2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Multi-View Control for Robust 3D Gaussian Splatting

YuNong Mao, ZhiBin Zhang*, YuFu Shi

Abstract— **3D Gaussian Splatting (3DGS) has recently demonstrated impressive capabilities in real-time novel view synthesis. However, the performance of 3DGS tends to degrade significantly when the quality of the initial point cloud is poor. Specifically, the lack of an effective pruning strategy to thoroughly eliminate suboptimal points (defined as erroneous points in this paper). The excessive accumulation of these erroneous points leads to overfitting in specific viewpoints, thereby affecting the visual appearance and geometric accuracy in novel view synthesis. To address these challenges, we propose a novel 3DGS optimization method named MVC-GS, which introduces two key innovative contributions. First, based on multi-view geometric constraints, we use image rendering errors as a guiding criterion for optimization. By performing point calibration in the target region, we effectively mitigate the impact of erroneous Gaussian points. Subsequently, we introduce a multi-view Gaussian attribute optimization method that further enhances the precision of 3D Gaussian attributes representation, while avoiding overfitting to the training views. We conducted comprehensive visualization analysis across multiple scenes in various datasets. Extensive experiments on public datasets show that the proposed method achieves state-of-the-art performance across diverse scenes.**

I. INTRODUCTION

New View Synthesis plays a critical role in applications such as visualization, simulation, automation, and VR/AR. The advent of Neural Radiation Fields (NeRF)[11] has significantly improved the quality of view synthesis by bypassing the need for explicit reconstruction of geometry, textures, materials, and lighting, which are often challenging and uncertain inverse problems. In recent years, 3D Gaussian Splatting (3DGS)[1] has garnered substantial attention due to its exceptionally fast rendering speed and compositing quality that rivals, or even surpasses, NeRF. The fundamental premise of 3DGS is the utilization of a set of Gaussian ellipsoids to simulate a given scene. The efficiency of the rendering process is achieved through the rasterisation of the aforementioned Gaussian ellipsoids into a visual representation. This representation allows for efficient blending and interpolation, resulting in high-quality visuals.

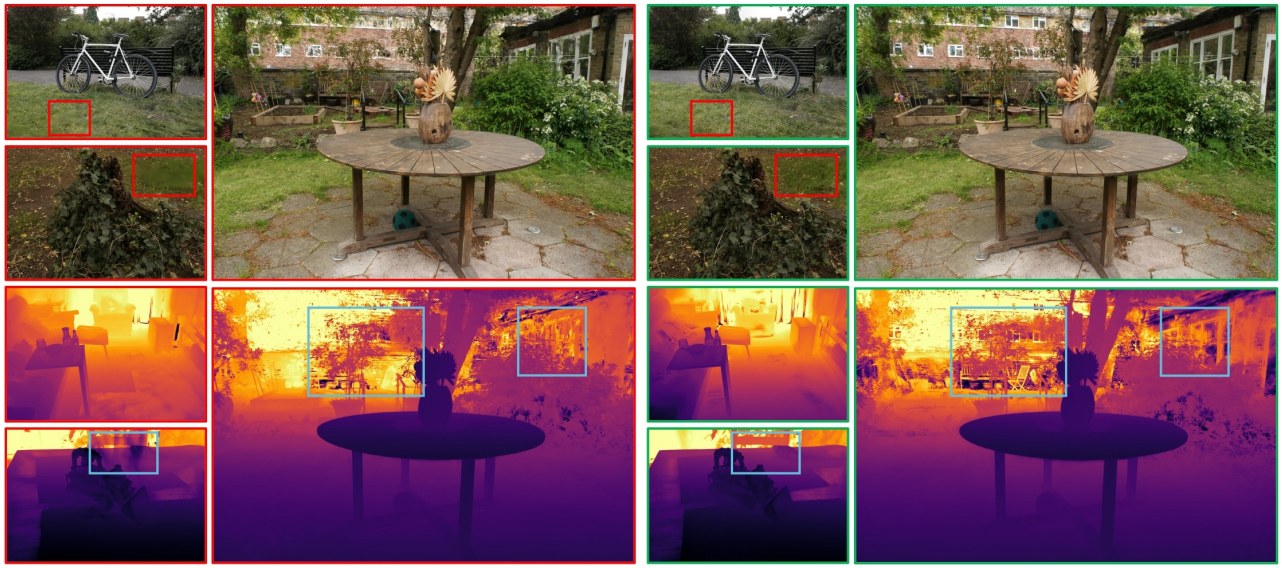
The performance of 3DGS for scene reconstruction heavily depends on the quality of the initial point cloud. When starting training with a randomly initialized point cloud, the performance of 3DGS drops significantly, resulting in blurred generated maps. Similar performance degradation may also occur in real-world scenarios, especially in scenarios where Structure-of-Motion (SfM)[2] techniques are difficult to converge. Jung et al. [5] proposed a method based on the sparse initialization of the SfM point cloud with a Gaussian distribution assigned a large variance. By gradually applying low-pass filtering, they prevented the generation of 2D Gaussian projections smaller than the pixel size. They successfully demonstrated that even with a random point cloud initialization, 3DGS could still converge effectively. However, this convergence does not always imply high-precision reconstruction results.

Specifically, the presence of erroneous Gaussian points has a detrimental impact on the outcomes. These erroneous points typically arise from inaccuracies during the model training process, and due to the lack of an effective pruning strategy, they accumulate in subsequent steps, preventing precise localization of the Gaussian distribution points. As the accumulation of erroneous points increases, certain views may become distorted, thus affecting the quality of new view generation. As shown in Fig 1, the region outlined by the rectangular box in part (a) exhibits significant image distortion when compared to the corresponding region in part (b), where our method is applied. The results indicate that, although previous methods have shown improvements in certain scenarios, they still fail to completely eliminate these erroneous Gaussian points. Therefore, removing these erroneous points and ensuring the precise localization of the Gaussian distribution points is a critical issue for improving the performance of 3DGS.

To overcome the limitations mentioned earlier, in this paper, we propose two innovations. First, we introduce a novel dual-view point correction method aimed at identifying the 3D Gaussian points responsible for rendering errors. Starting with an image rendering error map of a specific view, we leverage feature mapping in conjunction with multi-view geometric constraints to utilize region correspondences between different views. For each pair of corresponding regions, we cast rays through their respective camera views and identify error source region. Simultaneously, we reset points positioned in front of these regions that exhibit high opacity, as they may substantially impact the rendering. To minimize model expansion, we prune points based on opacity while considering density. The second innovation lies in altering the traditional training paradigm, which typically uses single-view supervision per iteration. Based on this, we propose a multi-view constraint method, where multiple views are introduced per iteration during training. This method forces the 3D Gaussian points to jointly learn the structure and appearance of multiple views, effectively

School of Computer Science, Inner Mongolia University, China.
*Corresponding author: ZhiBin Zhang (cszhibin@imu.edu.cn)

This work was supported by the Inner Mongolia Natural Science Foundation project (No.2024MS06019), the Major Special Project of the Inner Mongolia Autonomous Region (No.2021ZD0043), the Inner Mongolia University independent project (No.21500-5247012), and the fund of supporting the reform and development of local universities (Disciplinary construction).



(a) RAIN-GS (Random) **(b) Ours (Random)**

Fig. 1. **Visual comparison between RAIN-GS and our proposed method.** The figure show the results of RAIN-GS and our trained with randomly initialized point cloud.

avoiding overfitting to a specific view. This optimization ensures that the 3DGS kernels are constrained to meet the rendering requirements for multiple views, rather than over-relying on any single view. Comprehensive visual analysis on public datasets demonstrates that our proposed method achieves state-of-the-art performance across diverse tasks and scenarios. The main contributions of this work can be summarized as follows:

- We propose a Dual-view Point Correction (DPC) mechanism that uses multi-view geometry to refine the distribution of Gaussian points, thereby enhancing spatial accuracy and minimizing noise.
- We propose Multi-View Constraints (MVC) for capturing high-frequency details, efficiently combining information from multiple viewpoints to enhance the reconstruction of sharp edges and fine details.
- Comprehensive evaluations on several public datasets reveal that our model achieves state-of-the-art performance, affirming the efficacy of the proposed approach, while also show-casing its robust versatility across a wide range of scenarios.

II. Related work

A. Novel view synthesis(NVS)

NVS involves generating new images from viewpoints that are distinct from the original captures. Recently, NeRF[11]has garnered significant attention due to its impressive performance in NVS. Further research has broadened the usefulness of NeRF to various applications, including mesh reconstruction[18]-[20], inverse rendering[26]-[29], autonomous driving[21], [22], [36] and video generation[23]-[25]. Unlike NeRF-style scene representation, 3DGS[1] models a scene as a set of anisotropic Gaussian primitives in 3D space, typically initialized using Structure-from-Motion (SfM). This method distributes Gaussians across the scene to approximate geometry and radiance, allowing for faster training and rendering, particularly in large-scale or complex environments. The efficiency and scalability of 3DGS have led to its rapid adoption in diverse research areas[4],[30]-[33]. Beyond static scenes, it has been successfully adapted to dynamic scene reconstruction[16], [17], [34] and explored as a fast alternative for text-driven 3D content generation[14], [15], [35] By transitioning from NeRF's dense neural network representation to the more explicit and sparse Gaussian primitive representation of 3DGS, the latter achieves a balance between quality and efficiency, making it a promising direction for future novel view synthesis applications.

B. Structure-from-Motion (SfM)

SfM[2] techniques have become one of the most widely used algorithms for 3D scene reconstruction. Given a set of input images, SfM typically estimates the corresponding camera poses and reconstructs a sparse set of 3D points with approximate geometric locations and color attributes. Relatively speaking, 3DGS relies on an accurate initial point cloud to initialize the position and color of the 3D Gaussians. However, performance significantly deteriorates when the initial point cloud becomes noisy. To address the initialization sensitivity issue in 3DGS, RAIN-GS[5] proposes a strategy that initializes sparse Gaussians with large variance from SfM point clouds and progressively applies low-pass filtering to avoid 2D Gaussian projections error that is smaller than a pixel. 3DGS-MCMC[3] samples a fixed number of Gaussians from the learned probability distribution, aiming to eliminate initialization dependency. However, these methods cannot further enhance scene reconstruction quality beyond addressing the sensitivity of 3DGS initialization. The proposed method fundamentally solves

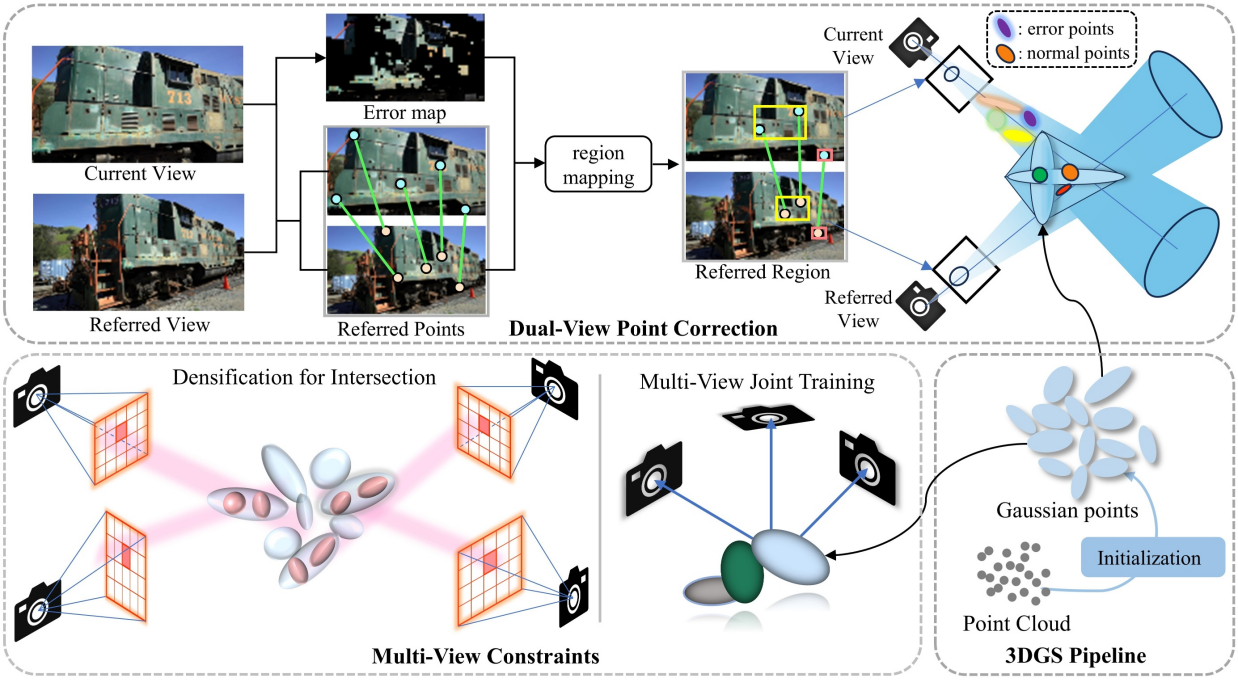


Fig. 2. **The overall pipeline of the proposed MVC-GS:** Building upon the original 3DGS framework,we propose a dual-view point correction method to identify and correct erroneous Gaussian points. Additionally, multi-view constraints are applied, incorporating two key steps: multi-view joint training and adaptive densification.

this issue, as shown in Tab. I. Even with randomly initialized point clouds, our method still outperforms the 3DGS method, which relies on precise initialization of point clouds.

III. Method

A. Preliminaries: 3D Gaussian Splatting

Given the camera extrinsics E and intrinsics K, the view-dependent radiance $\mathbf{C}(\cdot)$ of each pixel p is computed by blending a set of 3D Gaussians along the ray $r(p, E, K)$. 3DGS[1] achieves precise blending by rasterizing with N parameterized kernels $G(r) = \{g_i \mid i = 1, \ldots, N\}$ along the ray $r(p, E, K)$. Assuming that the color $c_i \in \mathbb{R}^3$, the opacity $o_i \in \mathbb{R}$, the covariance $\Sigma_i \in \mathbb{R}^{3\times3}$ represents the attribute of the i-th Gaussian g_i, the rendered pixel radiance $C(r)$ is then expressed as:

$$\mathbf{C}(r) = \mathbf{C}\left(\{g_i \mid i = 1, \ldots, N\}\right) = \sum_{i=1}^{N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{1}$$

where the color c_i is weighted by the transmittance $\alpha_i = o_i exp(-\frac{1}{2}(x_i)^T \Sigma_i^{-1}(x_i)$. Here x_i denotes the distance between the position of Gaussian kernel and the query pixel p. N represents the number of 3D Gaussians.

B. Dual-view Point Correction

Our method introduces multi-view geometric constraints to accurately identify erroneous 3D points that contribute to rendered inaccuracies. As show in Fig. 2(b), the initial step is to generate the image rendering error map for the current view. For each pair of matched regions, the rays are cast from their respective camera views, and the intersection of these rays is identified as a potential area of error, where corrections are subsequently implemented.

1) Error Region Localization.: To implement the 3D error localization, we render the current view image through the steps of the original 3DGS. The error function[12] is then used to generate an error map of the current view against another view. After generating the error map, it needs to be mapped back into 3D space. It corresponds to the region mapping step in Fig. 2(b). We follow the procedure of LightGlue[13], predicting the partial matching relationships between the two sets of local features extracted from the two view images, A and B, because LightGlue has already demonstrated its efficiency and accuracy in feature matching.

Each local feature consists of 2D point locations p normalised by the image size. Images A and B have M and N local features, indexed by $A = \{1, \ldots, M\}$ and $B = \{1, \ldots, N\}$, respectively. LightGlue output a set of correspondence $M = \{(i, j)\} \subseteq A \times B$. In the event of occlusion or an inability to match specific points, the region of the 2D rendering error in the current view may not necessarily correspond to the reference image. Therefore, the paired region (P, P') should be selected by matching the points. Furthermore, this pairing region is adaptively adjusted throughout the training process, particularly during Gaussian densification.

After determining the paired regions of rendering error, the 2D error regions are projected into 3D space using multi-view geometric constraints. Specifically, we project a cone of rays, denoted as x, from the camera's center of projection o along a direction L that aligns with the center of the pixel in the corresponding region P. The radius of this cone lies in

the image plane and the vertex is located in the center of the camera o. We set the radius to the smallest circumferential radius of the corresponding error region P in the 2D image to more accurately locate the Gaussian points that cause the 2D error region. Concurrently, the same operation is executed in the P' region to generate another cone. To achieve 3D error region recognition, we directly use the smallest sphere containing these error points as error 3D regions P_{error}.

2) Point Correction.: In 3DGS, gaussian points only relies on the view-averaged gradient magnitude to determine point densification globally. In addition to this, we further perform localized points addition and geometry calibration within the identified error source 3D zone P_{error}. For the point addition, we consider two common situations: (1) when points are already present, we apply a lower threshold to identify regions that need further densification, enhancing fine geometric details. This is similar to the original 3DGS approach but prioritizes specific 3D regions that require more detail. For smaller Gaussians in areas of low variance, we replicate them while maintaining their size and move them along the position gradient to capture emerging geometric structures. In regions with high variance, the larger Gaussians are subdivided into smaller ones to more accurately capture the geometry information. (2) in cases of point sparsity, we add new Gaussian points at the center of the 3D zone.

In the context of α-blending in 3DGS, points located at the forefront of the identified 3D region with higher opacity may occlude valid points, leading to image distortion, as shown in Fig. 1. The impact of erroneous points is more pronounced when using random initialization, as the color and position of the points are randomly distributed, increasing the likelihood of occlusion. To address this issue, we treat such points as potentially erroneous and reset (i.e., prune) them. To minimize model complexity, we adaptively prune points based on their opacity values, from low to high opacity. The number of points to be pruned is determined by the point density within the region.

C. Multi-Views Constraints

1) Multi-Views Joint Training.: The training strategy for 3DGS[1] follows NeRF[11] and optimizes the model parameters through single view supervision at each iteration. 3DGS is typically optimized under single-view supervision. The loss function can accordingly be expressed as:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_1(I, \mathbf{C}(r)) + \lambda(1 - \mathcal{L}_{\text{D-SSIM}}(I, \mathbf{C}(r))) \quad (2)$$

where $\mathcal{L}_1$ and $\mathcal{L}_{\text{D-SSIM}}$ denote the mean absolute error and D-SSIM loss, respectively. The $\mathbf{C}(r)$ same as Eq. 1.

Under single-view supervision, the loss function L is predominantly influenced by the gradients computed from a single view. This results in optimization biased towards the geometry and appearance visible from that view, while regions poorly represented in this perspective are often neglected. The reliance on single-view gradients creates an imbalance, where high-gradient regions dominate the training, ignoring areas with low gradients, such as occluded or under-represented details. This leads to suboptimal 3D reconstructions. In particular, the effect of randomly initialized Gaussian points can amplify this error, as their random positioning and appearance further exacerbate the imbalance in gradient distribution. To address these issues, our multi-view joint training method balances the gradient contributions from different views by combining gradients from various perspectives. This approach can be expressed as follows:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{G}} = \frac{\partial \mathcal{L}_1}{\partial G_1} + \frac{\partial \mathcal{L}_2}{\partial G_2} + \cdots + \frac{\partial \mathcal{L}_{\text{n}}}{\partial G_{\text{n}}} \quad (3)$$

Here, $\mathbf{G} = \{G_1, G_2, \ldots, G_n\}$ signifies that during multi-view training, a subset of 3D Gaussians $\mathbf{G}$ is influenced by significant gradients from each view. $\mathcal{L}_{\text{n}}$ represents the loss for the n-th view. By incorporating multi-view information, the optimization process for each Gaussian kernel g_i (see the form Eq. 1) is refined, effectively mitigating the impact of large gradients and alleviating potential overfitting issues that may arise in one or more specific views of them.

2) Adaptive Densification: Due to the nature of volume rendering and the explicit representation of 3DGS, 3D Gaussians in some regions have a significant impact on distinct views when rendering. For example, the central 3D Gaussian is crucial when rendering scenes with cameras being positioned around them in various poses. However, identifying these regions is challenging, especially in 3D space. As shown in Fig. 2(c), we propose a cross-ray densification strategy that begins in 2D space and adapts to a 3D search. Initially, the average loss for each window (i,j) is computed, where a window represents a region within the view. Subsequently, a sliding window of dimensions (h, w) is employed to identify the region exhibiting the highest average loss value.

The rays are then projected from the vertices of these regions, using four rays per window per viewpoint. The intersection points of these rays from different perspectives delineate cuboidal regions where significant 3D Gaussians are concentrated, which is essential for accurate rendering across multiple views. Guided by a loss metric that prioritizes regions requiring improvement for each view, these areas are accurately localized through light projection techniques. The 3D regions containing key Gaussians play a crucial role in the joint optimization of rendering across multiple views. Consequently, we increased the density of 3D Gaussians in the overlapping regions to enhance the effectiveness of multi-view training.

IV. Experiments

A. Experimental Setup

Datasets. Experiments are performed on three widely used datasets: Mip-NeRF 360[8], Tanks&Temples[10], and Deep Blending[9]. For evaluation, PSNR, LPIPS, and SSIM are computed using the split protocol of 3DGS, where 1 image in every group of 8 is reserved for testing and the other seven images are used for training. We keep the image resolution identical to that in 3DGS throughout all experiments.

Fig. 3. **Qualitative results on Mip-NeRF360, Tanks&Temples.** Our method generalizes well across different scenes. The results highlight and adaptability of our method, achieving consistent improvements in rendering quality compared to baseline, even in varied environments.

Implementation Details. All experiments were conducted using an A100 GPU. In addition to the original Gaussian densification strategies used in 3DGS, we also performed dual-views point correction, including additions, resets, and pruning. We maintained the same thresholds for splitting and cloning points as in the original 3DGS[1]. Furthermore, the model is trained with 30,000 iterations across all scenes, following the identical training schedule and hyperparameters as those employed by 3DGS. We compare our model with Plenoxels[6], InstantNGP-Base[7], InstantNGP-Big[7], 3DGS-MCMC[3], RAIN-GS[5] and 3DGS[1]. Two different types of point clouds were used to train 3DGS and our method: noise-free precisely initialized SfM point clouds and randomly initialized point clouds. 3DGS-MCMC and RAIN-GS were trained using randomly initialised point clouds.

B. Experimental results

Quantitative Comparison. The quantitative evaluation results are presented in Tab. I, our method consistently outperforms existing approaches across multiple datasets, fully validating the effectiveness and superiority of the proposed modules. Through comparative experiments, it is evident that our method demonstrates significant advantages in performance across different types of datasets, proving its broad applicability and robust capability in diverse scenarios. Notably, in all dataset tests, our random point cloud initialization strategy consistently outperforms the traditional SfM-based point cloud trained 3DGS method, highlighting the effectiveness of our initialization strategy. It is worth noting that, by comparing the results of training with SfM initialization and random initialization, we find that the final reconstruction results show negligible differences. This indicates that our model does not rely on high-quality initial point clouds, effectively avoiding the dependency on initialization quality seen in traditional methods, and completely eliminating the limitations imposed by initialization.

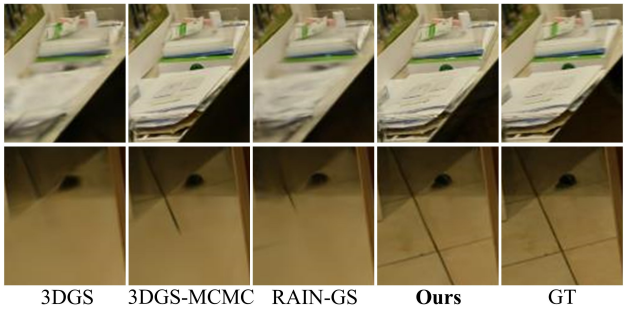


Fig. 4. **Visual Comparison of Scene Textures.**

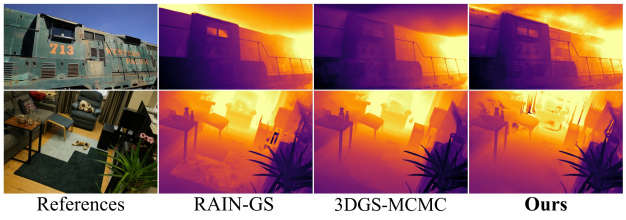


Fig. 5. **Visual Comparison of Scene Depth.**

Qualitative Comparison. In this section, we present a qualitative comparison of the proposed method with 3DGS (a), RAIN-GS (b), and 3DGS-MCMC (c) on new test views, as shown in Fig. 3. All of these methods use randomly initialized point clouds. By comparing (b), (c), and (d), we successfully validate the effectiveness of the proposed

TABLE I

EVALUATION OF NOVEL VIEW SYNTHESIS ON MIP-NERF360, TANKS&TEMPLES AND DEEP BLENDING. OUR METHOD OUTPERFORMS ALL BASELINES EVEN WHEN STARTING FROM RANDOM INITIALIZATION, WITH A LARGE GAP IN PERFORMANCE WHEN COMPARED WITH OTHER METHODS. WE HIGHLIGHT THE **BEST** AND SECOND-BEST FOR EACH COLUMN.

Method	Init Points	Mip-NeRF360			Tanks&Temples			Deep Blending		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Plenoxels[6]	✗	23.08	0.626	0.463	21.08	0.719	0.379	23.06	0.795	0.510
INGP-Base[7]	✗	25.30	0.671	0.371	21.72	0.723	0.330	23.62	0.797	0.423
INGP-Big[7]	✗	25.59	0.699	0.331	21.92	0.745	0.305	24.96	0.817	0.390
3DGS[1]	Random	22.19	0.704	0.313	20.99	0.765	0.237	28.51	0.895	0.258
RAIN-GS[5]	Random	27.23	0.807	0.229	23.13	0.826	0.207	29.42	0.899	0.255
3DGS-MCMC[3]	Random	27.51	0.814	0.186	23.50	0.845	0.175	29.31	0.905	0.252
Ours	Random	28.42	0.832	0.168	**24.07**	**0.863**	**0.147**	30.02	0.917	0.236
3DGS[1]	SfM	27.21	0.815	0.214	23.14	0.841	0.183	29.41	0.903	0.243
Ours	SfM	**28.55**	**0.837**	**0.162**	24.01	0.862	0.149	**30.10**	**0.923**	**0.234**

TABLE II

ABLATION ON CORE COMPONENTS ON THE MIP-NERF360 DATASET. WE CONDUCTED TESTS USING BOTH RANDOMLY INITIALIZED POINT CLOUDS AND SFM INITIALIZED POINT CLOUDS.

Method	DPC	MVC	Mip-NeRF360		
			PSNR↑	SSIM↑	LPIPS↓
Random	✗	✗	27.23	0.807	0.229
	✓	✗	27.65	0.818	0.209
	✗	✓	27.83	0.815	0.197
	✓	✓	28.41	0.832	0.168
SfM	✗	✗	27.31	0.812	0.213
	✓	✓	**28.55**	**0.837**	**0.162**

TABLE III

PERFORMANCE COMPARISON FOR DIFFERENT CONFIGURATIONS. WE TESTED THE IMPACT OF ADD AND RESET POINTS ON THE RESULTS.

Methods	Train			Truck		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Full DPC	21.90	0.808	0.213	25.33	0.875	0.147
wo/ add	21.82	0.808	0.215	25.12	0.875	0.153
wo/ reset	21.77	0.806	0.215	25.25	0.875	0.149

TABLE IV

THE ABLATION STUDIES OF THE MULTI-VIEW NUMBER. WE REPORT RESULTS ON TWO REPRESENTATIVE DATASETS.

Views	Mip-NeRF360			Tanks&Temples		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
1	27.23	0.807	0.229	23.13	0.826	0.207
2	27.55	0.810	0.218	23.41	0.831	0.191
4	27.77	0.818	0.206	23.69	0.835	0.178
8	**27.96**	**0.822**	**0.196**	**23.86**	**0.841**	**0.166**
12	27.78	0.815	0.196	23.72	0.838	0.174

strategy. The red boxed areas in the figure highlight the zoomed-in results. From the comparison, it is evident that our method outperforms the others in terms of similarity to the ground truth. For example, in the 'garden' scene, other methods exhibit noticeable distortion in the zoomed-in red-boxed region, while our method effectively captures clearer details, demonstrating its superiority in detail reconstruction.

To further validate the capability of our model in detail modeling, we present additional texture visualization results. As shown in Fig. 4, other methods fail to generate the texture structure of the floor, and the generated table exhibits image distortion. The results demonstrate that our method exhibits significant advantages in texture detail representation. Additionally, Fig. 5 presents depth maps, where we can clearly observe that our method achieves higher accuracy in depth estimation. For example, in the 'room' scene, our method clearly captures the contours of the curtains, while the contours in other methods are extremely faint. These results strongly confirm the superior performance of our model in capturing fine details.

C. Ablation Studies and Analyses

Module Gain. To better verify the effectiveness of our components, we provide a comprehensive ablation study in Tab. II. we validate each component of the method trained on the Mip-NeRF360 dataset[8] using randomly initialised point clouds and SfM initialized point clouds.. We compare the DPC module and the MVC module to the baseline[5]. Specifically, dual-view point correction precisely identifies and corrects Gaussian points that cause rendering errors, thereby eliminating local distortions caused by initialization errors or sparse point clouds. Meanwhile, multi-view constraints, by incorporating supervision from multiple views, prevent overfitting to a single view and effectively enhance scene texture details. The ablation results all show that significantly better results can be achieved using our modules. We also present an ablation study in Fig. 6, where we incrementally integrate our proposed components onto the baseline to demonstrate the effectiveness of our approach. The results

Fig. 6. **Visualization comparisons of the ablation of the proposed components.** We employ RAIN-GS as our baseline and improve it by gradually integrating our proposed components into it. It can be observed that our method gradually improves the novel view synthesis performance of the one.

TABLE V

EFFECT OF DIFFERENT TRAINING VIEW RATIOS IN THE GARDEN AND BONSAI.

Scene	Method	25%		50%		75%		100%	
		PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
Garden	RAIN-GS	22.10	0.208	25.26	0.139	26.31	0.124	26.88	0.114
	Ours	22.65	0.199	26.01	0.127	27.53	0.110	27.94	0.093
Bonsai	RAIN-GS	27.90	0.251	30.36	0.223	30.75	0.222	31.55	0.218
	Ours	29.79	0.222	32.27	0.189	32.69	0.186	33.51	0.173

demonstrate that each of our two modules achieves good performance individually, and their combination further enhances the overall performance.

Points Manipulation. To investigate the effects of point operations in DPC, including point addition and error point reset, we conducted experiments on the Train and Truck scenes from the Tanks&Temples dataset. All experimental analyses were performed under the condition of randomly initialized point clouds. The results shown in Tab. III indicate that: (1) each operation leads to a positive gain, demonstrating the effectiveness of DPC; (2) the point addition operation effectively fills the under-optimized areas, which may have been overlooked in 3DGS, but further captures more geometric details, thereby improving overall performance; (3) resetting points in certain regions not only provides an opportunity to correct potential erroneous points but also creates conditions for geometric calibration, ultimately enhancing the model's stability and accuracy.

Multi-View Number. To validate the effectiveness of MVC, we conducted an ablation study under the same condition of randomly initialized point clouds, similar to DPC, show in Tab. IV. As described in the method section, MVC consists of two key components: Multi-Views Joint Training and Cross-Ray Densification. We compared the baseline method with our proposed multi-view training-enhanced approach. It can be observed that the introduction of multi-view training led to a substantial improvement in novel view synthesis quality. However, as the amount of views exceeds a certain threshold, its performance starts to degrade. This is because an excessive number of views leads to an increase in similarity between the sampled view regions, which causes 3D Gaussians to overfit to certain areas of the scene. Therefore, a moderate or limited amount of views are more favorable for the optimization of 3DGS.

Sparse Training Images. We conducted further ablation studies on the Mip-NeRF 360 dataset, focusing on indoor (bonsai) and outdoor (garden) scenes, to evaluate the impact of the number of training images on model performance. In Tab. V, we present the results of training RAIN-GS and our method using randomly selected subsets of training images, including 25%, 50%, 75%, and 100% of the total dataset. The experimental results show that, regardless of the number of training images, our method consistently outperforms RAIN-GS in terms of rendering quality, demonstrating exceptional performance. The experimental results suggest that the proposed method exhibits strong robustness to changes in the number of training images, while preserving stable rendering quality across different training scales.

V. CONCLUSIONS

This paper introduces MVC-GS, an innovative approach designed to overcome the limitations of previous methods in capturing the Gaussian point cloud distribution under random initialization conditions, as well as the resulting view overfitting issues. MVC-GS effectively addresses these challenges by integrating dual-view point correction (DPC) and multi-view constraints (MVC), successfully correcting errors in the Gaussian point distribution during the optimization process while generating finer texture details. Extensive experiments on multiple public datasets validate the superiority of the proposed method. In the future, our goal is to extend this method to a broader range of applications, such as dynamic scenes and sparse reconstructions, to achieve high-precision reconstructions based on randomly initialized point clouds.

REFERENCES

[1] Kerbl, Bernhard, Georgios Kopanas, Thomas Leimkuehler and George Drettakis. "3D Gaussian Splatting for Real-Time Radiance Field Rendering." ACM Transactions on Graphics (TOG) 42 (2023): 1 - 14.

[2] Schönberger, Johannes L. and Jan-Michael Frahm. "Structure-from-Motion Revisited." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 4104-4113.

[3] Kheradmand, Shakiba, Daniel Rebain, Gopal Sharma, Weiwei Sun, Jeff Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi and Kwang Moo Yi. "3D Gaussian Splatting as Markov Chain Monte Carlo." ArXiv abs/2404.09591 (2024): n. pag.

[4] Yang, Haosen, Chenhao Zhang, Wenqing Wang, Marco Volino, Adrian Hilton, Li Zhang and Xiatian Zhu. "Gaussian Splatting with Localized Points Management." ArXiv abs/2406.04251 (2024): n. pag.

[5] Jung, Jaewoo, Jisang Han, Honggyu An, Jiwon Kang, Seonghoon Park and Seungryong Kim. "Relaxing Accurate Initialization Constraint for 3D Gaussian Splatting." ArXiv abs/2403.09413 (2024): n. pag.

[6] Yu, Alex, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht and Angjoo Kanazawa. "Plenoxels: Radiance Fields without Neural Networks." 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021): 5491-5500.

[7] Müller, Thomas, Alex Evans, Christoph Schied and Alexander Keller. "Instant neural graphics primitives with a multiresolution hash encoding." ACM Transactions on Graphics (TOG) 41 (2022): 1 - 15.

[8] Barron, Jonathan T., Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan and Peter Hedman. "Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields." 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021): 5460-5469.

[9] Hedman, Peter, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis and Gabriel J. Brostow. "Deep blending for free-viewpoint image-based rendering." ACM Transactions on Graphics (TOG) 37 (2018): 1 - 15.

[10] Knapitsch, Arno, Jaesik Park, Qian-Yi Zhou and Vladlen Koltun. "Tanks and temples." ACM Transactions on Graphics (TOG) 36 (2017): 1 - 13.

[11] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 65(1), 99-106.

[12] Li, Zhan, Zhang Chen, Zhong Li and Yi Xu. "Spacetime Gaussian Feature Splatting for Real-Time Dynamic View Synthesis." 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023): 8508-8520.

[13] Lindenberger, Philipp, Paul-Edouard Sarlin and Marc Pollefeys. "LightGlue: Local Feature Matching at Light Speed." 2023 IEEE/CVF International Conference on Computer Vision (ICCV) (2023): 17581-17592.

[14] Tang, Jiaxiang, Jiawei Ren, Hang Zhou, Ziwei Liu and Gang Zeng. "DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation." ArXiv abs/2309.16653 (2023): n. pag.

[15] Yi, Taoran, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian and Xinggang Wang. "GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models." 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023): 6796-6807.

[16] Luiten, Jonathon, Georgios Kopanas, Bastian Leibe and Deva Ramanan. "Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis." 2024 International Conference on 3D Vision (3DV) (2023): 800-809.

[17] Yang, Ziyi, Xinyu Gao, Wenming Zhou, Shaohui Jiao, Yuqing Zhang and Xiaogang Jin. "Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction." 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023): 20331-20341.

[18] Li, Zhaoshuo, Thomas Muller, Alex Evans, Russell H. Taylor, M. Unberath, Ming-Yu Liu and Chen-Hsuan Lin. "Neuralangelo: High-Fidelity Neural Surface Reconstruction." 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023): 8456-8465.

[19] Liu, Yuan, Peng Wang, Cheng Lin, Xiaoxiao Long, Jie-Chao Wang, Lingjie Liu, Taku Komura and Wenping Wang. "NeRO: Neural Geometry and BRDF Reconstruction of Reflective Objects from Multiview Images." ACM Transactions on Graphics (TOG) 42 (2023): 1 - 22.

[20] Yuan, Yu-Jie, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia and Lin Gao. "NeRF-Editing: Geometry Editing of Neural Radiance Fields." 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022): 18332-18343.

[21] Chen, Yurui, Junge Zhang, Ziyang Xie, Wenye Li, Feihu Zhang, Jiachen Lu and Li Zhang. "S-NeRF++: Autonomous Driving Simulation via Neural Reconstruction and Generation." IEEE Transactions on Pattern Analysis and Machine Intelligence 47 (2024): 4358-4376.

[22] Li, Zhuopeng, Lu Li, Zeyu Ma, Ping Zhang, Junbo Chen and Jian-Zong Zhu. "READ: Large-Scale Neural Scene Rendering for Autonomous Driving." AAAI Conference on Artificial Intelligence (2022).

[23] Li, Lingzhi, Zhen Shen, Zhongshu Wang, Li Shen and Ping Tan. "Streaming Radiance Fields for 3D Video Synthesis." ArXiv abs/2210.14831 (2022): n. pag.

[24] Peng, Sida, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao and Xiaowei Zhou. "Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans." 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020): 9050-9059.

[25] Xian, Wenqi, Jia-Bin Huang, Johannes Kopf and Changil Kim. "Space-time Neural Irradiance Fields for Free-Viewpoint Video." 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020): 9416-9426.

[26] Srinivasan, Pratul P., Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall and Jonathan T. Barron. "NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis." 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020): 7491-7500.

[27] Wu, Haoqian, Zhipeng Hu, Lincheng Li, Yongqiang Zhang, Changjie Fan and Xin Yu. "NeFII: Inverse Rendering for Reflectance Decomposition with Near-Field Indirect Illumination." 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023): 4295-4304.

[28] Zhang, Yuanqing, Jiaming Sun, Xingyi He He, Huan Fu, Rongfei Jia and Xiaowei Zhou. "Modeling Indirect Illumination for Inverse Rendering." 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022): 18622-18631.

[29] Yang, Wenqi, Guanying Chen, Chaofeng Chen, Zhenfang Chen and Kwan-Yee Kenneth Wong. "PS-NeRF: Neural Inverse Rendering for Multi-view Photometric Stereo." ArXiv abs/2207.11406 (2022): n. pag.

[30] Du, Xiaobiao, Yida Wang and Xin Yu. "MVGS: Multi-view-regulated Gaussian Splatting for Novel View Synthesis." ArXiv abs/2410.02103 (2024): n. pag.

[31] Yang, Ziyi, Xinyu Gao, Yang-Tian Sun, Yi-Hua Huang, Xiaoyang Lyu, Wen Zhou, Shaohui Jiao, Xiaojuan Qi and Xiaogang Jin. "Spec-Gaussian: Anisotropic View-Dependent Appearance for 3D Gaussian Splatting." ArXiv abs/2402.15870 (2024): n. pag.

[32] Turkulainen, Matias, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu and Juho Kannala. "DN-Splatter: Depth and Normal Priors for Gaussian Splatting and Meshing." 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2024): 2421-2431.

[33] Li, Haolin, Jinyang Liu, Mario Sznaier and Octavia I. Camps. "3D-HGS: 3D Half-Gaussian Splatting*." 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024): 10996-11005.

[34] Wu, Guanjun, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian and Xinggang Wang. "4D Gaussian Splatting for Real-Time Dynamic Scene Rendering." 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023): 20310-20320.

[35] Huang, Shuo, Shikun Sun, Zixuan Wang, Xiaoyu Qin, Yanmin Xiong, Yuan Zhang, Pengfei Wan, Dingyun Zhang and Jia Jia. "PlacidDreamer: Advancing Harmony in Text-to-3D Generation." Proceedings of the 32nd ACM International Conference on Multimedia (2024): n. pag.

[36] Tancik, Matthew, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron and Henrik Kretzschmar. "Block-NeRF: Scalable Large Scene Neural View Synthesis." 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022): 8238-8248.

[37] Zhang, Zheng, Wenbo Hu, Yixing Lao, Tong He and Hengshuang Zhao. "Pixel-GS: Density Control with Pixel-aware Gradient for 3D Gaussian Splatting." ArXiv abs/2403.15530 (2024): n. pag.