inf. Secur. Appl.*, vol. 73, Mar. 2023, Art. no. 103440.

[33] P. Bunn and R. M. Ostrovsky, "Secure two-party k-means clustering," in *Proc. 14th ACM Conf., Comput. Commun. Secur.,*, Oct. 2007, pp. 486–497, doi: 10.1145/1315245.1315306.

[34] D. Liu, E. Bertino, and X. Yi, "Privacy of outsourced k-means clustering," in *Proc. 9th ACM Symp. Inf., Comput. Commun. Secur.*, Jun. 2014, pp. 123–134.

[35] W. Wu, J. Liu, H. Wang, J. Hao, and M. Xian, "Secure and efficient outsourced k-Means clustering using fully homomorphic encryption with ciphertext packing technique," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 10, pp. 3424–3437, Oct. 2021.

[36] R. Bellafqira, G. Coatrieux, D. Bouslimi, and G. Quellec, "Content-based image retrieval in homomorphic encryption domain," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2015, pp. 2944–2947.

[37] R. Gandikota, J. Materzynska, J. Fiotto-Kaufman, and D. Bau, "Erasing concepts from diffusion models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 2426–2436.

[38] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2014, pp. 740–755.

[39] D. Podell et al., "SDXL: Improving latent diffusion models for high-resolution image synthesis," in *Proc. 12th Int. Conf. Learn. Represent.*, 2024, pp. 1–8. [Online]. Available: https://openreview.net/forum?id=di52zR8xgf

[40] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=St1giarCHLP

[41] World Wide Web. (2024). *The National Average 5G Downlink and Uplink Speeds Reached 340.56M and 81.14M*. Accessed: Nov. 2024. [Online]. Available: https://finance.sina.cn/2024-04-07/detail-inaqyhsa8799944.d.html

[42] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Apr. 2015, pp. 1–5.

[43] Y. Luo, T. Zhou, S. Cui, Y. Ye, F. Liu, and Z. Cai, "Fixing the double agent vulnerability of deep watermarking: A patch-level solution against artwork plagiarism," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 3, pp. 1670–1683, Mar. 2024.

[44] Q. Liu et al., "Mitigating cross-modal retrieval violations with privacy-preserving backdoor learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 3, pp. 2526–2540, Mar. 2025.
[45] Y. Shi, J. Xi, D. Hu, Z. Cai, and K. Xu, "RayMVSNet++: Learning ray-based 1D implicit fields for accurate multi-view stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13666–13682, Jul. 2023.

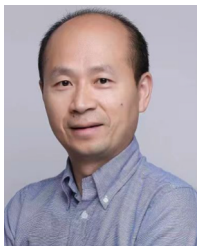
Zhangdong Wang received the bachelor's degree in communication engineering and the master's degree in information and communication engineering from the Central South University of Forestry and Technology, Changsha, China, in 2019 and 2023, respectively. He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, National University of Defense Technology. His research interests include deep learning, information security, and multimedia security.

Zhihuang Liu received the B.E. and M.S. degrees from the College of Computer and Data Science, Fuzhou University, in 2020 and 2023, respectively. He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, National University of Defense Technology. His research interests include privacy protection, applied cryptography, and privacy in generative AI.

Yuanjing Luo received the Ph.D. degree in computer science and technology from the National University of Defense Technology, Changsha, China, in 2024. She is currently a Lecturer with the College of Computer and Mathematics, Central South University of Forestry and Technology, Changsha. Her research has been published in multiple top-tier conferences and journals, such as WWW, AAAI, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. Her research interests include information security, especially image steganography, digital watermarking, and copyright protection in deep learning.

Tongqing Zhou received the B.S., M.S., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), Changsha, in 2012, 2014, and 2018, respectively. He is currently an Assistant Researcher with the College of Computer, NUDT. His main research interests include network measurement, crowd sensing, and data privacy. He was a recipient of the Outstanding Ph.D. Dissertation Award and Outstanding Post-Doctoral Award both of Hunan, China.

Jiaohua Qin received the B.S. degree in mathematics from Hunan University of Science and Technology, China, in 1996, the M.S. degree in computer science and technology from the National University of Defense Technology, China, in 2001, and the Ph.D. degree in computing science from Hunan University, China, in 2009. She was a Visiting Professor with the University of Alabama, Tuscaloosa, AL, USA, from 2016 to 2017. She is currently a Full Professor with the College of Computer Science and Information Technology, Central South University of Forestry and Technology, China. Her research interests include networks and information security and machine learning and image processing.



Zhiping Cai (Member, IEEE) received the B.Eng., M.A.Sc., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), China, in 1996, 2002, and 2005, respectively. He is currently a Full Professor with the College of Computer, NUDT. His current research interests include artificial intelligence, network security, and big data. He is a Senior Member of CCF.