Softmax-GS: Generalized Gaussians Learning When to Blend or Bound

Supplementary Material

6. More implementation details

For simple-geometry fitting experiments, we place a camera at the origin facing the +z direction, and initialize four black Gaussians with identical shapes slightly apart at a depth of 1 unit in front of the camera. Optimization is run for 10K steps without opacity reset against the target image using default rendering losses. We set the learning rates for α, β, and γ to 0.0003, 0.0003, and 0.0002, respectively. For real-world benchmarks, we set the learning rates for α, β, and γ to 0.0008, 0.008, and 0.0004, respectively. To accelerate training, we adopt the tile culling strategies proposed in [7, 25].

To improve stability, we further introduce a variance regularization on β and γ along each camera ray. This term penalizes abrupt changes in these parameters when the depth ordering of Gaussians changes. Specifically, the variance is computed over Gaussians intersecting a ray and weighted by the inverse distance to the nearest (front-most) Gaussian, so that the regularization primarily affects the first visible cluster. The weights for both variance terms are set to $\lambda = 0.01$.

For our quantitative comparison, since previous works may have used different image resolutions for their benchmarks, we download the official code of the baselines and re-run all algorithms at a consistent resolution. We keep all methods' default hyperparameters unchanged, and all scenes are optimized for 30K steps. For GS-MCMC [14], SSS [31], and Softmax-GS-MCMC, all of which employ Monte Carlo-based optimization, we initialize from the SfM point cloud and cap the maximum number of Gaussians to match that of the original 3D GS.

7. Measurement of view consistency

We evaluate the view consistency of Softmax-GS following the protocol from StopThePop [25]. We first render a video from the reconstructed Gaussians and, for each frame $\mathbf{I}_i$, pair it with the frame seven steps ahead, $\mathbf{I}_{i+7}$. We then apply the RAFT optical-flow method [28] to warp $\mathbf{I}_i$ to $\mathbf{I}_{i+7}$, producing $\hat{\mathbf{I}}_i$, and compute MSE and ꟻLIP$_7$ similarity metrics between $\hat{\mathbf{I}}_i$ and $\mathbf{I}_{i+7}$. Because [25] does not disclose the camera-trajectory generation process, we simply insert 128 interpolated frames between each pair of target frames. Following [25], we run RAFT using the model pre-trained on SINTEL [2] and resize the input frames to SINTEL's resolution of 1024×436.

We present detailed evaluation results on five representative scenes in Table 4, reporting both rendering quality (PSNR) and view-consistency metrics (MSE and ꟻLIP). All methods are evaluated under the same protocol. We observe

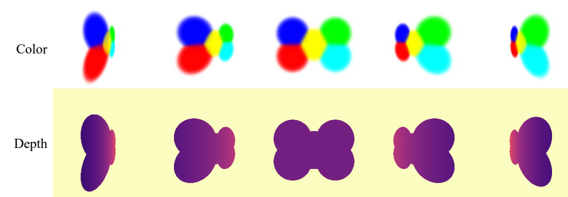

Figure 8. Pixel-wise depth rendering of synthetic pattern.

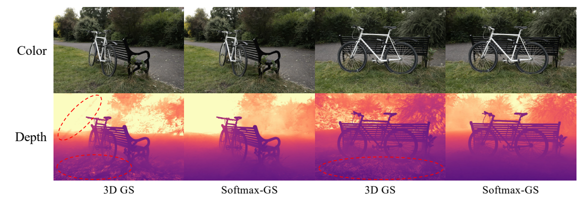

Figure 9. Depth rendering comparison with 3D GS.

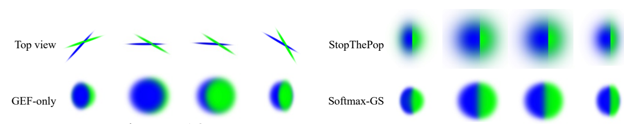

Figure 10. Two Gaussians crossing at 30° angle.

that Softmax-GS, by addressing overlapping Gaussians, and StopThePop, by introducing a re-sorting mechanism, each improves view consistency individually. However, as shown in the rendered videos provided the supplementary materials, Softmax-GS exhibits significantly fewer "floater" artifacts compared to StopThePop (especially in the `kitchen` scene), indicating that our approach leads to more stable optimization behavior.

8. Depth rendering of Softmax-GS

We visualize the depth rendering of Softmax-GS in Fig. 8 for a synthetic pattern and in Fig. 9 for a real-world scene. Note the smooth depth transitions at Gaussian boundaries.

9. Non-coplanar intersection

We visualize two Gaussians crossing at 30° in Fig. 10. In contrast to the popping artifacts of 3D GS and the abrupt color changes of STP, Softmax-GS produces smooth, flicker-free transitions at the crossing. Note Softmax-GS effectively merges intersecting Gaussians into a single surface, consistent with the physical assumption that real-world object surfaces do not cross.

Method	bicycle			counter			kitchen			train			truck		
	PSNR↑	MSE↓	ꟻLIP$_7$ ↓	PSNR↑	MSE↓	ꟻLIP$_7$ ↓	PSNR↑	MSE↓	ꟻLIP$_7$ ↓	PSNR↑	MSE↓	ꟻLIP$_7$ ↓	PSNR↑	MSE↓	ꟻLIP$_7$ ↓
3D GS	25.13	0.017	0.032	29.26	0.0037	0.016	31.42	0.0045	0.019	22.19	0.0126	0.048	25.12	0.0106	0.031
StopThePop	25.14	0.016	0.028	28.91	0.0037	0.015	31.33	0.0047	0.019	21.45	0.0121	0.044	24.56	0.0105	0.030
Softmax-GS	25.24	0.016	0.031	29.38	0.0037	0.016	32.11	0.0045	0.018	22.62	0.0124	0.044	25.55	0.0105	0.029

Table 4. View consistency measurement on five representative scenes.

Method	Mip-NeRF360							Tanks&Temples		Deep Blending	
	bicycle	bonsai	counter	garden	kitchen	room	stump	train	truck	drjohnson	playroom
3D GS	25.13 (4.8M)	32.45 (1.0M)	29.26 (1.0M)	27.29 (4.5M)	31.42 (1.5M)	31.79 (1.2M)	26.75 (4.1M)	22.19 (1.1M)	25.12 (2.7M)	28.72 (3.1M)	30.18 (2.0M)
Softmax-GS	25.24 (4.6M)	32.78 (1.0M)	29.38 (1.0M)	27.41 (4.4M)	32.11 (1.4M)	32.18 (1.2M)	26.78 (3.7M)	22.62 (1.1M)	25.55 (2.6M)	29.01 (2.9M)	30.53 (1.9M)

Table 5. Per-scene PSNR and number of Gaussians (in bracket) comparison between 3D GS and Softmax-GS.

10. Per-scene results

Per-scene comparisons of PSNR and Gaussian counts are presented in Table 5, showing that Softmax-GS achieves higher rendering quality with a similar number of Gaussians across all scenes.

11. Full Algorithm

We provide the complete forward-pass of the Softmax-GS algorithm in Algorithm 1.

12. Limitations

Softmax-GS has three main limitations. First, the proposed splatting algorithm is applied only to the first 128 Gaussians along each ray in order to maintain linear complexity in the backward pass. As a result, coverage is incomplete: on Mip-NeRF360 indoor scenes, Softmax-GS accounts for approximately 85% of pixels across test images, while for outdoor scenes the coverage drops to around 70%. Second, the order-invariance mechanism in Softmax-GS does not extend to cases with three or more overlapping Gaussians with distinct colors. This limitation arises from the bookkeeping complexity required to preserve permutation invariance under a strict linear-time constraint. Third, the current formulation struggles with semi-transparent Gaussians. In particular, distant semi-transparent Gaussians can bias the running estimates of accumulated depth and opacity toward intermediate values, which in turn affects the softmax-based competition among subsequent Gaussians along the ray. Addressing these limitations is an important direction for future work.

Algorithm 1 The Softmax-GS rendering algorithm. For simplicity, we ignore invalid Gaussians and background color, and assume there are K Gaussians in total.

Require: $T_{\text{past}} = 1, c_{\text{past}} = 0$ ▷ Transmittance and pixel color so far
Require: $d_{\text{past}} = 0, p_{\text{past}} = 0$ ▷ Moving average of past depth and Gaussian exponent
Require: $K, \mathbf{x}_{\text{pixel}}$ ▷ Total number of Gaussians along the ray, pixel coordinate
Require: $\mathbf{x}[K], \sigma[K], o[K], c[K], d[K]$ ▷ Gaussian splat center, conics, opacity, color, depth
Require: $\alpha[K], \beta[K], \gamma[K]$ ▷ Softmax-GS parameters

$k \leftarrow 1$
while $k \leq K$ **do**
  $\mathbf{x}' = \mathbf{x}[k] - \mathbf{x}_{\text{pixel}}$
  $p_{\text{cur}} \leftarrow -0.5 \cdot \text{Mahalanobis\_distance}(\mathbf{x}', \sigma[k])$
  $a_{\text{cur}} \leftarrow o[k] \cdot \exp(-(-p_{\text{cur}})^{\alpha[k]})$ ▷ Control boundary sharpness using GEF
  if $T_{\text{past}} < 1$ **then**
    $T_{\text{orig}} \leftarrow T_{\text{past}} \cdot (1 - a_{\text{cur}})$ ▷ Record the original transmittance
    $a_{\text{past}} \leftarrow 1 - T_{\text{past}}$ ▷ Past absorbance
    $w_{\text{cur}} \leftarrow 1/(1 + \exp(\beta[k] \cdot (p_{\text{past}} - p_{\text{cur}}))$ ▷ Softmax competition between the Gaussians
    $\hat{a}_{\text{cur}} \leftarrow w_{\text{cur}} \cdot a_{\text{cur}}$ ▷ Softmax-weighted current absorbance
    $w_{\text{past}} \leftarrow 1 - w_{\text{cur}}$
    $\hat{a}_{\text{past}} \leftarrow w_{\text{past}} \cdot a_{\text{past}}$ ▷ Softmax-weighted past absorbance
    $\hat{a}_{\text{past}} \leftarrow \frac{\hat{a}_{\text{past}}(1-T_{\text{orig}})}{\hat{a}_{\text{past}}+\hat{a}_{\text{cur}}}$ ▷ Order invariance and transmittance maintenance
    $\hat{a}_{\text{cur}} = \frac{\hat{a}_{\text{cur}}(1-T_{\text{orig}})}{\hat{a}_{\text{cur}}+\hat{a}_{\text{past}}T_{\text{orig}}}$
    $s \leftarrow \exp(-\gamma[k]|d[k] - d_{\text{past}}|)$ ▷ Decay influence of softmax with distance
    $\bar{a}_{\text{past}} \leftarrow s \cdot \hat{a}_{\text{past}} + (1 - s) \cdot a_{\text{past}}$ ▷ Interpolate original and softmax-ed absorbance
    $a_{\text{cur}} \leftarrow s \cdot \hat{a}_{\text{cur}} + (1 - s) \cdot a_{\text{cur}}$
    $m \leftarrow \frac{a_{\text{cur}}+\bar{a}_{\text{past}}-\sqrt{(a_{\text{cur}}+\bar{a}_{\text{past}})^2-4(1-T_{\text{orig}})a_{\text{cur}}\bar{a}_{\text{past}}}}{2a_{\text{cur}}\bar{a}_{\text{past}}}$ ▷ Transmittance maintenance again
    $\bar{a}_{\text{past}} \leftarrow m \cdot \bar{a}_{\text{past}}$
    $a_{\text{cur}} \leftarrow m \cdot a_{\text{cur}}$
    $T_{\text{past}} \leftarrow 1 - \bar{a}_{\text{past}}$ ▷ Update past transmittance
    $c_{\text{past}} \leftarrow c_{\text{past}} \cdot \bar{a}_{\text{past}}/a_{\text{past}}$ ▷ Update past color
  end if
  $c_{\text{past}} \leftarrow c_{\text{past}} + c[k] \cdot a_{\text{cur}} \cdot T_{\text{past}}$
  $d_{\text{past}} \leftarrow \frac{d_{\text{past}}\cdot(1-T_{\text{past}})+d[k]\cdot a_{\text{cur}}\cdot T_{\text{past}}}{1-T_{\text{past}}+a_{\text{cur}}\cdot T_{\text{past}}}$ ▷ Average depth weighted by absorbance
  $p_{\text{past}} \leftarrow \frac{p_{\text{past}}\cdot(1-T_{\text{past}})+p_{\text{cur}}\cdot a_{\text{cur}}\cdot T_{\text{past}}}{1-T_{\text{past}}+a_{\text{cur}}\cdot T_{\text{past}}}$ ▷ Average Gaussian exponet weighted by absorbance
  $T_{\text{past}} \leftarrow T_{\text{past}} \cdot (1 - a_{\text{cur}})$
  $k \leftarrow k + 1$
end while