ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3315-6701-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICASSP55912.2026.11465042

EDGE COLLABORATIVE GAUSSIAN SPLATTING WITH INTEGRATED RENDERING AND COMMUNICATION

Yujie Wan[1,2], *Chenxuan Liu*[2], *Shuai Wang*[2,*], *Tong Zhang*[3], *James Jianqiao Yu*[3], *Kejiang Ye*[2], *Dusit Niyato*[4], *Chengzhong Xu*[5]

[1]Southern University of Science and Technology
[2]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[3]Harbin Institute of Technology, Shenzhen [4]Nanyang Technological University [5]University of Macau

ABSTRACT

Gaussian splatting (GS) struggles with degraded rendering quality on low-cost devices. To address this issue, we present edge collaborative GS (ECO-GS), where each user can switch between a local small GS model to guarantee timeliness and a remote large GS model to guarantee fidelity. However, deciding how to engage the large GS model is nontrivial, due to the interdependency between rendering requirements and resource conditions. To this end, we propose integrated rendering and communication (IRAC), which jointly optimizes collaboration status (i.e., deciding whether to engage large GS) and edge power allocation (i.e., enabling remote rendering) under communication constraints across different users by minimizing a newly-derived GS switching function. Despite the nonconvexity of the problem, we propose an efficient penalty majorization minimization (PMM) algorithm to obtain the critical point solution. Furthermore, we develop an imitation learning optimization (ILO) algorithm, which reduces the computational time by over 100x compared to PMM. Experiments demonstrate the superiority of PMM and the real-time execution capability of ILO.

***Index Terms*—** Edge intelligence, Gaussian splatting

1. INTRODUCTION

Gaussian splatting (GS) is an emerging paradigm for 3D reconstruction [1]. However, when adapting GS to low-cost mobile devices, resource limitation necessitates model quantization or pruning, which degrades the rendering performance [2, 3]. To address the issue, a promising solution is edge collaborative GS (ECO-GS), where users can upload the pose sequences to a proximal edge server and render the corresponding images using edge GPUs [2–6].

The main obstacle to realizing ECO-GS is determining the collaboration timing for engaging the edge GS. First, collaboration may not be beneficial if the queried poses are not at the "pain points" (i.e., poses with significant discrepancies between rendered images and the actual environments) of the local GS model [7]. Second, collaboration involves periodic data exchange between user and server. Communication delays may lead to non-smooth rendering [2]. Existing edge collaboration schemes [2, 4–6, 8–13] fail in addressing the above challenges, since their objective function cannot account for the non-uniform GS discrepancies, and ignore the inter-dependency between rendering requirements and resource conditions.

To fill the gap, this paper proposes an integrated rendering and communication (IRAC) framework for ECO-GS systems, which enables users to adaptively switch between a small GS model executed locally to guarantee timeliness and a large GS model executed non-locally to enhance fidelity. Specifically, we propose a novel GS switching model to distinguish the heterogeneous view discrepancies of different GS models. The new model serves as objective function and is integrated with the communication resource constraints for cross-layer IRAC optimization, enabling us to maximize the information gain brought by collaboration. Next, the IRAC problem involves both binary edge-device collaboration and continuous power control variables. Existing methods [11, 13–17] may lead to ineffective solutions, since they involve continuous relaxation of discrete variables. To tackle the challenge, we propose penalty majorization minimization (PMM) and imitation learning optimization (ILO) methods to solve the problem for high-quality real-time IRAC designs. Finally, experimental results demonstrate the superiority of IRAC-based ECO-GS over benchmark methods in various scenarios.

2. PROBLEM FORMULATION

We consider an ECO-GS system shown in Fig. 1, which consists of an edge server with N antennas and K single-antenna users ($N \gg K$). The ECO-GS system aims to render RGB images $\{\mathbf{v}_k \in \mathbb{R}^{3LW}\}$ at all K users given their 6D view poses $\{\mathbf{s}_k \in \mathbb{R}^6\}$, where L and W are the length and width of camera images, respectively [6]. For each $\mathbf{v}_k$, it can be either rendered by a local GS model $\Phi_k(\cdot)$, or a remote edge GS model $\Phi_{\text{edge}}(\cdot)$. A binary vector $\mathbf{x} = [x_1, \cdots, x_K]^T$ with $x_k \in \{0, 1\}$ is introduced, where $x_k = 1$ indicates that the k-th user adopts edge GS, and $x_k = 0$ otherwise. The output of ECO-GS is thus

$$\mathbf{v}_k = (1 - x_k)\, \Phi_k(\mathbf{s}_k) + x_k\, \Phi_{\text{edge}}(\mathbf{s}_k). \tag{1}$$

For a user with $x_k = 1$, it needs to download $\mathbf{v}_k$ from the server, where the image data volume (in bits) is V_k. This

This work is supported by the National Key R&D Program of China (No. 2025YFE0204100), the Science and Technology Development Fund of Macao S.A.R (FDCT) (No. 0074/2025/AMJ), the National Natural Science Foundation of China (Grant No. 62371444), and the Shenzhen Science and Technology Program (Grant No. RCYX20231211090206005, JCYJ20241202124934046).

Corresponding author: Shuai Wang (s.wang@siat.ac.cn).

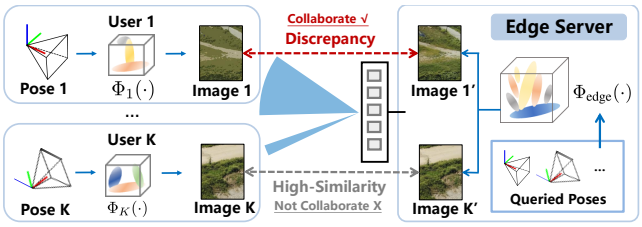


Fig. 1: The ECO-GS system.

procedure can be modeled as $r_k = \sum_{j=1}^{K} x_j \mathbf{h}_k^H \mathbf{w}_j z_j + n_k$, where $r_k \in \mathbb{C}$ is the received signal, $\mathbf{h}_k \in \mathbb{C}^{N\times 1}$ is the downlink channel from the server to the k-th user, $\mathbf{w}_k \in \mathbb{C}^{N\times 1}$ is the beamforming vector with power $\|\mathbf{w}_k\|^2 = p_k$, $z_k \in \mathbb{C}$ is the modulated signal with power $\mathbb{E}[|z_k|^2] = 1$, and $n_k \in \mathbb{C}$ is additive white Gaussian noise (AWGN) with zero mean and variance σ_k^2. To decode z_k, we employ the maximum ratio combining (MRC) beamformer $\mathbf{w}_k = \sqrt{p_k}\,\|\mathbf{h}_k\|_2^{-1}\mathbf{h}_k$, which is asymptotically optimal when $N \gg K$ [18]. The data rate of user k is given by $R_k(p_k) = B_k \log_2\left(1 + \|\mathbf{h}_k\|_2^2 p_k/\sigma_k^2\right)$, where B_k in Hz is the bandwidth of user k and we have adopted $|\mathbf{h}_k^H \mathbf{h}_j|^2/\|\mathbf{h}_k\|_2^2 \to 0$ for $k \neq j$ as $N \to +\infty$ [18, 19].

In ECO-GS systems, the design variables are the collaboration status vector $\mathbf{x} = [x_1, \cdots, x_K]^T$ and transmit power vector $\mathbf{p} = [p_1, \cdots, p_K]^T$. The edge transmit power should not exceed its budget P, i.e., $\sum_{k=1}^{K} p_k \leq P$ [13, 15]. To ensure smoothness, remote rendering should be completed within a time duration T, i.e., $T_0 + x_k V_k R_k^{-1} \leq T$ [2], where T_0 is edge rendering time and $x_k V_k R_k^{-1}$ is image downloading time. To avoid computation overload, the server supports at most S users, i.e., $\sum_{k=1}^{K} x_k \leq S$ [12]. Having the resource constraints satisfied, it is then crucial to minimize the discrepancy between the rendered image $\mathbf{v}_k$ and the ground truth image $\widehat{\mathbf{v}}_k$. To quantify the deviation from the desired image, we adopt a *rendering error function* based on [1], defined as:

$$\mathcal{L}\left(\mathbf{v}_k, \widehat{\mathbf{v}}_k\right) = (1-\lambda)\|\mathbf{v}_k - \widehat{\mathbf{v}}_k\|_1 + \lambda \mathcal{L}_{\text{DS}}\left(\mathbf{v}_k, \widehat{\mathbf{v}}_k\right), \quad (2)$$

where the structural dissimilarity metric is computed as $\mathcal{L}_{\text{DS}}\left(\mathbf{v}_k, \widehat{\mathbf{v}}_k\right) = 1 - \text{SSIM}\left(\mathbf{v}_k, \widehat{\mathbf{v}}_k\right)$, with SSIM being the structural similarity index measure function detailed in [20, Eqn. 5], and λ is a small weighting factor [1]. Combining the above leads to the **IRAC** problem below:

$$\text{P}: \min_{\mathbf{x},\mathbf{p}} \sum_{k=1}^{K} \mathcal{L}\left(\mathbf{v}_k, \widehat{\mathbf{v}}_k\right) \quad (3a)$$

$$\text{s.t.}\ B_k \log_2\left(1 + \frac{\|\mathbf{h}_k\|_2^2 p_k}{\sigma_k^2}\right) \geq \frac{x_k V_k}{T - T_0},\ \forall k, \quad (3b)$$

$$\sum_{k=1}^{K} p_k \leq P,\ \sum_{k=1}^{K} x_k \leq S, \quad (3c)$$

$$p_k \geq 0,\ x_k \in \{0,1\},\ \forall k. \quad (3d)$$

Solving problem P is challenging due to: (i) the inaccessibility of ground truth images, i.e., the cost function $\mathcal{L}\left(\mathbf{v}_k, \widehat{\mathbf{v}}_k\right)$ cannot be evaluated directly; (ii) P involves coupling between binary variables $\mathbf{x}$ and continuous variables $\mathbf{p}$.

3. PENALTY MAJORIZATION MINIMIZATION

To tackle challenge (i), we find a surrogate cost function of problem P. According to the triangle inequality, we have

$$\begin{aligned}&\|\widehat{\mathbf{v}}_k - \Phi_k(\mathbf{s}_k)\|_1 - \|\Phi_{\text{edge}}(\mathbf{s}_k) - \widehat{\mathbf{v}}_k\|_1 \\ &\leq \|\Phi_{\text{edge}}(\mathbf{s}_k) - \Phi_k(\mathbf{s}_k)\|_1. \end{aligned} \quad (4)$$

Putting (4) into (2), and with a sufficiently small λ, we have

$$\mathcal{L}\left(\widehat{\mathbf{v}}_k, \Phi_k(\mathbf{s}_k)\right) - \mathcal{L}\left(\Phi_{\text{edge}}(\mathbf{s}_k), \widehat{\mathbf{v}}_k\right) \leq \mathcal{L}\left(\Phi_{\text{edge}}(\mathbf{s}_k), \Phi_k(\mathbf{s}_k)\right).$$

Putting the above inequality and (1) into (2), we have

$$\begin{aligned}&\mathcal{L}\left(\mathbf{v}_k, \widehat{\mathbf{v}}_k\right) = (1-x_k)\mathcal{L}\left(\Phi_k(\mathbf{s}_k), \widehat{\mathbf{v}}_k\right) + x_k \mathcal{L}\left(\Phi_{\text{edge}}(\mathbf{s}_k), \widehat{\mathbf{v}}_k\right) \\ &\leq \underbrace{\mathcal{L}\left(\Phi_{\text{edge}}(\mathbf{s}_k), \widehat{\mathbf{v}}_k\right)}_{\text{const.}} + \underbrace{(1-x_k)\mathcal{L}\left(\Phi_{\text{edge}}(\mathbf{s}_k), \Phi_k(\mathbf{s}_k)\right)}_{\text{var.}}.\end{aligned}$$

As such, minimizing $\mathcal{L}$ in problem P can be safely converted to minimizing its upper bound, which is now an explicit function of x_k. Problem P thus becomes

$$\begin{aligned}\text{P1}: \min_{\mathbf{x},\mathbf{p}} &\sum_{k=1}^{K} (1-x_k)\mathcal{L}\left(\Phi_{\text{edge}}(\mathbf{s}_k), \Phi_k(\mathbf{s}_k)\right) + \text{const.} \\ \text{s.t.}\ &(3b),\ (3c),\ (3d),\end{aligned} \quad (5)$$

where term const. can be ignored. The new objective function in P1 is termed **GS switching** function, which quantifies the difference between collaborative and non-collaborative rendering. This function no longer involves ground truth images, and can be computed using GS models Φ_{edge} and $\{\Phi_k\}$. Consequently, we can deploy all GS models at the server to evaluate the GS switching cost.

To tackle challenge (ii), we relax the binary constraint $x_k \in \{0,1\}$ in (3d) into an affine constraint $x_k \in [0,1]$. To promote a binary solution for the relaxed variable $\{x_k\}$, we augment the objective function with a penalty term [21]: $\varphi(\mathbf{x}) = \frac{1}{\beta}\sum_{k=1}^{K} x_k(1-x_k)$, where $\beta > 0$ is the penalty parameter. Then problem P1 is transformed into

$$\begin{aligned}\text{P2}: \min_{\mathbf{x},\mathbf{p}} &\sum_{k=1}^{K} (1-x_k)\mathcal{L}\left(\Phi_{\text{edge}}(\mathbf{s}_k), \Phi_k(\mathbf{s}_k)\right) + \varphi(\mathbf{x}) \\ \text{s.t.}\ &p_k \geq 0,\ 0 \leq x_k \leq 1,\ \forall k,\ (3b),\ (3c).\end{aligned} \quad (6)$$

According to [21, Proposition 1], there always exists a β such that P1 and P2 are equivalent problems.

To further tackle the non-convexity of $\varphi(\mathbf{x})$, we propose to leverage the MM framework, which solves a sequence of surrogate problems $(\text{P2}[1], \text{P2}[2], \cdots)$ of P2 in an iterative fashion. Specifically, given solution $\mathbf{x}[n]$ at the n-th iteration, problem P2$[n+1]$ is considered at the $(n+1)$-th iteration:

$$\begin{aligned}\min_{\mathbf{x},\mathbf{p}} &\sum_{k=1}^{K} (1-x_k)\mathcal{L}\left(\Phi_{\text{edge}}(\mathbf{s}_k), \Phi_k(\mathbf{s}_k)\right) + \widehat{\varphi}(\mathbf{x}|\mathbf{x}^{[n]}) \\ \text{s.t.}\ &p_k \geq 0,\ 0 \leq x_k \leq 1,\ \forall k,\ (3b),\ (3c).\end{aligned} \quad (7)$$

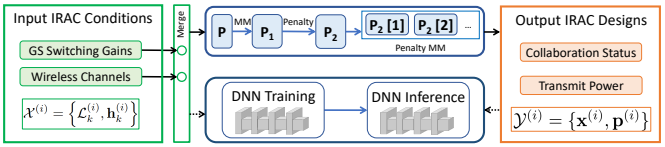


Fig. 2: Pipeline of the ILO-IRAC algorithm.

where the surrogate function is given as follows:

$$\widehat{\varphi}(\mathbf{x}|\mathbf{x}^{[n]}) = \sum_{k=1}^{K} \left(\frac{1}{\beta} x_k - \frac{2}{\beta} x_k^{[n]} x_k + \frac{1}{\beta} x_k^{[n]^2} \right). \quad (8)$$

Proposition 1 *The function $\widehat{\varphi}$ satisfies the following:*
(i) Convexity: $\widehat{\varphi}(\mathbf{x}|\mathbf{x}^{[n]})$ is convex in $\mathbf{x}$;
(ii) Upper bound: $\widehat{\varphi}(\mathbf{x}|\mathbf{x}^{[n]}) \geq \varphi(\mathbf{x})$;
(iii) Local equivalence:

$$\widehat{\varphi}(\mathbf{x}^{[n]}|\mathbf{x}^{[n]}) = \varphi(\mathbf{x}^{[n]}), \ \nabla_{\mathbf{x}} \widehat{\varphi}(\mathbf{x}^{[n]}|\mathbf{x}^{[n]}) = \nabla_{\mathbf{x}} \varphi(\mathbf{x}^{[n]}). \quad (9)$$

Part (i) is proved by checking the semi-definiteness of the Hessian of $\widehat{\varphi}$. Part (ii) is proved by checking $\widehat{\varphi}(\mathbf{x}|\mathbf{x}^{[n]}) - \varphi(\mathbf{x}) = \frac{1}{\beta} \sum_{t=1}^{T} (x_k - x_k^{[n]})^2 \geq 0$. Part (iii) is proved by comparing the function and gradient values of $\widehat{\varphi}$ and φ. Based on part (i) of **Proposition 1**, problem P2$[n+1]$ is convex and can be solved by off-the-shelf software packages (e.g., Mosek). Denote its optimal solution as $\mathbf{x}^*$. Then, we set $\mathbf{x}^{[n+1]} = \mathbf{x}^*$, such that the process repeats with solving the problem P2$[n+2]$. According to parts (ii)–(iii) of **Proposition 1** and [16, Theorem 1], every limit point of the sequence $(\mathbf{x}^{[0]}, \mathbf{p}^{[0]}), (\mathbf{x}^{[1]}, \mathbf{p}^{[1]}), \cdots$ is a *critical point* to problem P2 as long as the starting point $(\mathbf{x}^{[0]}, \mathbf{p}^{[0]})$ is feasible to problem P2. The complexity of solving problem P2 (thus P1) with PMM is $\mathcal{O}(\mathcal{M}(2K)^{3.5})$, where $\mathcal{M}$ is the number of iterations for PMM to converge.

4. IMITATION LEARNING OPTIMIZATION

In Section 3, we have developed a PMM method to solve the problem. However, the complexity of the PMM algorithm is high when K is large, resulting in additional latency in problem solving.

To address this issue, we propose an ILO method for faster IRAC design. The ILO trains an agent to solve P1 by imitating the PMM algorithm, such that time-consuming iterative computations are converted into real-time feed-forward inference [22,23]. The system architecture of ILO is shown in Fig. 2, which consists of three phases.

1) Offline Demonstration Data Generation: We adopt the PMM algorithm to generate the demonstration dataset $\mathcal{D} = \{\mathcal{S}^{(i)}\}_{i=1}^{I}$, where the i-th sample is given by $\mathcal{S}_i = \{\mathcal{X}^{(i)}, \mathcal{Y}^{(i)}\}$, and the input tuple is given by

$$\mathcal{X}^{(i)} = \left\{ \underbrace{\mathcal{L}(\Phi_{\text{edge}}(\mathbf{s}_k^{(i)}), \Phi_k(\mathbf{s}_k^{(i)}))}_{:=\mathcal{L}_k^{(i)}}, \mathbf{h}_k^{(i)} \right\},$$

Table 1: Large GS Versus Small GS

Model	Platform	↓Dim.	↓Model Size	↑PSNR	↑SSIM	↓L1	↓Loss
Φ_{edge}	Edge	64	509.8 M	27.49	0.98	0.031	0.029
Φ_k	Device	14	9.3 M	24.99	0.97	0.043	0.041

with $\mathcal{L}_k^{(i)}$ generated by GS model comparisons at sampled poses $\{\mathbf{s}_k^{(i)}\}$. Note that we can also include other factors $\{B_k, V_k, \sigma_k^2, T, P, S\}$ into $\mathcal{X}^{(i)}$. The output tuple $\mathcal{Y}^{(i)} = \{\mathbf{x}^{(i)}, \mathbf{p}^{(i)}\}$ is generated by executing the PMM algorithm on problem configuration $\mathcal{X}^{(i)}$.

2) Offline DNN Training: The DNN consists of three fully-connected layers with 100, 72, and 20 units, respectively. The first two hidden layers utilize the rectified linear unit (ReLU) activation function. For model training, we use the Focal Loss function and optimize the network parameters using the AdamW algorithm.

3) Online DNN Inference: The pre-trained DNN model is deployed at the edge server to make real-time decisions. At each time step t, the server estimates the current rendering and communication conditions $\mathcal{X}^{(t)}$ using users' poses and pilots. Feeding $\mathcal{X}^{(t)}$ into the DNN model, we can obtain the optimal collaboration status $\mathbf{x}^{(t)}$ and transmit power $\{\mathbf{p}^{(t)}\}$.

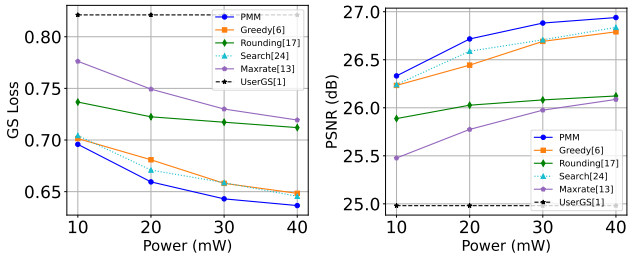
5. EXPERIMENTS

We implement the proposed ECO-GS system with PMM and ILO algorithms in Python based on the 3D GS project [1]. We deploy the system on a Linux workstation with AMD 3.7 GHz CPU and NVIDIA A6000 GPU. We conduct experiments exploiting the truck dataset [1].

We consider the case of $N = 600$ and $(K, S) = (20, 10)$ [19]. Edge-user distances $\{d_k\}$ are computed based on their locations, where users are randomly located within a $100 \times 100\,\text{m}^2$ area and edge server is at $(0, 0)$. Channels are generated by $\mathbf{h}_k \sim \mathcal{CN}(0, \varrho_k \mathbf{I}_N)$, where $\mathcal{CN}$ is complex Normal distribution, $\mathbf{I}_N$ is $N \times N$ identity matrix, and pathloss $\varrho_k = d_k^{-\alpha}$ with $\alpha = 3$ [19]. The noise power $\sigma^2 = -70\,\text{dBm}$ and the bandwidth is $B_k = 2\,\text{MHz}$. The end-to-end time budget is $T = 60\,\text{ms}$, the edge rendering time is $T_0 = 6.5\,\text{ms}$ (tested $\Phi_{\text{edge}}(\cdot)$ on A6000 server), and the user rendering time is 16.7 ms (tested Φ_k on a desktop computer). All quantitative results are obtained by averaging over 100 random simulation runs, with independent channels and user views in each run.

We compare ECO-GS with PMM and ILO with the following baselines: 1) **UserGS**: GS [1] (but with model compression) (i.e., $\{x_k = 0\}$); 2) **MaxRate**: ECO-GS with sum-rate maximization [13]; 3) **Greedy**: ECO-GS with large-$\mathcal{L}_k$-first principle [6]; 4) **Search**: ECO-GS by solving P_{GS} with iterative local search [24]; 5) **Rounding**: ECO-GS by solving P_{GS} with continuous relaxation and rounding [17];

Table I compares small and large GS models. The small GS model is obtained by reducing the dimension of each Gaussian from 64 to 14. It reduces the model size by **over** 50**x** compared to the large GS model. This demonstrates the effectiveness of employing model compression for low-cost



(a) GS Loss versus P. (b) PSNR versus P.

Fig. 3: Quantitative comparison of different schemes.

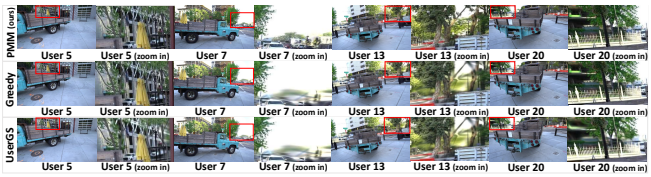


Fig. 4: Comparison of rendered images.

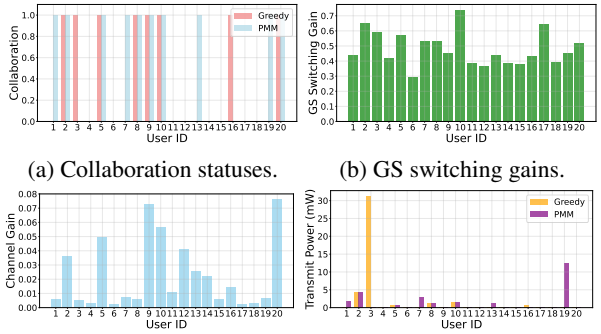
GS deployment. However, the PSNR and SSIM performances are degraded, meaning that remote rendering is necessary if small GS models fail in rendering qualified images.

Next, we compare the proposed IRAC with PMM to other benchmark schemes under $P = \{10, 20, 30, 40\}$ mW. It can be seen from Fig. 3 that the proposed IRAC with PMM consistently outperforms all the other schemes. Compared with UserGS, the loss reduction is at least 15%, and the improvement of PSNR is over 1.3 dB.

To obtain deeper insights, we consider the case of $P_{\text{sum}} = 40$ mW, and the rendering results are shown in Fig. 4. It can be seen that all the images of PMM have high qualities. This corroborates Fig. 5a and Fig. 5b, where the selected users have high GS switching gains. This implies that PMM is able to distinguish heterogeneous rendering requirements. We also observe that images $(7, 13)$ are blurred for the Greedy scheme. This is because the Greedy scheme selects user 3 with higher GS loss for collaboration. However, this user experiencing bad channel conditions costs excessive power resources. In contrast, our scheme avoids collaborating remote users as shown in Fig. 5c and Fig. 5d. Lastly, all images are blurred for the UserGS scheme, which demonstrates the necessity of leveraging edge GS to compliment local GS.

One may wonder whether we can adopt remote rendering for all users, i.e., $\{x_k = 1\}$. This is known as the **EdgeGS** scheme [4], which may lead to non-smooth rendering due to latency. To see this, the end-to-end latency at $P_{\text{sum}} = 10$ mW is provided in Fig. 6. It can be seen that EdgeGS leads to over 80 ms latency, while our method strictly guarantees less than 60 ms for smooth rendering.

Finally, we evaluate the proposed ILO-IRAC. We generate 10000 random samples of $\{\mathcal{X}^{(i)}, \mathcal{Y}^{(i)}\}$, including 5000 training and 5000 test samples. We train the ILO DNN for 200 epochs with a learning rate of 6×10^{-4} and batch size of 96. Fig. 7a and Fig. 7b illustrate the training and testing losses and accuracies, respectively. It can be seen that



(a) Collaboration statuses. (b) GS switching gains.

(c) Channel gains. (d) Transmit powers.

Fig. 5: Case study for PMM and Greedy schemes.

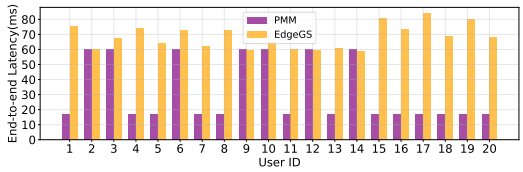
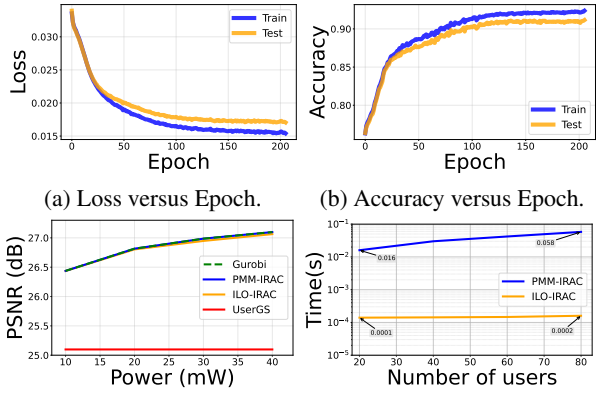


Fig. 6: Case study for PMM and EdgeGS schemes.



(a) Loss versus Epoch. (b) Accuracy versus Epoch.

(c) PSNR versus P. (d) Execution Time versus K.

Fig. 7: Evaluation of ILO-IRAC.

the proposed ILO-IRAC achieves an accuracy of 0.9 and a loss of 0.0175, indicating that ILO can learn how to imitate PMM. The PSNR performances and computation times of ILO-IRAC and PMM-IRAC are compared in Fig. 7c and Fig. 7d. ILO reduces the time by 100x compared to that of PMM, while achieving a competitive PSNR with only 0.02 dB degradation. The computation time of ILO-IRAC is within 1 ms, making it suitable for real-time execution.

6. CONCLUSION

This paper has presented the IRAC framework for ECO-GS, which mitigates the discrepancies between GS-rendered images and real-world environments under communication constraints. We have proposed two algorithms to obtain the best trade-off between timeliness and fidelity in IRAC designs. Extensive experiments have demonstrated that collaborative rendering indeed helps, but we need to distinguish non-beneficial users. Furthermore, remote users should be carefully treated, as they may cost excessive transmit powers.

7. REFERENCES

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, Jul. 2023.

[2] Z. Liu et al., "Voyager: Real-time splatting city-scale 3d Gaussians on your phone," *arXiv preprint arXiv:2506.02774*, 2025.

[3] Y. Gao et al., "Cosurfgs: Collaborative 3D surface Gaussian splatting with distributed learning for large scene reconstruction," *arXiv preprint arXiv:2412.17612*, 2024.

[4] A. Mehrabi, M. Siekkinen, T. Kämäräinen, and A. Ylä-Jääski, "Multi-tier CloudVR: Leveraging edge computing in remote rendered virtual reality," *ACM Trans. Multimedia*, vol. 17, no. 2, pp. 1–24, May 2021.

[5] L. Li, Y. Huang, X. Qiao, Y. Meng, D. Yu, P. Ren, and S. Dustdar, "Toward distributed collaborative rendering service for immersive mobile web," *IEEE Netw.*, vol. 38, no. 3, pp. 137–145, May-Jun 2023.

[6] C. Liu et al., "Communication efficient robotic mixed reality with Gaussian splatting cross-layer optimization," *IEEE Trans. Cogn. Commun. Networking*, vol. 12, pp. 1948–1962, 2026.

[7] G. Li, X. Jin, Y. Wan, C. Liu, T. Zhang, S. Wang, and C. Xu, "Embodied edge intelligence meets near field communication: Concept, design, and verification," *IEEE Netw.*, vol. 39, no. 6, pp. 78–86, 2026.

[8] S. Qiu, B. Xie, Q. Liu, and P.-A. Heng, "Advancing extended reality with 3D Gaussian splatting: Innovations and prospects," in *Proc. IEEE AIxVR*, 2025, pp. 1–6.

[9] K. Luo, K. Zhao, S. Ye, T. Ouyang, and X. Chen, " MIX3D: A mixed representation for communication-efficient distributed 3DGS training," in *Proc. MSN*, 2024, pp. 382–389.

[10] A. D. Jagtap, R. Song, S. T. Sadashivaiah, and A. Festag, "V2X-Gaussians: Gaussian splatting for multi-agent cooperative dynamic scene reconstruction," in *Proc. IV*, 2025, pp. 1033–1039.

[11] S. Wang, Y.-C. Wu, M. Xia, R. Wang, and H. V. Poor, "Machine intelligence at the edge with learning centric power allocation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7293–7308, Nov. 2020.

[12] G. Li, R. Han, S. Wang, F. Gao, Y. C. Eldar, and C. Xu, "Edge accelerated robot navigation with collaborative motion planning," *IEEE/ASME Trans. Mechatron.*, vol. 30, no. 2, pp. 1166–1178, Apr. 2025.

[13] Q. Zhang, M. Shao, T. Zhang, G. Chen, J. Liu, and P. Ching, "An efficient sum-rate maximization algorithm for fluid antenna-assisted ISAC system," *IEEE Commun. Lett.*, vol. 29, no. 1, pp. 200–204, Jan. 2025.

[14] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2909–2913, Jan. 2016.

[15] W. Yu, W. Rhee, S. Boyd, and J. M. Cioffi, "Iterative water-filling for Gaussian vector multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 50, no. 1, pp. 145–152, Jan. 2004.

[16] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017.

[17] R. Hübner and A. Schöbel, "When is rounding allowed in integer nonlinear optimization?" *Eur. J. Oper. Res.*, vol. 237, no. 2, pp. 404–410, Sep. 2014.

[18] Y. Song, Z. Gong, Y. Chen, and C. Li, "Distributed massive MIMO with low resolution ADCs for massive random access," *IEEE J. Sel. Top. Sign. Proces.*, vol. 18, no. 7, pp. 1381–1395, Oct. 2024.

[19] S. Wang, M. Wen, M. Xia, R. Wang, Q. Hao, and Y.-C. Wu, "Angle aware user cooperation for secure massive mimo in rician fading channel," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 9, pp. 2182–2196, Sep. 2020.

[20] S. Wang, A. Rehman, Z. Wang, S. Ma, and W. Gao, "SSIM-motivated rate-distortion optimization for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 516–529, Apr. 2012.

[21] S. Lucidi and F. Rinaldi, "Exact penalty functions for nonlinear integer programming problems," *J. Optim. Theory Appl.*, vol. 145, pp. 479–488, Apr. 2010.

[22] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *Proc. IEEE*, vol. 111, no. 5, pp. 465–499, May 2023.

[23] Y.-F. Liu et al., "A survey of recent advances in optimization methods for wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 11, pp. 2992–3031, Nov. 2024.

[24] F. Neumann and I. Wegener, "Randomized local search, evolutionary algorithms, and the minimum spanning tree problem," *Theor. Comput. Sci.*, vol. 378, no. 1, pp. 32–40, Jun. 2007.