

Plug-and-Play PDE Optimization for 3D Gaussian Splatting: Toward High-Quality Rendering and Reconstruction

Supplementary Material

1. Supplementary Material

1.1. Gaussian Gradient Sensitivity Analysis

Attributes in 3DGS. For some attributes with restricted value ranges, 3DGS applies an activation function to map an unbounded attributes to a bounded value range. Below is a comparison of some attributes and their corresponding rendering properties used in 3DGS:

- position: $\mu_i \in R^3$
- color: $\hat{c}_{i,\phi} = f(\phi, c_i)$,
- opacity: $\hat{o}_i = \text{Sig}(o_i)$,
- scale: $\hat{s}_i = e^{(s_i)}$,
- rotation: $q_i \in R^4$

where μ_i denotes the center position, c_i represents the spherical harmonic coefficients, o_i is the opacity attribute, s_i refers to the scale attributes, and q_i is the quaternion representing the rotation attributes. Here, $\hat{o}_i = \text{Sig}(o_i)$ denotes the opacity, where $\text{Sig}(\cdot)$ represents the sigmoid function, and $\hat{s}_i = e^{(s_i)} \in R^3$ denotes the scaling vector.

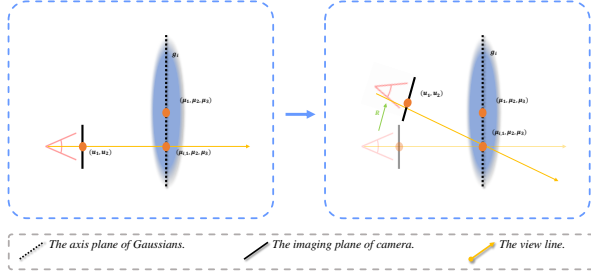


Figure 1. Projection process of 3D Gaussians. Left. The view line of camera is orthogonal to the axis plane of 3D Gaussian. Right. The situation is the same for rotation Gaussians and rotation cameras, so we choose to rotate cameras. After a rotation R , the view line is not orthogonal to the plane.

3DGS to 2D splatting. For a simple non-rotated 3D Gaussian basis function:

$$g_i(\mu) = e^{-\frac{1}{2}(\mu - \mu_i)^T \Sigma^{-1}(\mu - \mu_i)},$$

here $\mu = (\mu_1, \mu_2, \mu_3)$ is the sampling position and $\mu_i = (\mu_{i,1}, \mu_{i,2}, \mu_{i,3})$ is the position of the Gaussian g_i . If we integrate along one of the coordinate axes $(1, 0, 0)$ through the point $(\mu_{i,1}, \mu_{i,2}, \mu_{i,3})$ and the corresponding pixel is $u = (u_1, u_2)$, To simplify, we let $x_i = \mu - \mu_i$, ($x_i =$

$(x_{i,1}, x_{i,2}, x_{i,3})$). We obtain the following integral result:

$$\begin{aligned} \text{splat}_i(u) &= \text{splat}_i(u_1, u_2) = \int g_i(\mu) d\mu_1 = \int e^{-\left(\frac{x_{i,1}^2}{2 \cdot \hat{s}_{i,1}^2} + \frac{x_{i,2}^2}{2 \cdot \hat{s}_{i,2}^2} + \frac{x_{i,3}^2}{2 \cdot \hat{s}_{i,3}^2}\right)} d\mu_1 \\ &= e^{-\left(\frac{x_{i,2}^2}{2 \cdot \hat{s}_{i,2}^2} + \frac{x_{i,3}^2}{2 \cdot \hat{s}_{i,3}^2}\right)} \cdot \sqrt{2\pi} \cdot \hat{s}_{i,1}, \end{aligned}$$

where Σ^{-1} is a 3×3 matrix as:

$$\begin{bmatrix} \frac{1}{2\hat{s}_{i,1}^2} & 0 & 0 \\ 0 & \frac{1}{2\hat{s}_{i,2}^2} & 0 \\ 0 & 0 & \frac{1}{2\hat{s}_{i,3}^2} \end{bmatrix}$$

This is the 2D splatting function at the pixel projected from point $(\mu_{i,1}, \mu_{i,2}, \mu_{i,3})$ in the absence of rotation.

And when we introduce a rotation matrix R , the integral of the rotated function along a line passing through point $(\mu_{i,1}, \mu_{i,2}, \mu_{i,3})$ and parallel to the viewing direction is equivalent to the integral of the non-rotated function along a line passing through point $(\mu_{i,1}, \mu_{i,2}, \mu_{i,3})$ that has been rotated by R .

Then We assume that the direction vector of the integration axis after rotation is $r = (r_1, r_2, r_3)$, where $r_1^2 + r_2^2 + r_3^2 = 1$. So the integral result of the rotated function

$$g_i(\mu) = e^{-x_i^T R^T \Sigma R x_i},$$

along $(1, 0, 0)$, we integrate it

$$\text{splat}_i(u) = \int g_i(\mu) d\mu_1 = \int e^{-\left(\frac{(r_1 \cdot t)^2}{2 \cdot \hat{s}_1^2} + \frac{(r_2 \cdot t + x_{i,2})^2}{2 \cdot \hat{s}_2^2} + \frac{(r_3 \cdot t + x_{i,3})^2}{2 \cdot \hat{s}_3^2}\right)} dt,$$

we simplify it to

$$e^{-\left(\frac{x_{i,2}^2}{2 \cdot \hat{s}_2^2} + \frac{x_{i,3}^2}{2 \cdot \hat{s}_3^2}\right)} \cdot \int e^{-\left(\frac{r_1^2 \cdot t^2}{2 \cdot \hat{s}_1^2} + \frac{r_2^2 \cdot t^2}{2 \cdot \hat{s}_2^2} + \frac{r_3^2 \cdot t^2}{2 \cdot \hat{s}_3^2} + \frac{r_2 \cdot t \cdot x_{i,2}}{\hat{s}_2^2} + \frac{r_3 \cdot t \cdot x_{i,3}}{\hat{s}_3^2}\right)} dt,$$

so we introduce two coefficients $A = \frac{r_1^2}{2 \cdot \hat{s}_1^2} + \frac{r_2^2}{2 \cdot \hat{s}_2^2} + \frac{r_3^2}{2 \cdot \hat{s}_3^2}$, and $B(x_{i,2}, x_{i,3}) = \frac{r_2 \cdot x_{i,2}}{\hat{s}_2^2} + \frac{r_3 \cdot x_{i,3}}{\hat{s}_3^2}$:

$$e^{-\left(\frac{x_{i,2}^2}{2 \cdot \hat{s}_2^2} + \frac{x_{i,3}^2}{2 \cdot \hat{s}_3^2}\right) + \frac{B(x_{i,2}, x_{i,3})^2}{4 \cdot A}} \cdot \int e^{-A(t + \frac{B(x_{i,2}, x_{i,3})}{2A})^2} dt,$$

so we can get

$$\text{splat}(x_i) = \text{splat}_i(u) = e^{-\left(\frac{x_{i,2}^2}{2 \cdot \hat{s}_2^2} + \frac{x_{i,3}^2}{2 \cdot \hat{s}_3^2}\right) + \frac{B(x_{i,2}, x_{i,3})^2}{4 \cdot A}} \cdot \sqrt{\frac{\pi}{A}},$$

as the splatting result of g_i at the pixel u .

Rendering Gradient. For the energy term of rendering supervision, we can write it as:

$$L = \sum_{\mathbf{u}} (\text{render}(\mathbf{u}) - \text{gt}(\mathbf{u}))^2,$$

here $\text{gt}(\cdot)$ is the ground truth of the view and the $\text{render}(\mathbf{u})$ is the render function of Gaussian splatting which can be written as:

$$\text{render}(\mathbf{u}) = \sum_i T_i \hat{\mathbf{c}}_i (\text{Sig}(o_i) \cdot \text{splat}(\mathbf{x}_i)).$$

where $\hat{\mathbf{c}}_i$ is color and $T_i = \prod_{k=1}^{i-1} (1 - \alpha_k)$ is transmittance of g_i , here $\alpha_k = \text{Sig}(o_k) \cdot \text{splat}(\mathbf{x}_k)$ is opacity. We find

$$\frac{\partial L}{\partial \gamma_i} = \sum_{\mathbf{u}} 2(\text{render}(\mathbf{u}) - \text{gt}(\mathbf{u})) \cdot \sum_k \frac{\partial (T_k \hat{\mathbf{c}}_k (\text{Sig}(o_k) * \text{splat}(\mathbf{x}_k)))}{\partial \gamma_i} \sim \frac{r_2 r_3 (\mu_{i,3} - \mu_3)}{\frac{r_1^2}{\hat{s}_{i,1}^2} + \frac{r_2^2}{\hat{s}_{i,2}^2} + \frac{r_3^2}{\hat{s}_{i,3}^2}} \sim \frac{r_2 r_3}{\frac{\hat{s}_{i,3} r_1^2}{\hat{s}_{i,1}^2} + \frac{\hat{s}_{i,3} r_2^2}{\hat{s}_{i,2}^2} + \frac{r_3^2}{\hat{s}_{i,3}}} \leq \frac{r_2 r_3}{2 r_2 r_3 \hat{s}_{i,2}} \sim \hat{s}_{i,2},$$

here k is also the index of gaussians, and $\gamma_i \in \{\mu_i^t, c_i^t, o_i^t, s_i^t, q_i^t\}$ is the attributes of g_i . So we can only discuss

$$\frac{\partial (T_k \cdot \hat{\mathbf{c}}_k \cdot \alpha_k)}{\partial \gamma_i} = \frac{\partial (T_k \hat{\mathbf{c}}_k (\text{Sig}(o_k) * \text{splat}(\mathbf{x}_k)))}{\partial \gamma_i},$$

if we want compare the gradients of different attributes. So when $k = i$, we have:

$$\frac{\partial (T_i \cdot \hat{\mathbf{c}}_i \cdot \alpha_i)}{\partial \hat{\mathbf{c}}_i} = T_i \alpha_i \frac{\partial \hat{\mathbf{c}}_i}{\partial c_i},$$

$$\frac{\partial (T_i \cdot \hat{\mathbf{c}}_i \cdot \alpha_i)}{\partial o_i} = T_i \hat{\mathbf{c}}_i (1 - \text{Sig}(o_i)) \text{Sig}(o_i) \text{splat}(\mathbf{x}_i),$$

$$\frac{\partial (T_i \cdot \hat{\mathbf{c}}_i \cdot \alpha_i)}{\partial \mu_{i,j}} = T_i \hat{\mathbf{c}}_i \text{Sig}(o_i) (\text{splat}(\mathbf{x}_i))_{\mu_{i,j}},$$

$$\frac{\partial (T_i \cdot \hat{\mathbf{c}}_i \cdot \alpha_i)}{\partial s_{i,j}} = T_i \hat{\mathbf{c}}_i \text{Sig}(o_i) (\text{splat}(\mathbf{x}_i))_{s_{i,j}},$$

here $(\cdot)_{\gamma}$ denotes the partial derivative. And when k is different from i :

$$\frac{\partial (T_k \cdot \hat{\mathbf{c}}_k \cdot \alpha_k)}{\partial c_i} = 0,$$

$$\frac{\partial (T_k \cdot \hat{\mathbf{c}}_k \cdot \alpha_k)}{\partial o_i} = -\frac{T_k \hat{\mathbf{c}}_k \alpha_k (1 - \text{Sig}(o_i)) \text{Sig}(o_i) \text{splat}(\mathbf{x}_i)}{1 - \alpha_i},$$

$$\frac{\partial (T_k \cdot \hat{\mathbf{c}}_k \cdot \alpha_k)}{\partial \mu_{i,j}} = -\frac{T_k \hat{\mathbf{c}}_k \alpha_k \text{Sig}(o_i) (\text{splat}(\mathbf{x}_i))_{\mu_{i,j}}}{1 - \alpha_i},$$

$$\frac{\partial (T_k \cdot \hat{\mathbf{c}}_k \cdot \alpha_k)}{\partial s_{i,j}} = -\frac{T_k \hat{\mathbf{c}}_k \alpha_k \text{Sig}(o_i) (\text{splat}(\mathbf{x}_i))_{s_{i,j}}}{1 - \alpha_i},$$

where

$$\text{splat}(\mathbf{x}_i) = e^{-\left(\frac{(\mu_{i,2} - \mu_2)^2}{2 \cdot \hat{s}_{i,2}^2} + \frac{(\mu_{i,3} - \mu_3)^2}{2 \cdot \hat{s}_{i,3}^2}\right) + \frac{B(\mu_{i,2} - \mu_2, \mu_{i,3} - \mu_3)^2}{4 \cdot A}} \cdot \sqrt{\frac{\pi}{A}}.$$

So if we ignore $(\text{splat}(\mathbf{x}_i))_{\mu_{i,j}}$ and $(\text{splat}(\mathbf{x}_i))_{s_{i,j}}$, we can find the remaining parts of the items in the same group are of the same magnitude. For $\alpha_i \sim \text{Sig}(o_i)$ and $\hat{\mathbf{c}}_i \sim \frac{\partial \hat{\mathbf{c}}_i}{\partial c_i} \sim \alpha_i \sim (1 - \text{Sig}(o_i)) \sim \text{splat}(\mathbf{x}_i) \sim 1$. So we can only judge $(\text{splat}(\mathbf{x}_i))_{\mu_{i,j}}$ and $(\text{splat}(\mathbf{x}_i))_{s_{i,j}}$ to compare the gradients.

Let $j = 2$, then we can get

$$(\text{splat}_i)_{\mu_{i,2}} = \frac{1}{\hat{s}_{i,2}^2} \text{splat}_i \cdot \left(\frac{\frac{r_2 r_3 (\mu_{i,3} - \mu_3)}{\hat{s}_{i,3}^2} - \left(\frac{r_1^2}{\hat{s}_{i,1}^2} + \frac{r_3^2}{\hat{s}_{i,3}^2} \right) (\mu_{i,2} - \mu_2)}{\frac{r_1^2}{\hat{s}_{i,1}^2} + \frac{r_2^2}{\hat{s}_{i,2}^2} + \frac{r_3^2}{\hat{s}_{i,3}^2}} \right),$$

here $\text{splat}_i = \text{splat}(\mathbf{x}_i)$. Obviously, we have $\text{splat}_i \sim 1$, $(\mu_{i,3} - \mu_3) \sim \hat{s}_{i,3}$ and $ax^2 + by^2 \geq 2\sqrt{ab}xy$, so we have

and we have $(\mu_{i,2} - \mu_2) \sim \hat{s}_{i,2}$, so

$$\left(\frac{r_1^2}{\hat{s}_{i,1}^2} + \frac{r_3^2}{\hat{s}_{i,3}^2} \right) (\mu_{i,2} - \mu_2) \leq \frac{r_1^2}{\hat{s}_{i,1}^2} + \frac{r_2^2}{\hat{s}_{i,2}^2} + \frac{r_3^2}{\hat{s}_{i,3}^2} \leq (\mu_{i,2} - \mu_2) \sim \hat{s}_{i,2}.$$

According to the definition of equivalence we can get $(\text{splat}_i)_{\mu_{i,2}} \lesssim \frac{1}{\hat{s}_{i,2}}$, and when $r_2 = 0$, $(\text{splat}_i)_{\mu_{i,2}} \sim \frac{1}{\hat{s}_{i,2}}$. So $(\text{splat}_i)_{\mu_{i,2}} \sim \frac{1}{\hat{s}_{i,2}}$. And similarly at $j = 3$, we have $(\text{splat}_i)_{\mu_{i,3}} \sim \frac{1}{\hat{s}_{i,3}}$.

Similarly, we also handle $(\text{splat}_i)_{s_{i,j}}$, as before, we only need to take $j = 2$, since the other value of j is the same as $j = 2$. Noting that $\hat{\mathbf{s}}_i = e^{s_i}$ and $(\hat{\mathbf{s}}_i)_{s_i} = \hat{\mathbf{s}}_i$.

$$(\text{splat}_i)_{s_{i,2}} = -\frac{\left(\frac{r_1^2}{\hat{s}_{i,1}^2} + \frac{r_3^2}{\hat{s}_{i,3}^2} \right) r_2^2 (\mu_{i,2} - \mu_2)^2}{2 A^2 \hat{s}_{i,2}^4} + \frac{\left(\frac{r_1^2}{\hat{s}_{i,1}^2} + \frac{r_3^2}{\hat{s}_{i,3}^2} \right) (\mu_{i,2} - \mu_2)^2}{A \hat{s}_{i,2}^2} - \frac{2 r_2 r_3 (\mu_{i,2} - \mu_2) (\mu_{i,3} - \mu_3) \left(\frac{\hat{s}_{i,3}^2 r_1^2}{\hat{s}_{i,1}^2} + r_3^2 \right)}{\hat{s}_{i,2}^2 \hat{s}_{i,3}^4 A^2}.$$

We analyze each item step by step. For $(\mu_{i,2} - \mu_2) \sim \hat{s}_{i,2}$,

$$\frac{\left(\frac{r_1^2}{\hat{s}_{i,1}^2} + \frac{r_3^2}{\hat{s}_{i,3}^2} \right) r_2^2 (\mu_{i,2} - \mu_2)^2}{2 A^2 \hat{s}_{i,2}^4} \lesssim \frac{r_2^2}{2 A \hat{s}_{i,2}^2} \sim 1,$$

Similarly, we obtain:

$$\frac{\left(\frac{r_1^2}{\hat{s}_{i,1}^2} + \frac{r_3^2}{\hat{s}_{i,3}^2} \right) (\mu_{i,2} - \mu_2)^2}{A \hat{s}_{i,2}^2} \sim 1$$

The third item is slightly more complex, so we will handle it in two parts. Firstly, We will address the first part:

$$E_1 := \frac{2 r_1^2 r_2 r_3 (\mu_{i,2} - \mu_2) (\mu_{i,3} - \mu_3)}{\hat{s}_{i,1}^2 \hat{s}_{i,2}^2 \hat{s}_{i,3}^2 A^2}.$$

Note $E_1 = \frac{2\hat{s}_{i,2}^2\hat{s}_{i,3}^2r_1^2\hat{s}_{i,1}^2r_2r_3(\mu_{i,2}-\mu_2)(\mu_{i,3}-\mu_3)}{(\hat{s}_{i,2}^2\hat{s}_{i,3}^2r_1^2+\hat{s}_{i,1}^2\hat{s}_{i,3}^2r_2^2+\hat{s}_{i,1}^2\hat{s}_{i,2}^2r_3^2)^2}$, so a natural thinking is dividing it into two parts:

$$E_1 = \frac{2\hat{s}_{i,2}^2\hat{s}_{i,3}^2r_1^2}{\hat{s}_{i,2}^2\hat{s}_{i,3}^2r_1^2 + \hat{s}_{i,1}^2\hat{s}_{i,3}^2r_2^2 + \hat{s}_{i,1}^2\hat{s}_{i,2}^2r_3^2} \cdot \frac{\hat{s}_{i,1}^2r_2r_3(\mu_{i,2}-\mu_2)(\mu_{i,3}-\mu_3)}{\hat{s}_{i,2}^2\hat{s}_{i,3}^2r_1^2 + \hat{s}_{i,1}^2\hat{s}_{i,3}^2r_2^2 + \hat{s}_{i,1}^2\hat{s}_{i,2}^2r_3^2}.$$

For $(\mu_{i,2}-\mu_2) \sim \hat{s}_{i,2}$ and $(\mu_{i,3}-\mu_3) \sim \hat{s}_{i,3}$, we have:

$$\frac{2\hat{s}_{i,2}^2\hat{s}_{i,3}^2r_1^2}{\hat{s}_{i,2}^2\hat{s}_{i,3}^2r_1^2 + \hat{s}_{i,1}^2\hat{s}_{i,3}^2r_2^2 + \hat{s}_{i,1}^2\hat{s}_{i,2}^2r_3^2} \lesssim 1, \\ \frac{\hat{s}_{i,1}^2r_2r_3(\mu_{i,2}-\mu_2)(\mu_{i,3}-\mu_3)}{\hat{s}_{i,2}^2\hat{s}_{i,3}^2r_1^2 + \hat{s}_{i,1}^2\hat{s}_{i,3}^2r_2^2 + \hat{s}_{i,1}^2\hat{s}_{i,2}^2r_3^2} \lesssim 1.$$

Similarly, we address the second part:

$$E_2 := \frac{2r_2r_3^3(\mu_{i,2}-\mu_2)(\mu_{i,3}-\mu_3)}{\hat{s}_{i,2}^2\hat{s}_{i,3}^4A^2}.$$

Note $E_2 = \frac{2r_2r_3(\mu_{i,3}-\mu_3)r_3^2(\mu_{i,2}-\mu_2)}{(\frac{\hat{s}_{i,3}^2\hat{s}_{i,2}^2r_1^2}{\hat{s}_{i,1}^2} + \frac{\hat{s}_{i,3}^2r_2^2}{\hat{s}_{i,2}} + \hat{s}_{i,2}r_3^2)^2}$, so we divide it

into two parts:

$$E_2 = \frac{2r_2r_3(\mu_{i,3}-\mu_3)}{\frac{\hat{s}_{i,3}^2\hat{s}_{i,2}^2r_1^2}{\hat{s}_{i,1}^2} + \frac{\hat{s}_{i,3}^2r_2^2}{\hat{s}_{i,2}} + \hat{s}_{i,2}r_3^2} \cdot \frac{r_3^2(\mu_{i,2}-\mu_2)}{\frac{\hat{s}_{i,3}^2\hat{s}_{i,2}^2r_1^2}{\hat{s}_{i,1}^2} + \frac{\hat{s}_{i,3}^2r_2^2}{\hat{s}_{i,2}} + \hat{s}_{i,2}r_3^2}.$$

Then we have:

$$\frac{2r_2r_3(\mu_{i,3}-x_{30})}{\frac{\hat{s}_{i,3}^2\hat{s}_{i,2}^2r_1^2}{\hat{s}_{i,1}^2} + \frac{\hat{s}_{i,3}^2r_2^2}{\hat{s}_{i,2}} + \hat{s}_{i,2}r_3^2} \lesssim 1, \\ \frac{r_3^2(\mu_{i,2}-x_{20})}{\frac{\hat{s}_{i,3}^2\hat{s}_{i,2}^2r_1^2}{\hat{s}_{i,1}^2} + \frac{\hat{s}_{i,3}^2r_2^2}{\hat{s}_{i,2}} + \hat{s}_{i,2}r_3^2} \lesssim 1.$$

So we have $(splat_i)_{s_{i,2}} \lesssim 1$, and when $r_2 = r_3 = 0$, $(splat_i)_{s_{i,2}} \sim 1$. According to the definition of equivalence, $(splat_i)_{s_{i,2}} \sim 1$. And the same as other value of j .

So we can find in certain i and k , we have the relation that:

$$\hat{s}_{i,j} \frac{\partial(T_k \hat{c}_k \alpha_k)}{\partial \mu_{i,j}} \sim \frac{\partial(T_k \hat{c}_k \alpha_k)}{\partial c_i} \sim \frac{\partial(T_k \hat{c}_k \alpha_k)}{\partial o_i} \sim \frac{\partial(T_k \hat{c}_k \alpha_k)}{\partial s_{i,j}}.$$

Specially, the rotation attribute q_i^t which is a quaternion array is updating as:

$$q_i^{t+1} = q_i^t + \Delta q_i^t, \\ \|\Delta q_i^t\| = \left\| \frac{\frac{\partial L}{\partial q_i} + q_i^t}{\left\| \frac{\partial L}{\partial q_i} + q_i^t \right\|} - q_i^t \right\| \leq 2 \max(\|q_i\|)$$

By the definition of a quaternionic array we have $\|q_i\| \leq 1$, then we obtain $\Delta q_i^t \sim 1$. So we can get

$$\hat{s}_i \frac{\partial L}{\partial \mu_i} \sim \frac{\partial L}{\partial c_i} \sim \frac{\partial L}{\partial o_i} \sim \frac{\partial L}{\partial s_i} \sim \Delta q_i^t,$$

so if we define the direction vector of Δq_i^t as $r_{q,i}^t$, by the definition of partial derivatives, the updating of rotation attribute have

$$\Delta q_i^t = \frac{\partial L}{\partial(q_i^t \cdot r_{q,i}^t)}.$$

When the scales of Gaussians are small, we can get

$$\frac{\partial L}{\partial \mu_i} \gg \frac{\partial L}{\partial c_i} \sim \frac{\partial L}{\partial o_i} \sim \frac{\partial L}{\partial s_i} \sim \frac{\partial L}{\partial(q_i^t \cdot r_{q,i}^t)}.$$

That means the Gaussians will more willing to change their places to reduce the energy, which will more likely cause the large-scale random drift and leading the local minimum. To achieve optimal results, we aim for all variables to change in a relatively consistent manner. To this end, it is natural to consider decelerating the changes in the positional attributes of the 3D Gaussians. Specifically, we formulate the 3DGS optimization procedure as the discretization of a Partial Differential Equation (PDE) and employ the viscosity coefficient, allowing spatial positions to absorb and gradually release the positional gradients of the 3D Gaussians.

And due to the Gaussian function property, we have

$$\sum_{splat_k \geq \epsilon} splat_k \sim 1,$$

where ϵ is the 0.99 confidence bound for the Gaussian function. So for the same 3D Gaussian g_k at different scales \hat{s}_k , we have

$$\sum_{splat_k \geq \epsilon} \frac{\partial(T_k \cdot \hat{c}_k \cdot \alpha_k)}{\partial \mu_k} = \sum_{splat_k \geq \epsilon} O\left(\frac{1}{\hat{s}_k}\right) T_k \hat{c}_k \text{Sig}(o_k) splat_k = O\left(\frac{1}{\hat{s}_k}\right),$$

and the position gradient $\frac{\partial L}{\partial \mu_{i,j}}$ of g_k is proportional to $\sum \frac{\partial(T_k \cdot \hat{c}_k \cdot \alpha_k)}{\partial \mu_k}$, so we have the relationship of position gradients between different 3D Gaussians:

$$\hat{s}_{i,j} \frac{\partial L}{\partial \mu_{i,j}} \sim \hat{s}_{k,j} \frac{\partial L}{\partial \mu_{k,j}}.$$

Observation. 3DGS represents a complex scene as a set of 3D Gaussians. However, various 3DGS methods [3, 13] suffer from the common limitation of blurring and floaters due to the reconstruction of redundant and ambiguous geometric structures, leading to degraded rendering and reconstruction quality. We attribute the blurring and floaters to the occlusion of redundant large Gaussians and the ambiguity of small Gaussians, as shown in Fig. 2. The large 3D

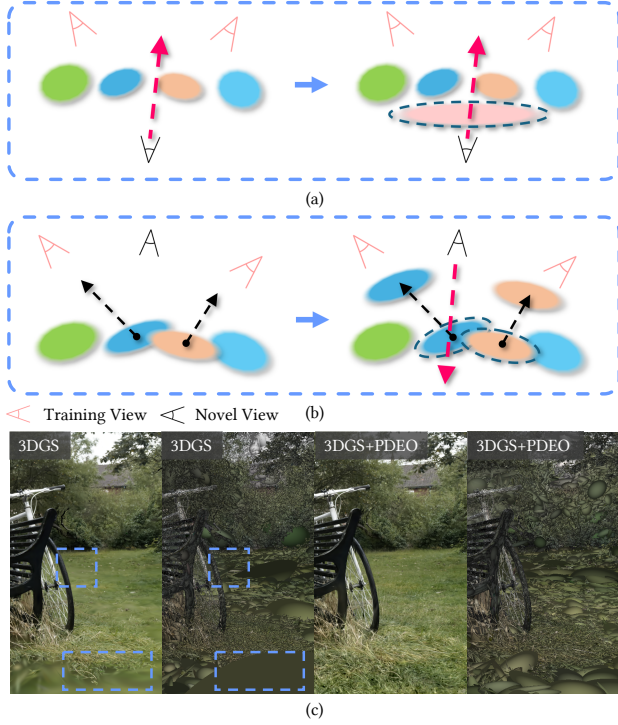


Figure 2. Optimization of 3D Gaussians. (a) The redundancy of large 3D Gaussians. (b) The ambiguity of small 3D Gaussians. (c) Visualization results.

Gaussians fail to capture high-frequency details and tend to obstruct other Gaussians, resulting in redundancy and manifesting as blurring in the novel view. For small 3D Gaussians, due to the unstable gradient, floaters tend to appear in regions of the scene that are poorly observed, as the Gaussians tend to shift their positions toward observed views during the 3DGS optimization process, thereby resulting in ambiguous geometric structures.

1.2. The The velocity Voxel in Space

Building. We aim to construct a loss function that considers the positional gradient field for a particle located at spatial position μ :

$$v(\mu) = \sigma \frac{\partial L}{\partial \mu}.$$

However, since the attributes of the particles are unknown, this term cannot be directly calculated. Moreover, as the motion equations in 3DGS are based on the gradients of the scene rendering results, particles with different color attributes will exhibit different movement tendencies, typically lacking a linear relationship. Therefore, simply averaging attributes of the particles near a spatial location and then using this averaged set of attributes to compute the positional gradient is meaningless.

We aim for the velocity field at μ to indicate the most

likely displacement of a Gaussian sphere at this location. Therefore, we choose to construct and update the velocity field by the local average velocity of μ , as shown in Eq.31.

This approach is mathematically meaningful: if we consider the positional gradients of 3D Gaussians as points in a three-dimensional space, the positional gradients of 3D Gaussians near μ form a point cloud in this velocity field. We want $v(\mu)$ to be positioned at the center of the largest cluster within this point cloud, which the arithmetic mean can achieve. Additionally, the arithmetic mean can counterbalance the impact of large-scale Brownian motion on the spatial velocity field caused by abrupt changes in positional gradients. Then we obtain the spatial velocity field $v(\mu)$.

Total Impact of Gradient Field. We renew the field by $\Delta v_n^t = \frac{1}{|R_n^t|} \sum_{g_i \in R_n^t} \Delta \mu_i^t$ in Eq.31. So we have the total impact of Δv_n^t in the field by adding it in every steps:

$$I(\Delta v_n^t) = \sum_{l \geq t} \Delta v_n^t(l),$$

where $\Delta v_n^t(l)$ means portion of $v_n^t(l)$ occupied by Δv_n^t . So we have

$$I(\Delta v_n^t) = (1 - \lambda_g \Delta v_n^t \sum_{l \geq t} \lambda_g^{(l-t+1)}) \rightarrow \Delta v_n^t.$$

Therefore, regardless of the coefficient λ_g , each updated vector will have a weight of 1 in the overall influence on the field throughout spacetime. At the same moment, the total weight of this vector on the gradient is always 1. Thus, no matter the chosen weighting coefficient, the value of this velocity field can naturally represent the magnitude of the gradient.

1.3. More Results

Table 1. The table below presents a quantitative comparison of training times between 3DGS and our method on the Mip-NeRF 360 dataset.

Method	3DGS	+PDEO	2DGS	+PDEO	RaDeGS	+PDEO	MCMC	+PDEO
Time (Min)	33.7	38.1	45.3	20.8	17.1	20.2	41.9	35.2
Mem (MB)	295	186	476	63.8	536	187	714	198

References

- [1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 5
- [2] D. Chen. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE*, 2024. 5
- [3] Andreas Geiger, Shenghua Gao, Anpei Chen, Zehao Yu, and Binbin Huang. 2d gaussian splatting for geometrically accurate radiance fields. *ACM SIGGRAPH 2024 Conference Papers*, 2024. 3

Table 2. Quantitative results on Mip-NeRF 360 [1], Tanks&Temples [9] and Scanet++ [12] for Novel view synthesis. We integrate the proposed PDEO into state-of-the-art 3DGS-based methods for novel view synthesis, and compare it with several methods [2–5, 7, 8, 10, 11, 13–15]. The best results are highlighted in bold. PDEO consistently improves the performance.

Dataset	Mip-NeRF360[1]					Tanks&Temples[9]					Scanet+++[12]				
Method	<i>PSNR</i> ↑	<i>SSIM</i> ↑	<i>LPIPS</i> ↓	<i>Mem</i> ↓	<i>FPS</i> ↑	<i>PSNR</i> ↑	<i>SSIM</i> ↑	<i>LPIPS</i> ↓	<i>Mem</i> ↓	<i>FPS</i> ↑	<i>PSNR</i> ↑	<i>SSIM</i> ↑	<i>LPIPS</i> ↓	<i>Mem</i> ↓	<i>FPS</i> ↑
3DGS	27.77	0.827	0.244	295	163.1	21.63	0.768	0.322	299	44.3	27.83	0.911	0.185	192	74.2
GES	27.71	0.844	0.224	369	106.3	21.59	0.768	0.330	162	64.1	27.86	0.912	0.190	94.1	97.9
AbaGS	27.81	0.850	0.207	804	125.1	21.37	0.755	0.326	340	40.0	27.67	0.907	0.185	121	101.8
MipGS	27.98	0.858	0.213	303	108.5	20.98	0.757	0.326	357	52.1	27.80	0.913	0.177	224	135.2
2DGS	27.42	0.841	0.228	476	42.3	21.02	0.756	0.357	188	21.8	27.91	0.911	0.196	107	36.8
RaDeGS	28.03	0.866	0.198	536	118.6	20.80	0.750	0.345	239	57.5	27.97	0.911	0.180	165	103.4
MCMC	27.91	0.845	0.186	714	40.4	21.03	0.744	0.318	691	55.7	28.01	0.918	0.182	470.3	52.5
SpecGS	27.96	0.866	0.173	1147	7.9	21.02	0.751	0.322	498	19.7	27.89	0.912	0.195	159	56.1
PGSR	27.77	0.864	0.179	331	156.1	21.67	0.770	0.313	343	52.4	27.87	0.912	0.178	208	85.9
MILO	28.11	0.871	0.178	103	189.4	21.53	0.783	0.301	110	85.2	28.02	0.917	0.176	93.0	140.3
QGS	27.52	0.839	0.226	325	90.9	20.95	0.762	0.319	323	40.7	27.80	0.911	0.188	195	33.6
3DGS+PDEO	27.78	0.831	0.242	186	225.5	21.89	0.768	0.320	125	146.9	27.87	0.911	0.190	66.7	260.0
GES+PDEO	27.99	0.834	0.232	133	166.1	22.08	0.768	0.325	97.0	176.0	27.92	0.911	0.192	53.5	283.0
MipGS+PDEO	28.08	0.870	0.211	137	167.9	22.12	0.761	0.320	79.0	148.5	27.91	0.913	0.169	48.5	254.5
2DGS+PDEO	27.42	0.832	0.273	63.8	94.5	21.03	0.749	0.363	100	64.6	27.93	0.911	0.195	102	81.7
RaDeGS+PDEO	28.16	0.852	0.213	187	171.1	22.61	0.768	0.332	95.1	118.4	28.06	0.911	0.189	65.0	227.9
MCMC+PDEO	28.12	0.833	0.213	198	73.3	22.77	0.780	0.295	210	73.9	28.23	0.919	0.182	212	86.9
SpecGS+PDEO	28.81	0.875	0.173	99.6	65.4	22.16	0.780	0.316	345	28.2	28.10	0.919	0.185	115	66.8
PGSR+PDEO	27.79	0.855	0.178	152	183.9	22.00	0.769	0.301	148	103.7	28.13	0.913	0.170	119	158.5
MILO+PDEO	28.35	0.873	0.180	78.4	201.5	21.88	0.788	0.297	80.1	153.6	28.25	0.919	0.169	64.5	249.8
QGS+PDEO	27.60	0.742	0.199	149	144.6	21.63	0.770	0.295	136	99.4	27.92	0.911	0.185	99.6	129.4
3DGS(rander)	26.61	0.764	0.318	258	78.8	20.84	0.734	0.380	261	66.1	27.55	0.908	0.202	164.9	92.4
+PDEO(rander)	27.75	0.825	0.233	89.4	122.3	21.71	0.745	0.361	101	71.5	27.64	0.908	0.199	59.5	138.0
MCMC(rander)	27.62	0.832	0.203	473	55.6	21.00	0.735	0.333	469	35.8	27.92	0.918	0.187	354	60.7
+PDEO(rander)	27.85	0.861	0.187	189	85.9	22.64	0.771	0.321	187	54.6	27.98	0.918	0.182	98.7	63.7

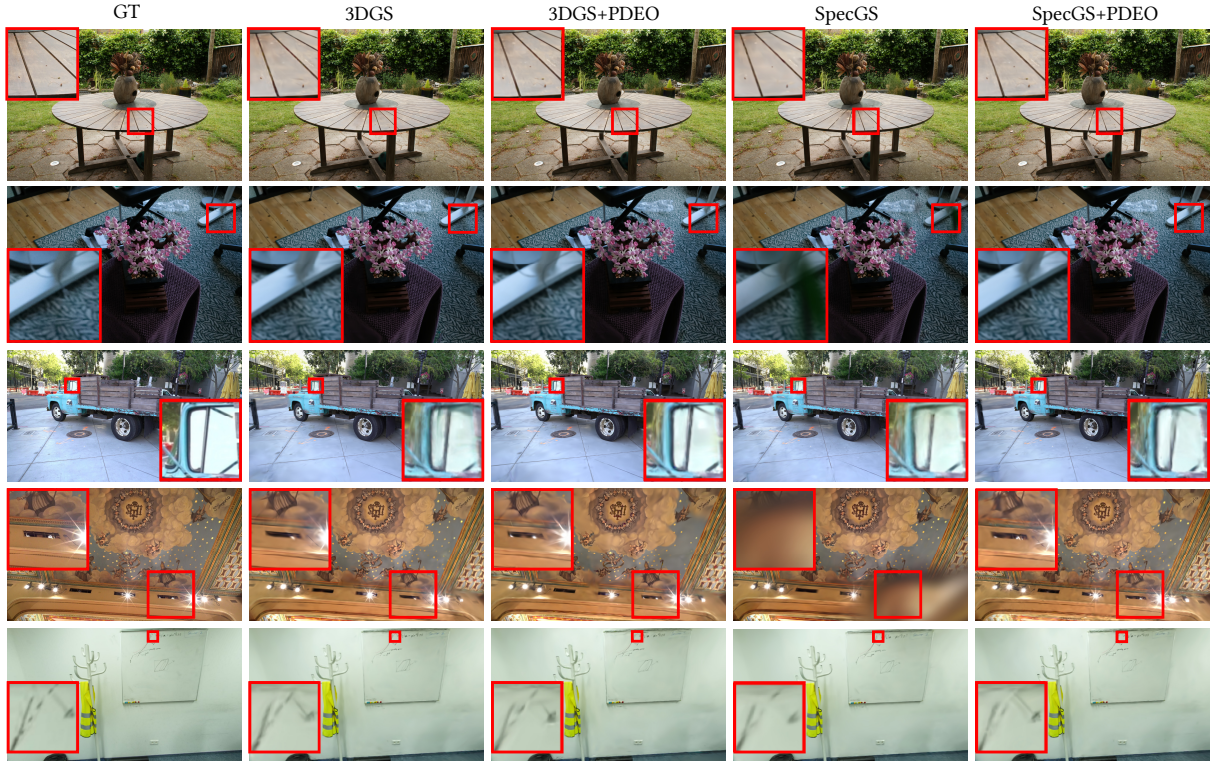


Figure 3. Qualitative comparisons of different methods on scenes from Mip-NeRF360 [1] and Tanks&Temples [9] and Scanet+++[12] datasets for novel view synthesis. PEDO significantly reduces artifacts and floaters while improving rendering quality.

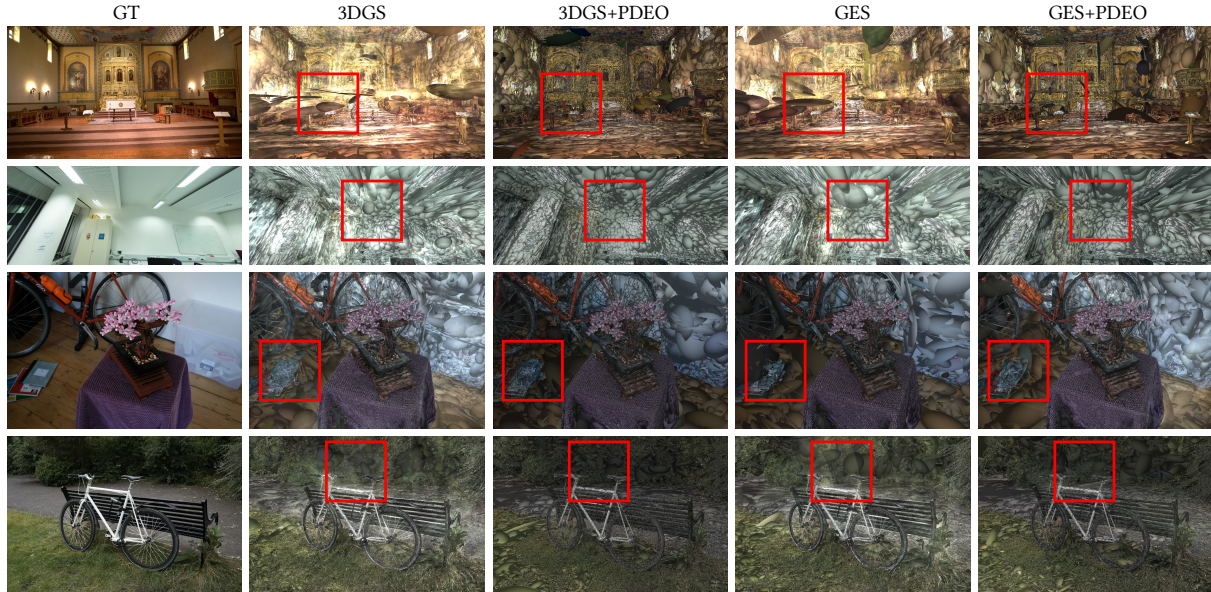


Figure 4. Visualization of Gaussian ellipsoids. PEDO eliminates floater Gaussians and recovers fine geometric details.

- [5] Abdullah Hamdi, Luke Melas-Kyriazi, Jinjie Mai, Guocheng Qian, Ruoshi Liu, Carl Vondrick, Bernard Ghanem, and Andrea Vedaldi. Ges: Generalized exponential splatting for efficient radiance field rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19812–19822, 2024. [5](#)
- [6] Rasmus Jensen, Anders Dahl, George Vogiatis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 406–413, 2014. [7](#)
- [7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):139:1–139:14, 2023. [5](#)
- [8] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. pages 80965–80986, 2024. [5](#)
- [9] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. [5](#), [7](#)
- [10] Ziyi Yang, Xinyu Gao, Yang-Tian Sun, Yihua Huang, Xiaoyang Lyu, Wen Zhou, Shaohui Jiao, Xiaojuan Qi, and Xiaogang Jin. Spec-gaussian: Anisotropic view-dependent appearance for 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:61192–61216, 2024. [5](#)
- [11] Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details for 3d gaussian splatting. *arXiv preprint arXiv:2404.10484*, 2024. [5](#)
- [12] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. [5](#)
- [13] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2024. [3](#), [5](#)
- [14] Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. Rade-gs: Rasterizing depth in gaussian splatting. *arXiv preprint arXiv:2406.01467*, 2024.
- [15] Z. Zhang. Quadratic gaussian splatting: High quality surface reconstruction with second-order geometric primitives. In *ICCV*, 2025. [5](#)

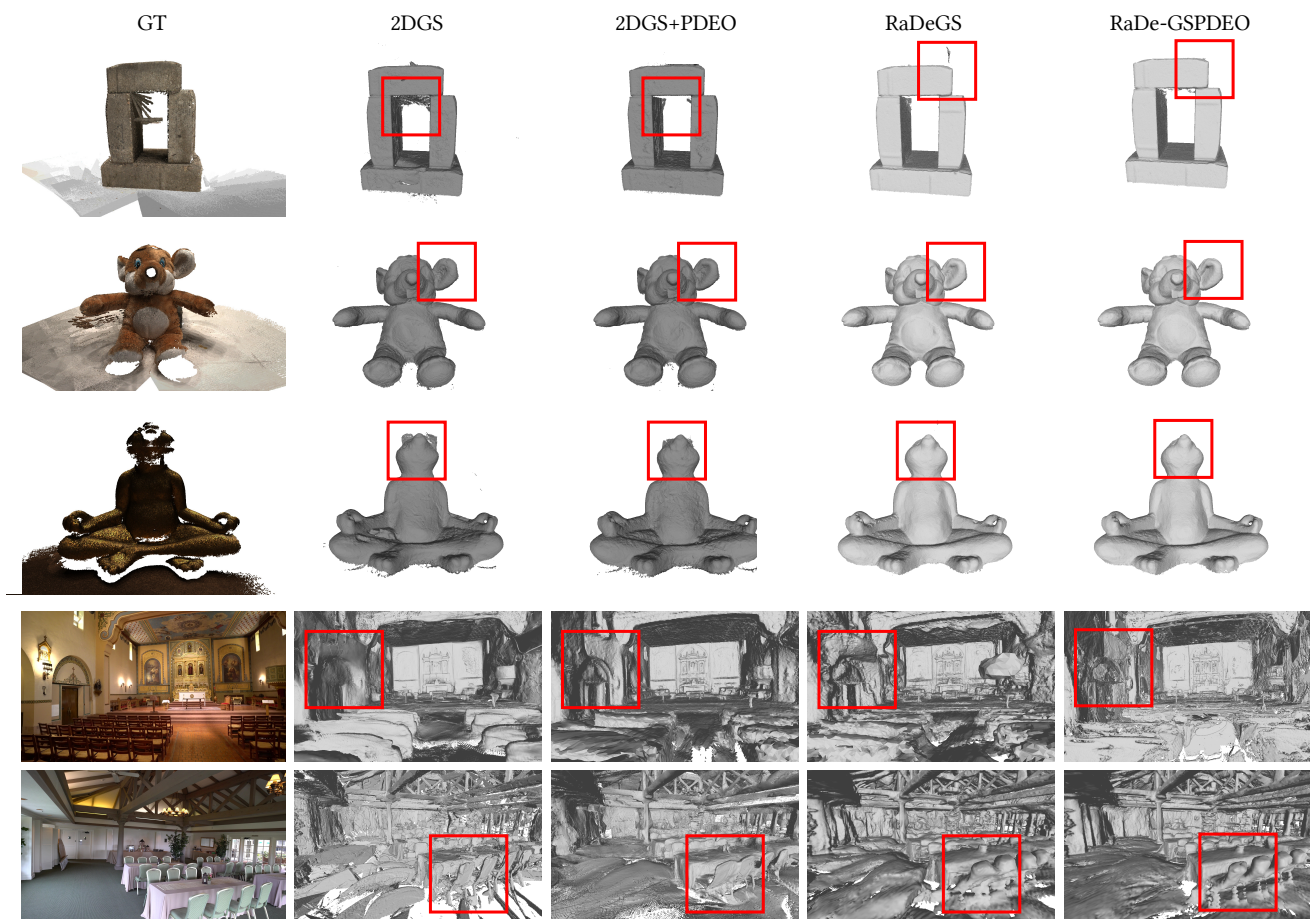


Figure 5. Qualitative comparisons of different methods on scenes from Tanks&Temples [9] and DTU [6] datasets for surface reconstruction. PEDO improves the quality of the reconstruction.