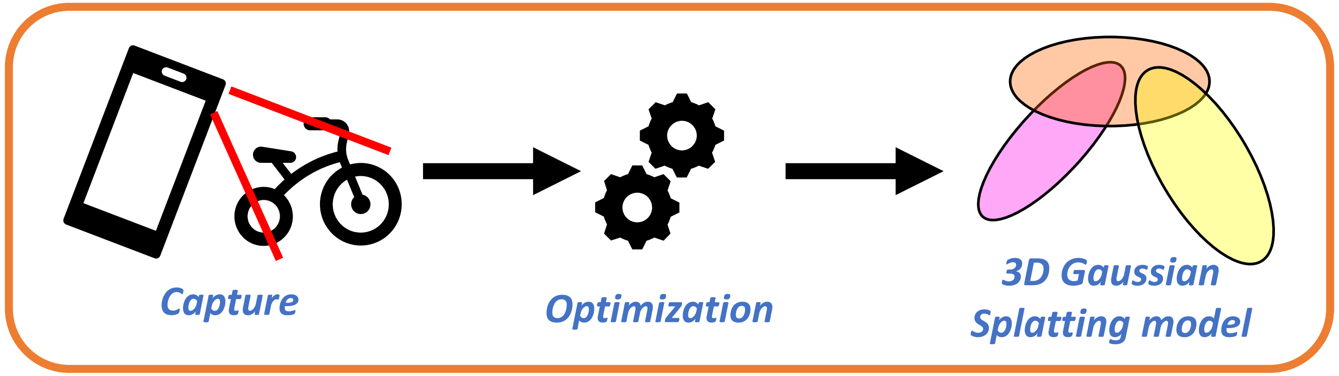
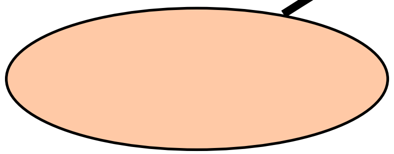
CAFe-GS: Compactness-Aware Frequency-Guided Densification for 3D Gaussian Splatting

HUAR Léo-Paul [1,2], SANDRI Gustavo [2], SABATER Neus [2], GUILLEMOT Christine [1], HELLIER Pierre [1]

[1] *Inria, University of Rennes, Rennes, France*
[2] *InterDigital, Video Lab, Cesson-Sévigné, France*

Novel View Synthesis - 3D Gaussian Splatting (3DGS)





A single 3D Gaussian primitive is represented with a lots of attributes

Position
Covariance
Opacity
Colour

↓

59 attributes

Typical count
≈ 2–3M primitives

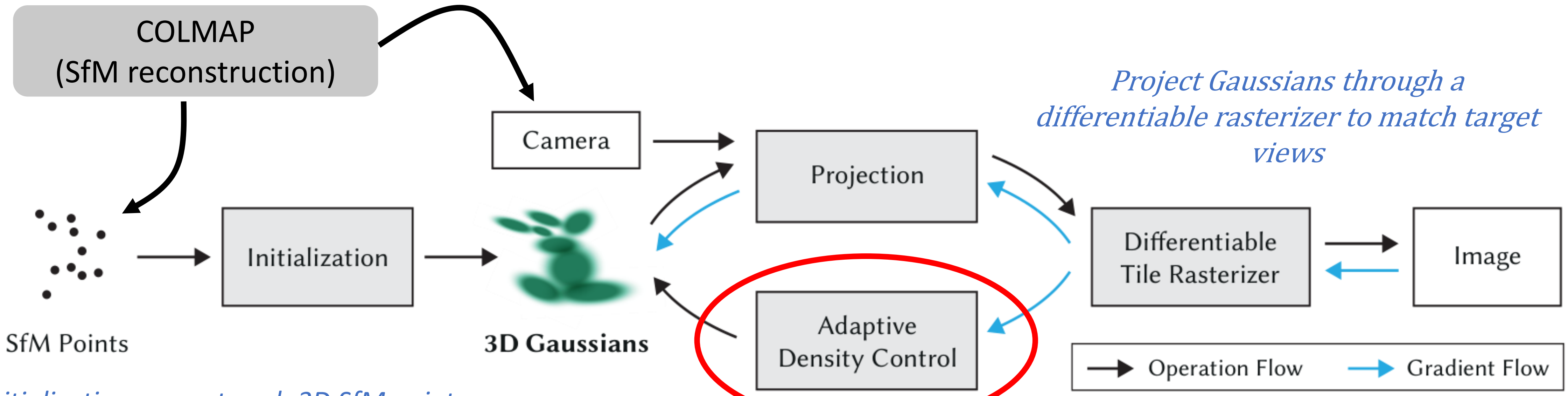
↓

Large storage footprint
≈ 700 MB / model

3D Gaussian Splatting example with the Bicycle scene (Mip-Nerf 360)

Optimization of 3D Gaussian Splatting

Input images provide initial 3D Structure from Motion (SfM) point cloud and Cameras positions and parameters



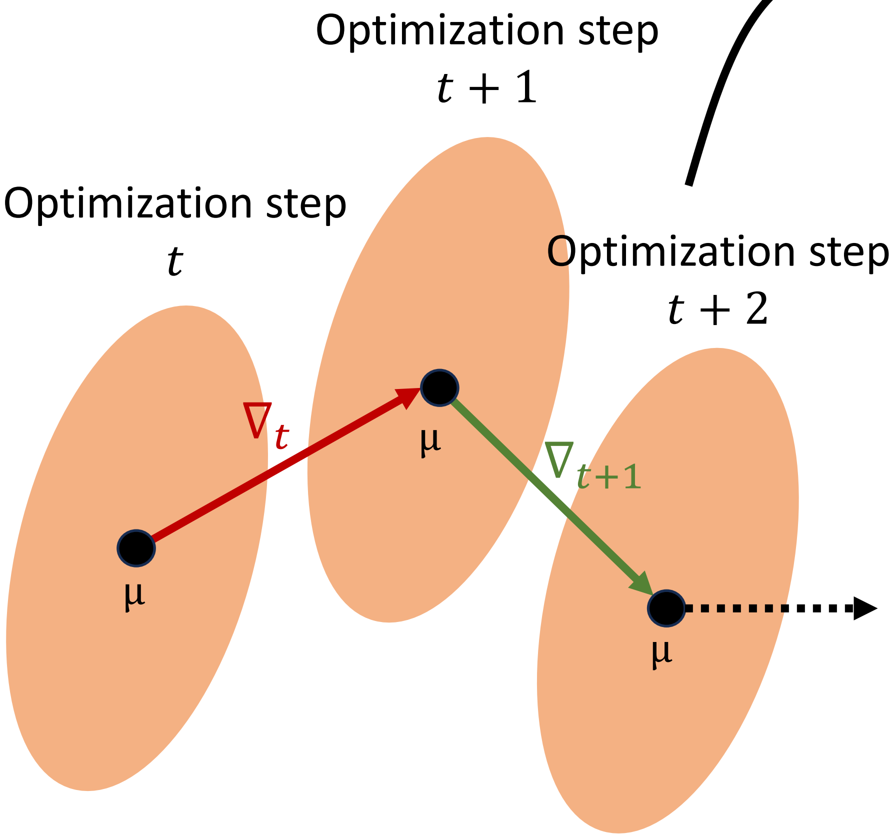
Project Gaussians through a differentiable rasterizer to match target views

Initialization convert each 3D SfM point into a Gaussian primitive $(\mu, \Sigma, sh_i, \alpha)$

Periodically clone or split Gaussians to refine geometry.

Update Gaussian parameters by minimizing photometric loss (∇ back-propagates through rendering)

Gradient-based Densification



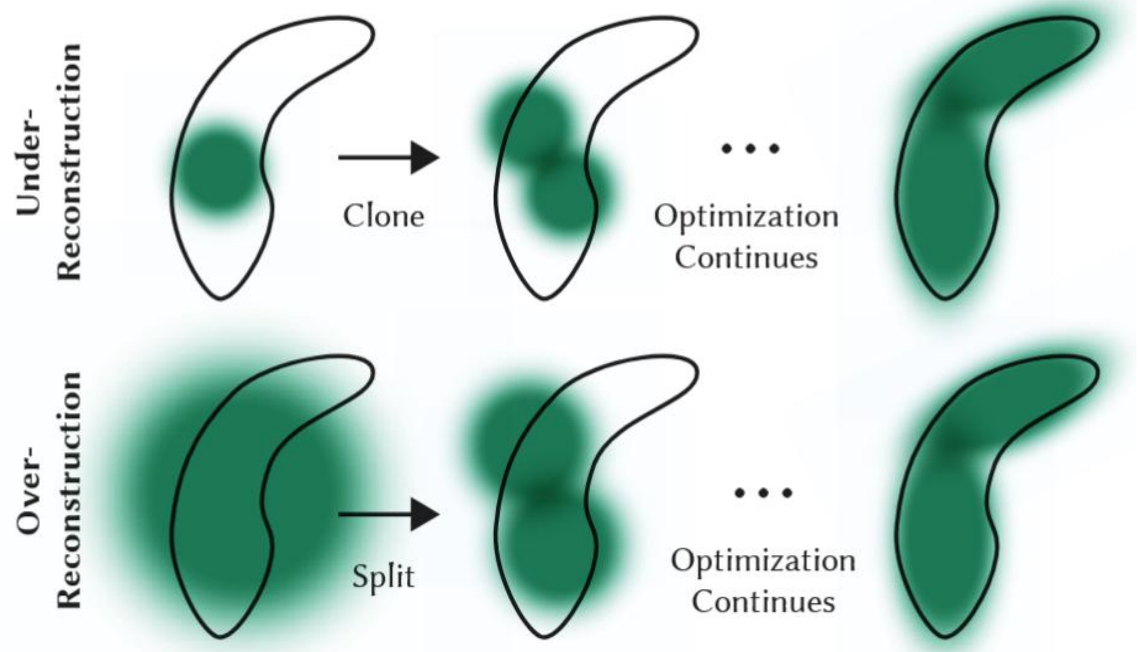
Track positional gradients during optimization

$$m\nabla = \frac{\nabla_t + \nabla_{t+1} + \cdots + \nabla_{t+T}}{T}$$

Average gradients between two ADC steps

If $m\nabla > \tau$

Densify if the average gradient exceeds a fixed threshold



Limitation of Gradient-based Densification

Over-densification

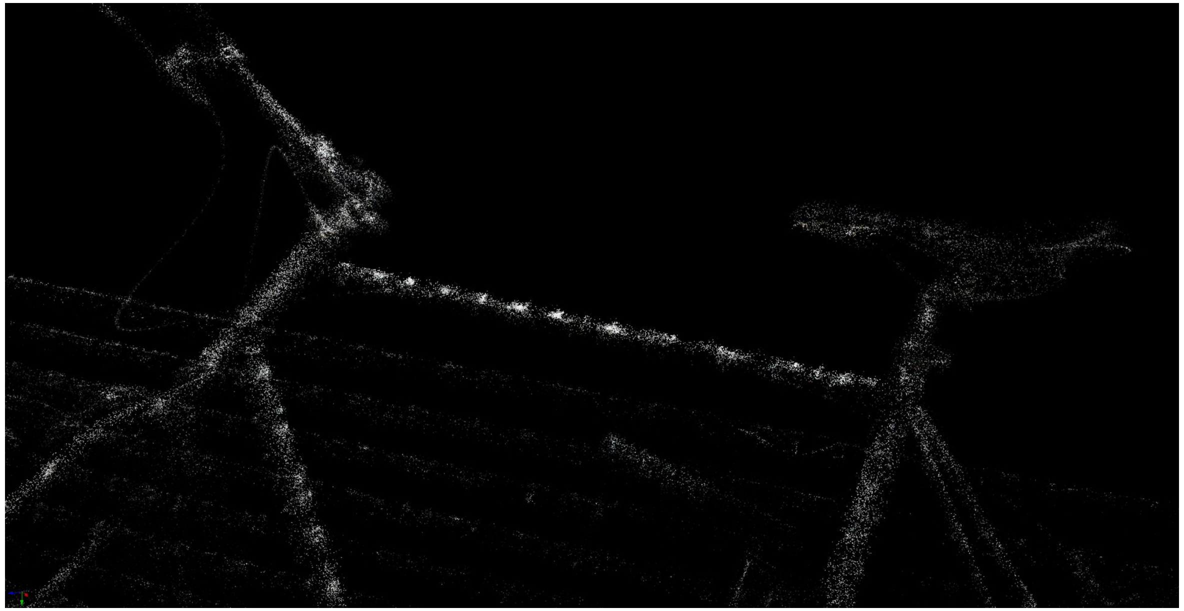
Large positional gradient persist and cause **redundant duplication** even in already well-reconstructed regions

High model size

This redundancy lead to **increase memory** and rendering cost

Leak perceptual control

The gradient threshold $s\nabla$ has **no visual meaning**, making it hard to balance compactness and quality.



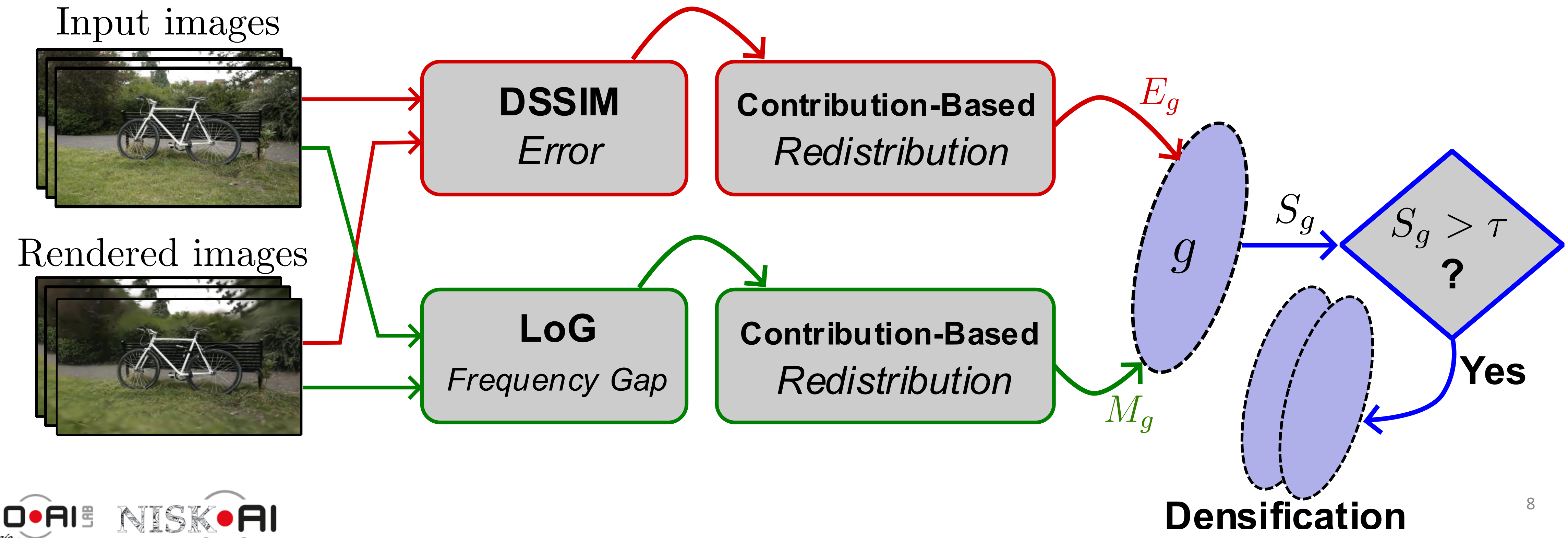
Compactness-Aware Frequency-Guided Densification (CAFe-GS)

Prioritize ***fewer* Gaussians** while preserving reconstruction quality. Producing lighter and faster model

Replace the positional gradient criterion with a per-gaussian **reconstruction error E_g**

Modulate the criterion using per-gaussian **local-frequency cues M_g** to **concentrate densification in detail-rich and underfitted regions**

A **single parameter τ** serves as a *knob* to **control balance between quality and model size**



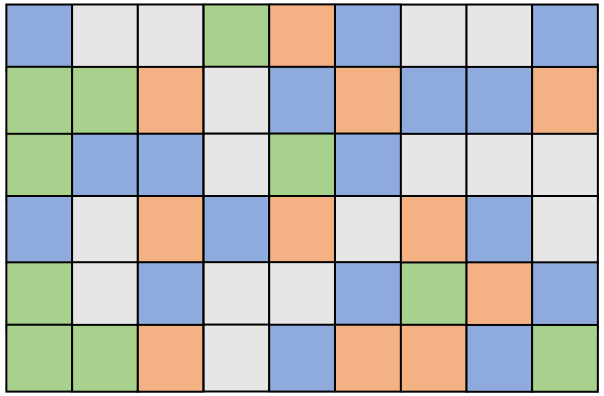
Per-Gaussian Reconstruction Error

We compute the visual error in image space, assign it back to the Gaussians responsible for it, and keep the largest error observed across views

Ground Truth

Render

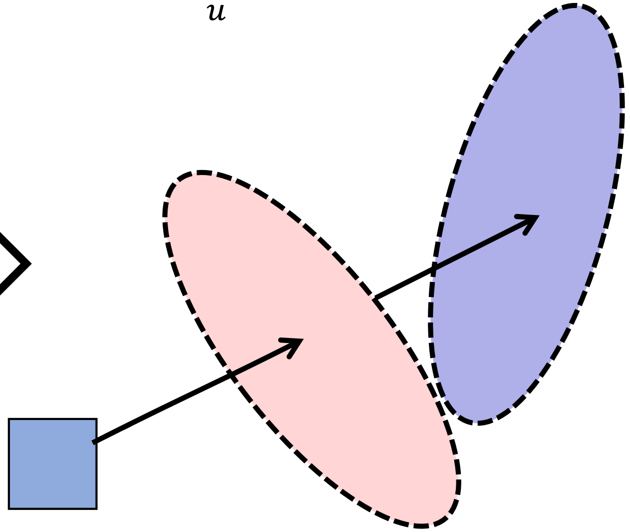
Per-pixel error map DSSIM



$$d_v(u) = 1 - SSIM(I_v, \widehat{I_v})(u)$$

Redistribute pixel error according to each Gaussian contribution

$$e_{g,v} = \sum_u C(g,u) * d_v(u)$$



$$C(g,u) = \alpha_g(u) \prod_{h \prec g} (1 - \alpha_h(u))$$

Pixel influence of each Gaussian, weighted by opacity and visibility

We retain the maximum error across views to preserve localized reconstruction errors

$$E_g = \max_v e_{g,v}$$

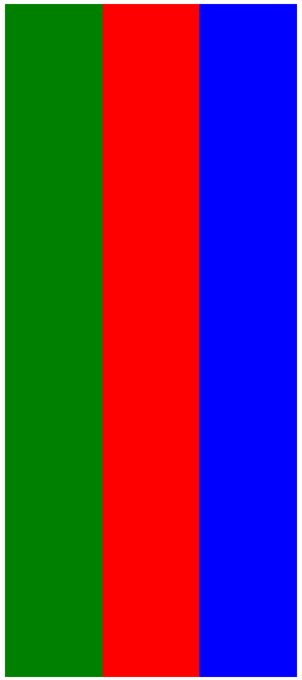
Frequency Guidance Score — Edge Sensitivity

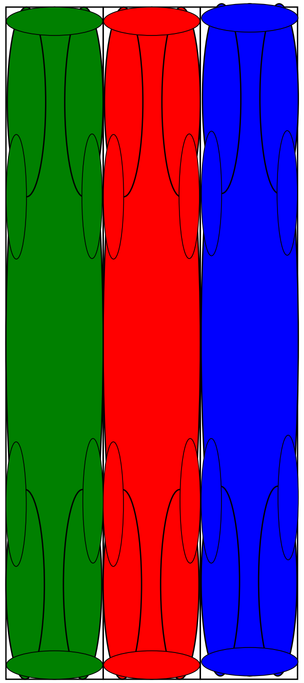
- **High-frequency regions** (edges, thin structures) indicate where adding new Gaussians most improves reconstruction fidelity.

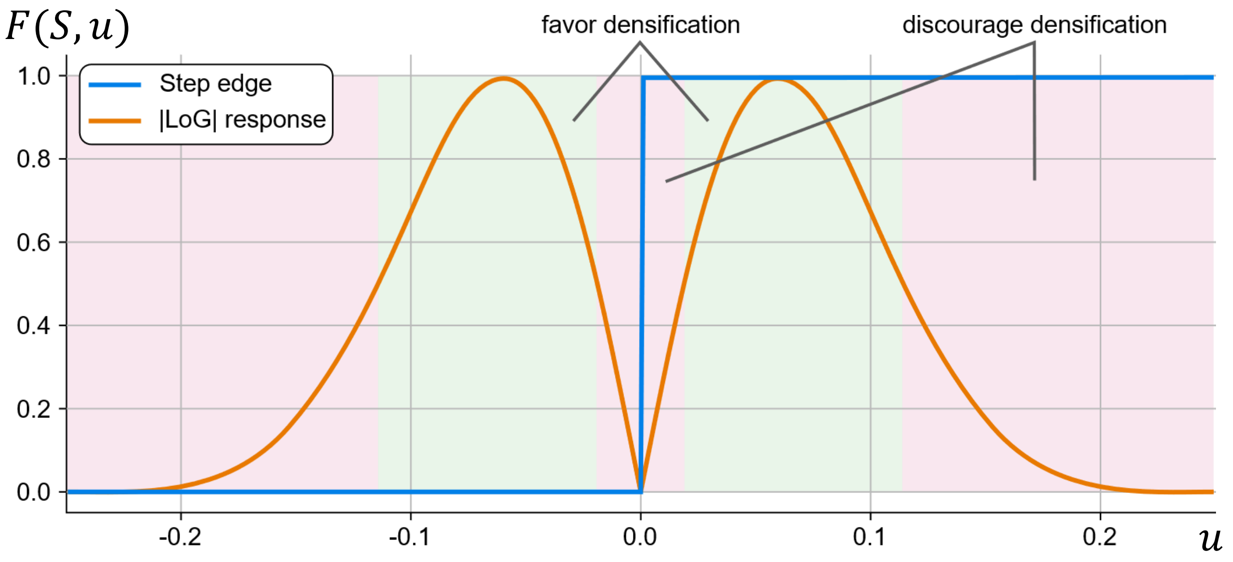
- We extract this signal using the **Laplacian-of-Gaussian (LoG)** filter:

$$F(I_v, u) = \ |(LoG_\sigma * I_v)(u)|$$

- The LoG response produces **symmetric peaks** around intensity edges, marking where to *favor densification* versus *suppress it*.







Frequency Guidance Score — Frequency Gap Modulation

- To identify **underfitted regions**, we compute the **frequency gap** between the input and rendered images:

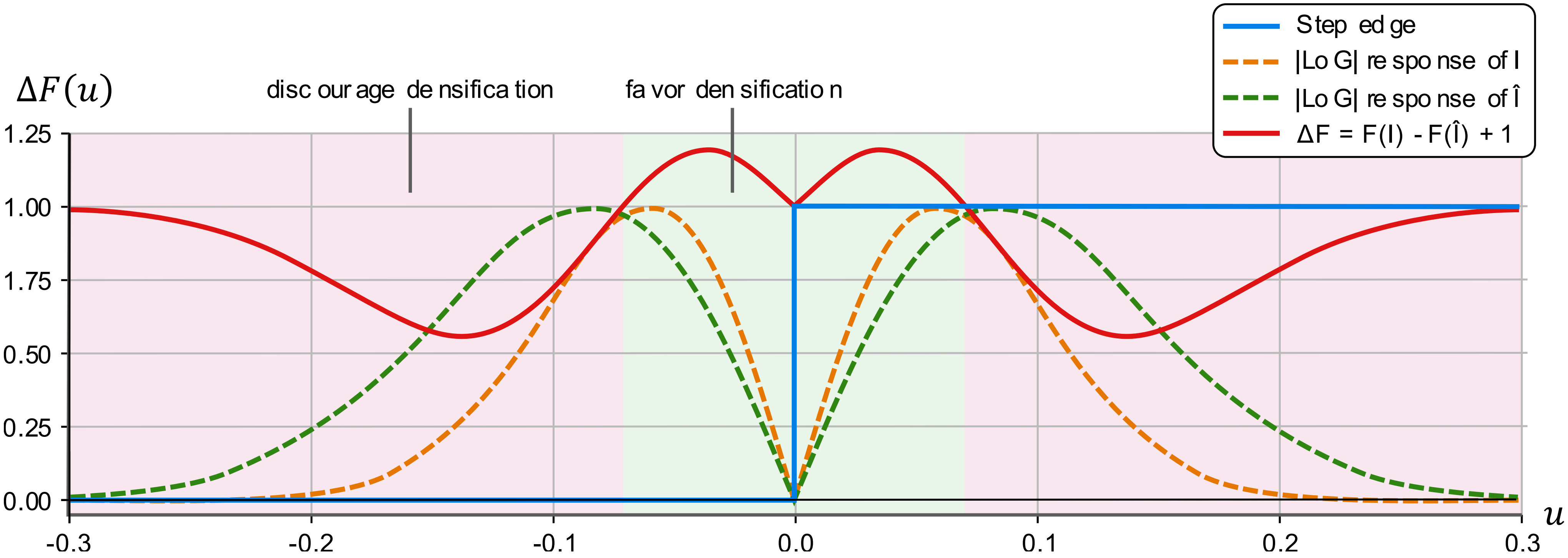
$$\Delta F(u) = F(I_v, u) - F\left(\widehat{I_v}, u\right) + 1$$

*Positive gaps highlight areas where the input contains **richer detail** than the render (missing structure)*

- This gap is aggregated per Gaussian, weighted by its opacity contribution $C(g,u)$

$$M_g^{\Delta} = \max_v \sum_{u \in \Omega_v} C(g,u) * \Delta F_v(u)$$

*M_g^{Δ} emphasizes **detail mismatch** while suppressing regions already well reconstructed.*

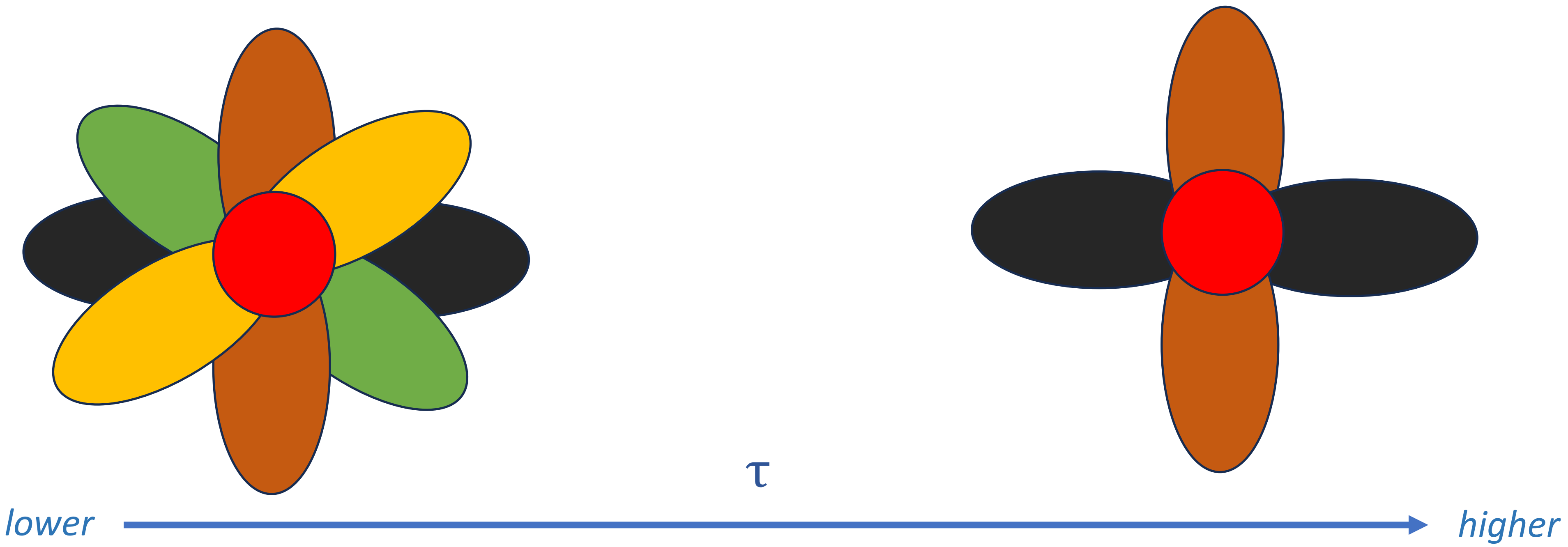


Final Densification Criterion

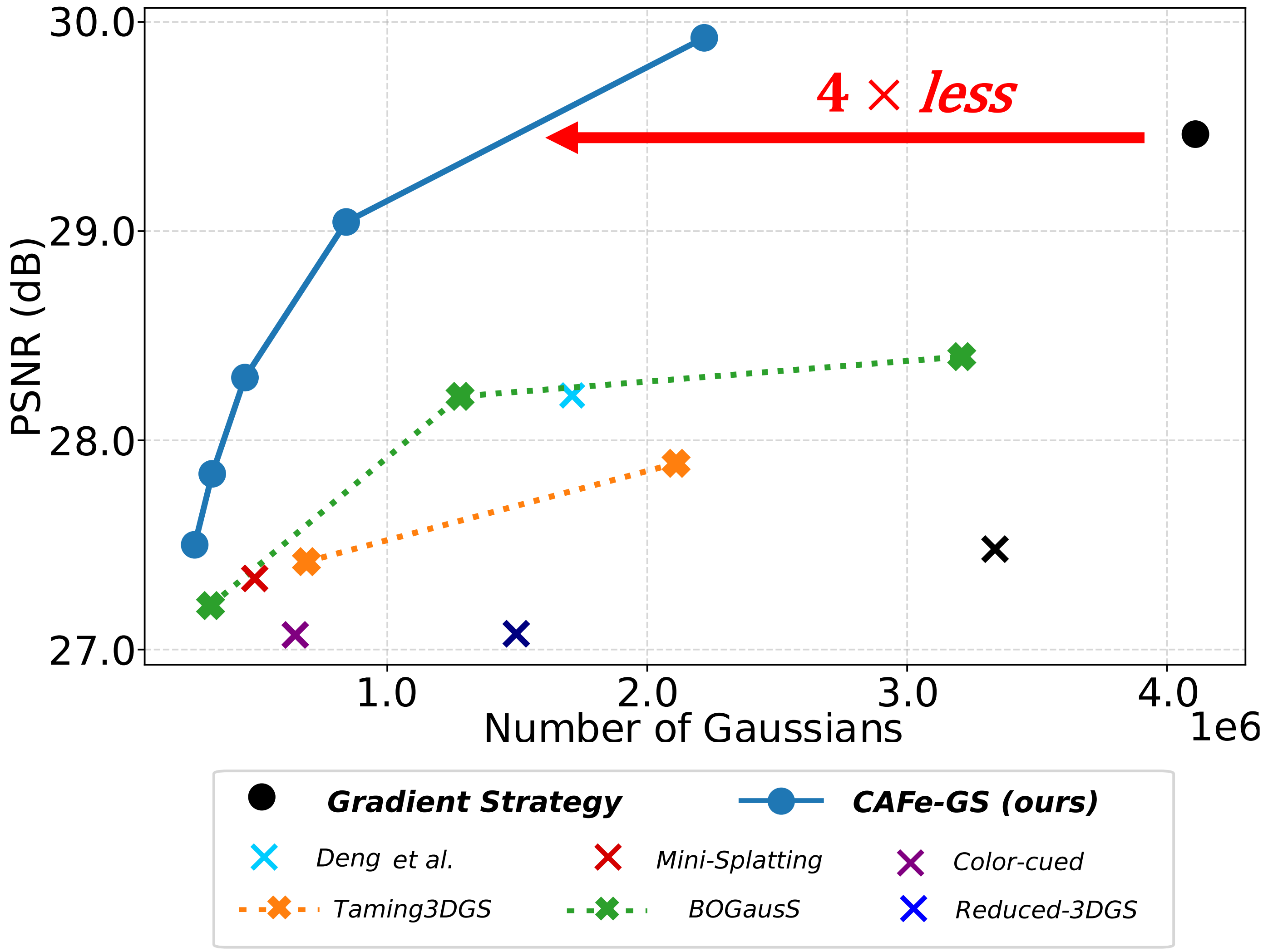
- The **gradient-based rule** is replaced with a **perceptual–frequency-guided score**:
$$S_g = E_g * M_g^{\Delta}$$
- At each **Adaptive Densification Control (ADC)** step, every Gaussian satisfying
$$S_g \geq \tau$$

becomes a candidate for duplication (*clone or split*).

- The threshold τ serves as a **compactness knob**:
  - **Low** τ → more Gaussians → higher fidelity.
  - **High** τ → fewer Gaussians → lighter, more compact models.

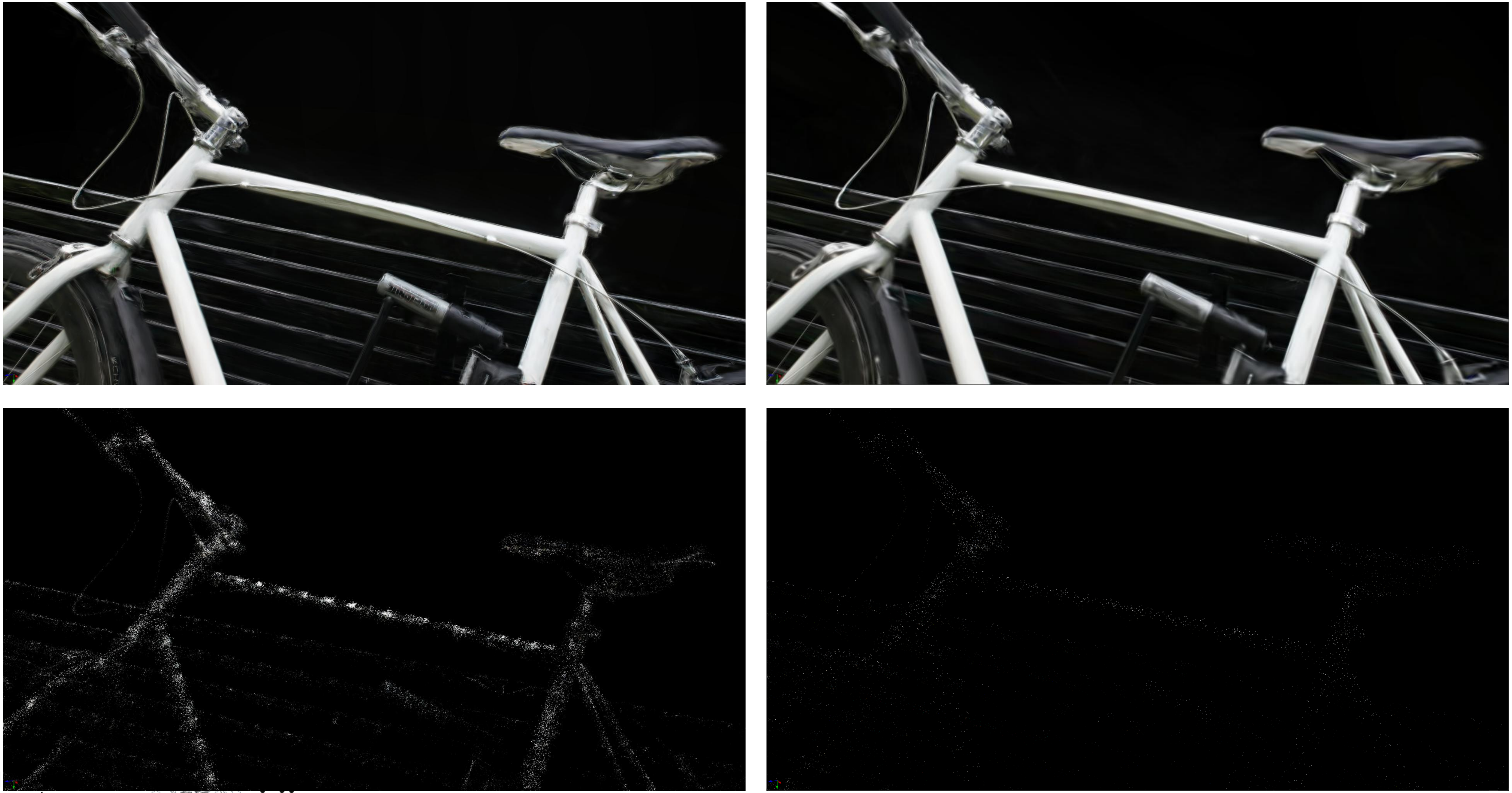


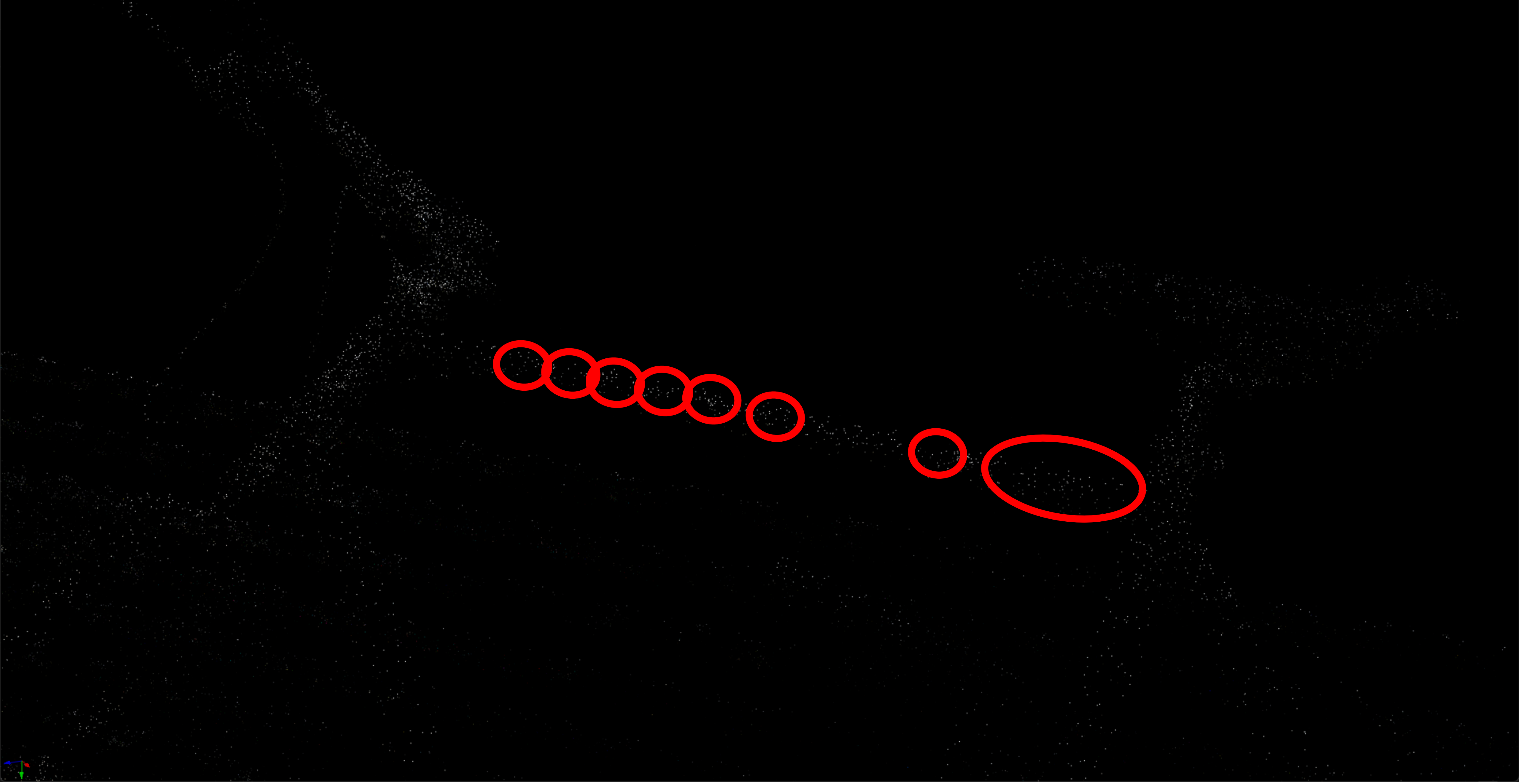
Quantitative Results — MipNeRF-360 Benchmark



- Evaluated for five compactness thresholds $\tau \in \{10, 25, 50, 75, 100\}$
- Across datasets (*MipNeRF-360*, *Tanks & Temples*, *Deep Blending*), **CAFe-GS matches baseline quality** while using ≈ 4× fewer Gaussians.
- At the most compact setting, CAFe-GS achieves up to **15× model size reduction** with negligible quality loss.

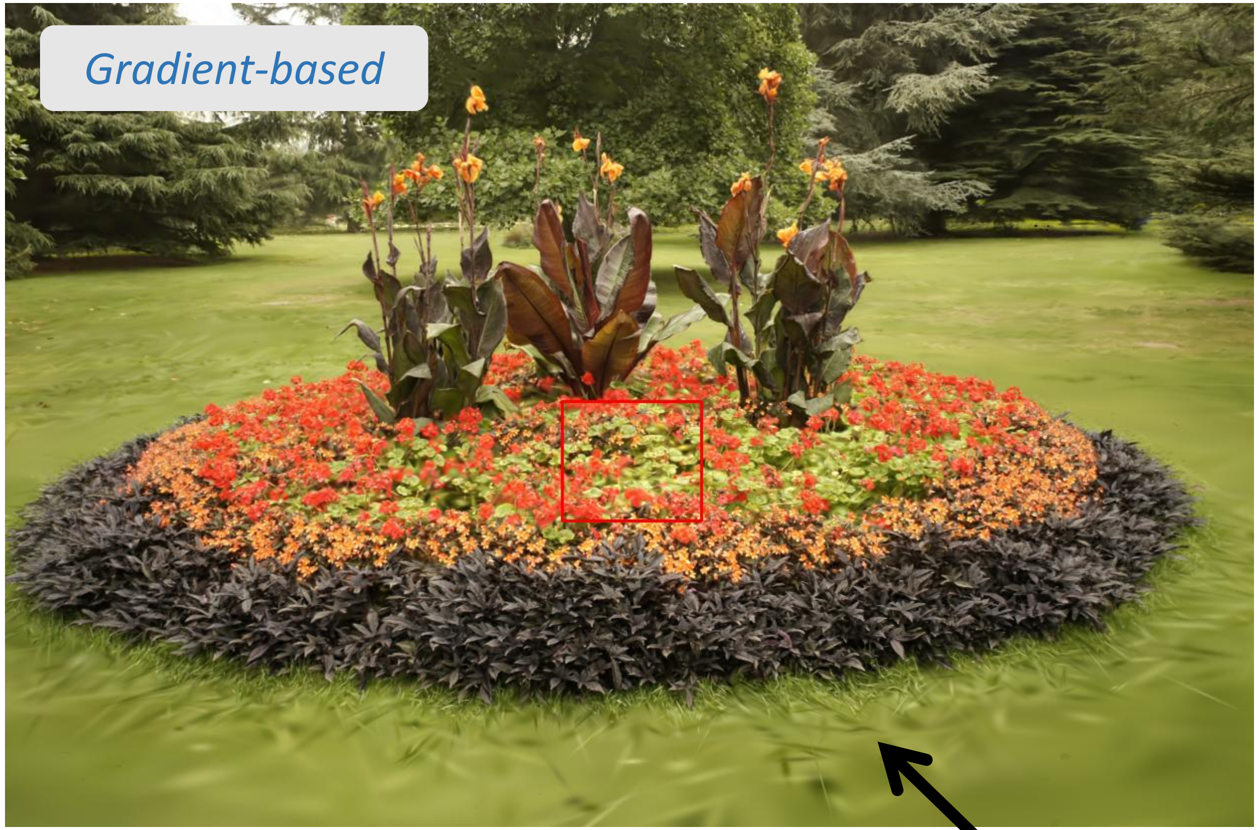
Gradient-based *CAFe-GS*





Qualitative results

By encouraging densification in **high-frequency regions** such as grass and fine vegetation, CAFe-GS **preserves sharper details** than gradient-based densification


Gradient-based


CAFe-GS (ours)

Conclusion

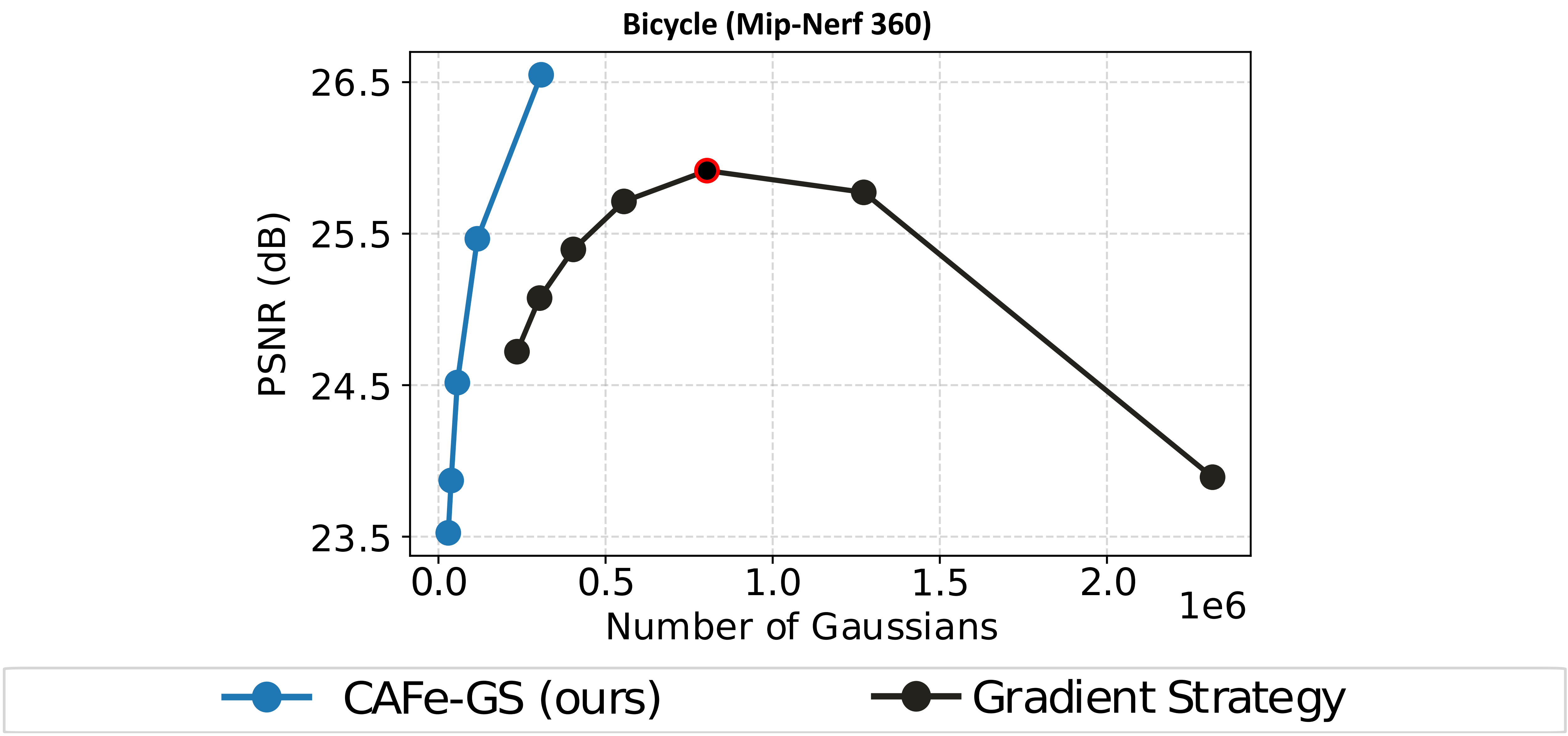
Prefer Quality
2-4x compactness
Same quality

3DGS

Prefer Compactness
12-15x compactness
2-2.5 dB quality loss

CAFe-GS
(Best quality)

CAFe-GS
(Compactness-first)



Varying the densification threshold reveals **unstable behavior** for the gradient baseline