Neural Graphics III

Multimodal Generative AI Theories and Applications
Lecture 11

Jin-Hwa Kim and Sangdoo Yun

NAVER AI LAB

Table of contents

3D Gaussian Splatting for Real-Time Radiance Field Rendering

Kerbl et al., ACM Trans. Graph. 2023; Demo by @AlexR4 https://x.com/AlexR4/status/1720378611289526471

3D Gaussians

- 3D Gaussian splatting (Kerbl et al., 2023) represents a scene with learnable 3D Gaussian primitives, where each primitive consists of mean $\mu \in \mathbb{R}^3$, covariance $\Sigma \in \mathbb{R}^{3\times3}$, opacity $\alpha \in [0, 1]$, and view-dependent color c in *spherical harmonics*.
- The covariance matrix is decomposed into $\Sigma = \mathrm{RSS}^\intercal\mathrm{R}^\intercal$, where R is a rotation matrix, parameterized by a quaternion $\in \mathbb{R}^4$, and $\mathrm{S} = \mathrm{diag}(s_0, s_1, s_2)$ is a scale parameter.
- There are several advantages of quaternions…

The 3DGS summary is excerpted from Hyung et al. (2024).

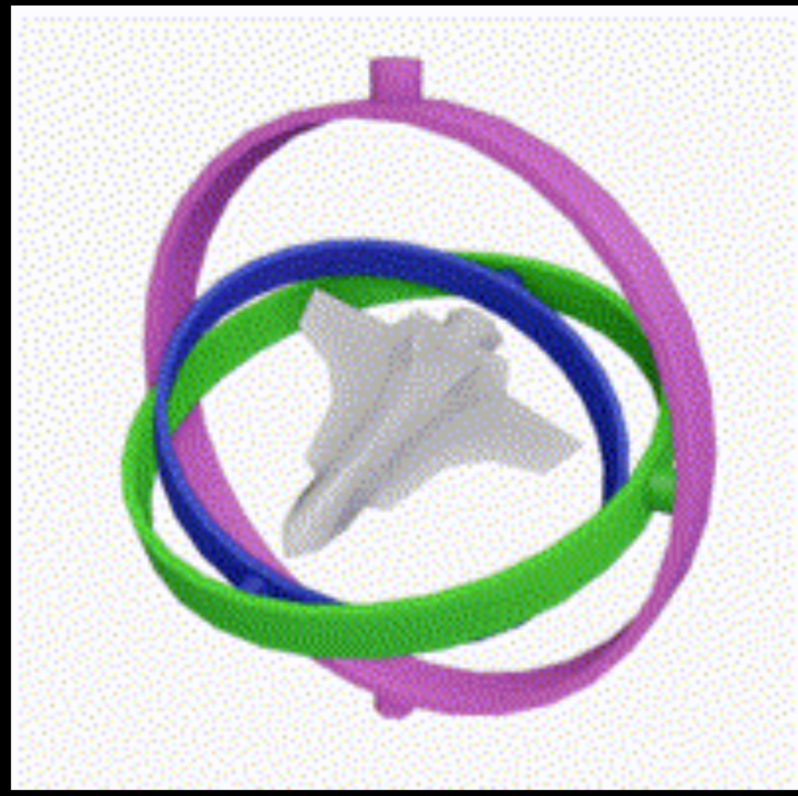
*Advantages of quaternion**

- Avoiding *gimbal lock*, a problem associated with systems that use Euler angles.
- Faster and more compact than matrices (*i.e.*, $\mathbb{R}^4$ instead of $\mathbb{R}^{3\times3}$ for 3D)
- Nonsingular representation (compared with Euler angles, for example).
- Pairs of unit quaternions can represent a 4D rotation (*ref.* 4DGS in Yang et al., 2024).

*Quaternions were first described by the Irish mathematician William Rowan Hamilton (1805-1865) in 1843.

Gimbal lock

- *Gimbal lock* is the loss of one degree of freedom (DoF) in a system due to the alignment of two rotation axes.
- The suspended object cannot rotate around one axis, which can cause control or orientation errors.
- Despite the name, no physical gimbal is "locked"—they all still rotate. But the system's effective rotation space becomes 2D instead of 3D.



Gimbal-locked airplane. When the pitch (green) and yaw (magenta) gimbals become aligned, changes to roll (blue) and yaw (magenta) apply the same rotation to the airplane.
"A well-known gimbal lock incident happened in the Apollo 11 Moon mission."

https://en.wikipedia.org/wiki/Gimbal_lock

Quaternion to rotation matrix

- Given a *unit quaternion* $\mathrm{q} = (w, x, y, z)$, or $\mathrm{q} = w + ix, + jy + kz$, you can find a 3D rotation matrix as follows:

$$R(\mathrm{q}) = \begin{bmatrix} 1 - 2(y^2 + z^2) & 2(xy - wz) & 2(xz + wy) \\ 2(xy + wz) & 1 - 2(x^2 + z^2) & 2(yz - wx) \\ 2(xz - wy) & 2(yz + wx) & 1 - 2(x^2 + y^2) \end{bmatrix}.$$

- Or, more elegantly with $\mathrm{v} = [x, y, z]^\top$ and a skew-symmetric matrix $[\mathrm{v}]_\times$,

$$R(\mathrm{q}) = (w^2 - \|\mathrm{v}\|^2) \cdot \mathrm{I} + 2\mathrm{v}\mathrm{v}^\top + 2w[\mathrm{v}]_\times \quad \text{where} \quad [\mathrm{v}]_\times = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}.$$

https://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Euler_angles

Rotation matrix to quaternion

- Given a 3D rotation defined by $\theta_x, \theta_y,$ and $\theta_z,$ you can find a quaternion that is $\mathrm{q} = q_z \cdot q_y \cdot q_x$ (quaternion multiplication or Hamilton multiplication) where

$$q_x = \left(\cos\frac{\theta_x}{2},\ \sin\frac{\theta_x}{2},\ 0,\ 0\right),$$

$$q_y = \left(\cos\frac{\theta_y}{2},\ 0,\ \sin\frac{\theta_y}{2},\ 0\right),$$

$$q_z = \left(\cos\frac{\theta_z}{2},\ 0,\ 0,\ \sin\frac{\theta_z}{2}\right).$$

https://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Euler_angles

Quaternion multiplication

- Given two quaternions $\mathrm{q}_1 = (w_1, x_1, y_1, z_1)$ and $\mathrm{q} = (w_2, x_2, y_2, z_2)$, the quaternion (Hamilton) multiplication $\mathrm{q} = \mathrm{q}_1 \cdot \mathrm{q}_2$ is defined as:

$$
\begin{aligned}
w &= w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2, \\
x &= w_1 x_2 + x_1 w_2 + y_1 z_2 - z_1 y_2, \\
y &= w_1 y_2 - x_1 z_2 + y_1 w_2 + z_1 x_2, \\
z &= w_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 w_2 .
\end{aligned}
$$

This yields a new quaternion $\mathrm{q} = (w, x, y, z)$.

Relation to Rodrigues' rotation formula

- We represent a quaternion q using a rotation axis u and an angle θ:

$$\mathrm{q} = (w, \mathrm{v}) = \left(\cos\frac{\theta}{2}, \mathrm{u}\sin\frac{\theta}{2}\right).$$

- Natural extension from 2D rotation in Euler's formula:

$$e^{i\theta} = \cos\theta + i\sin\theta.$$

- Now, we substitute w and v in

$$R(\mathrm{q}) = (w^2 - \|\mathrm{v}\|^2)\cdot \mathrm{I} + 2\mathrm{v}\mathrm{v}^\top + 2w[\mathrm{v}]_\times.$$

Relation to Rodrigues' rotation formula

- Now, we substitute w and v in

$$R(\mathrm{q}) = (w^2 - \|\mathrm{v}\|^2) \cdot \mathrm{I} + 2\mathrm{v}\mathrm{v}^\top + 2w[\mathrm{v}]_\times.$$

- We can rearrange as follows:

$$\begin{aligned} R(\mathrm{q}) &= (w^2 - \|\mathrm{v}\|^2)\mathrm{I} + 2\mathrm{v}\mathrm{v}^\top + 2w[\mathrm{v}]_\times \\ &= \cos\theta\,\mathrm{I} + (1 - \cos\theta)\,\mathrm{u}\mathrm{u}^\top + \sin\theta\,[\mathrm{u}]_\times \\ &= \mathrm{I} + \sin\theta\,[\mathrm{u}]_\times + (1 - \cos\theta)\,[\mathrm{u}]_\times^2 \end{aligned}$$

using the fact that $[\mathrm{u}]_\times^2 = \mathrm{u}\mathrm{u}^\top - \mathrm{I}$ and u is a unit vector.

3D Gaussian splatting

- The primitives are rasterized to an image via differentiable volume splatting for a faster rendering capability than previous works.
- 3D Gaussian is projected to a 2D space using $\Sigma' = \mathrm{J}\mathrm{W}\Sigma\mathrm{W}^\intercal\mathrm{J}^\intercal$, where W is a world-to-camera transform and J is the Jacobian of the *affine approximation* of the projection matrix (Zwicker et al., 2001), which we define the 2D Gaussian as $\mathscr{G}$.
- Then, the alpha-blended Gaussians in the *order of depth* for k as follows:

$$\mathrm{c}(\mathrm{u}) = \sum_{k=1}^{K} c_k \alpha_k \mathscr{G}_k(\mathrm{u}) \prod_{j=1}^{k-1} \left(1 - \alpha_j \mathscr{G}_j(\mathrm{u})\right)$$

where u is an image coordinate. The rendered images are supervised with the photometric loss similarly to NeRFs.

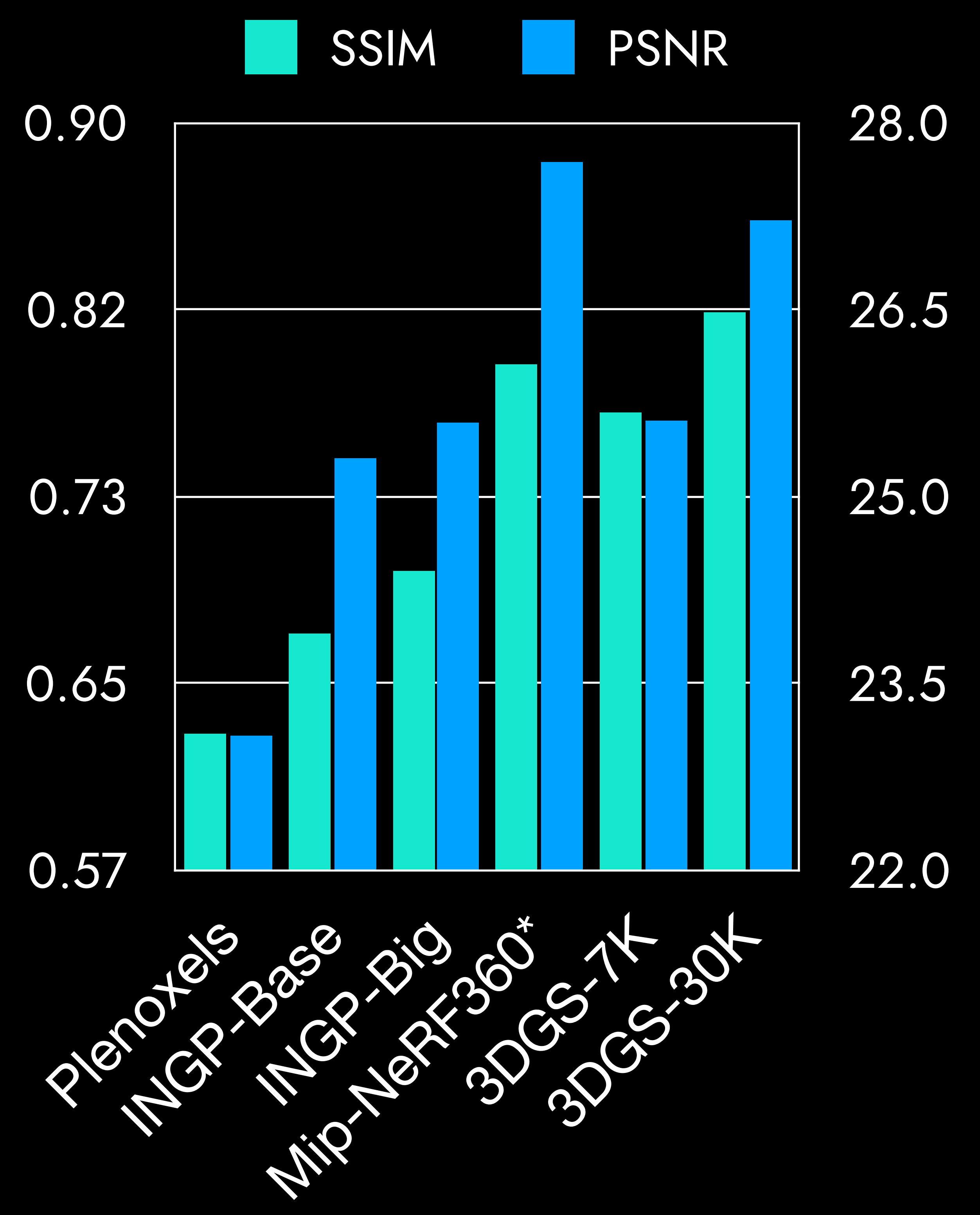
Adaptive density control

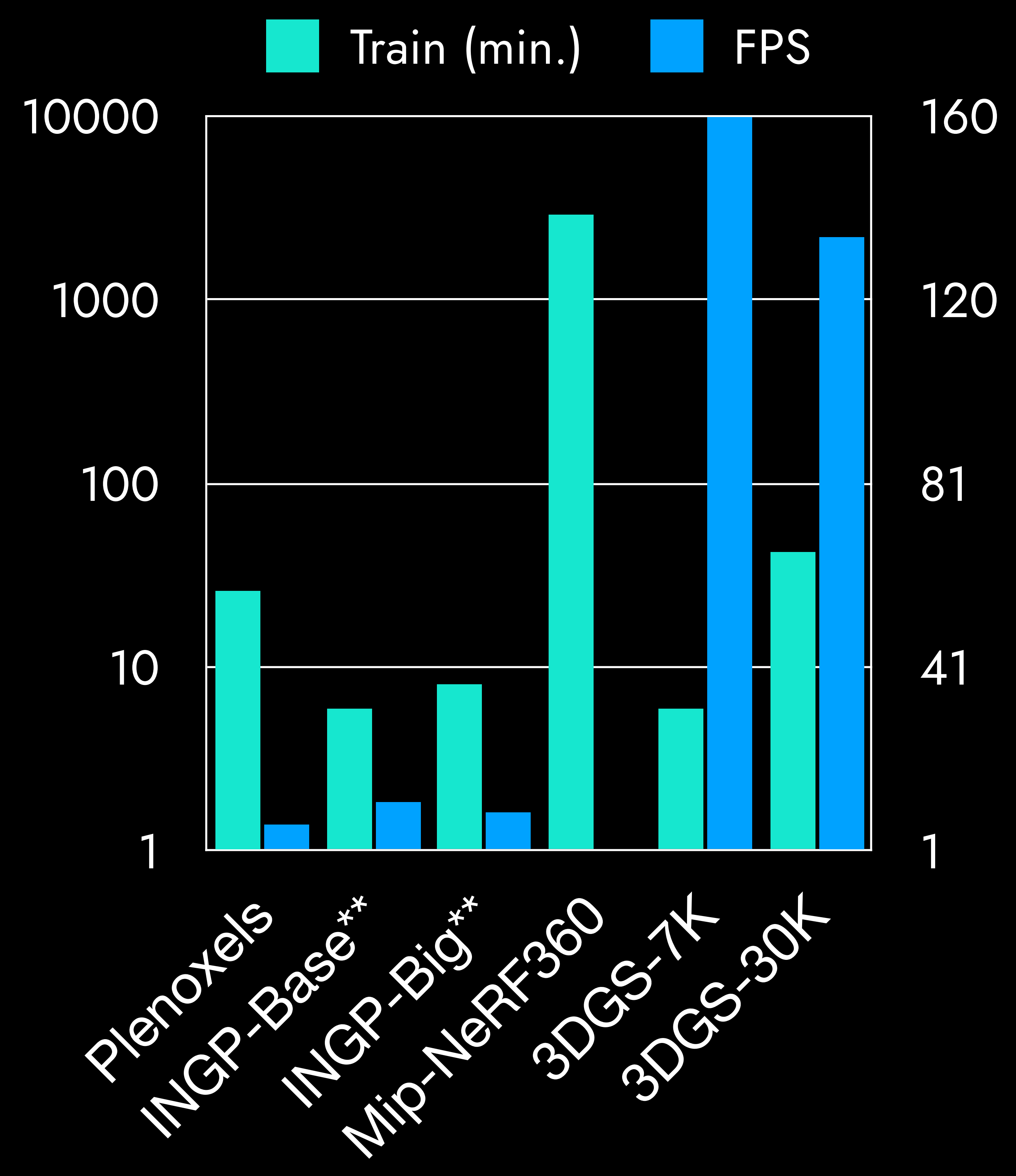
- The 3D Gaussians are initialized by sparse SfM points (Schönberger et al., 2016), although a random start shows reasonable performance.
- Adaptive density control (ADC) is designed for densification during optimization.
- ADC subsamples and splits Gaussians that satisfy the condition:

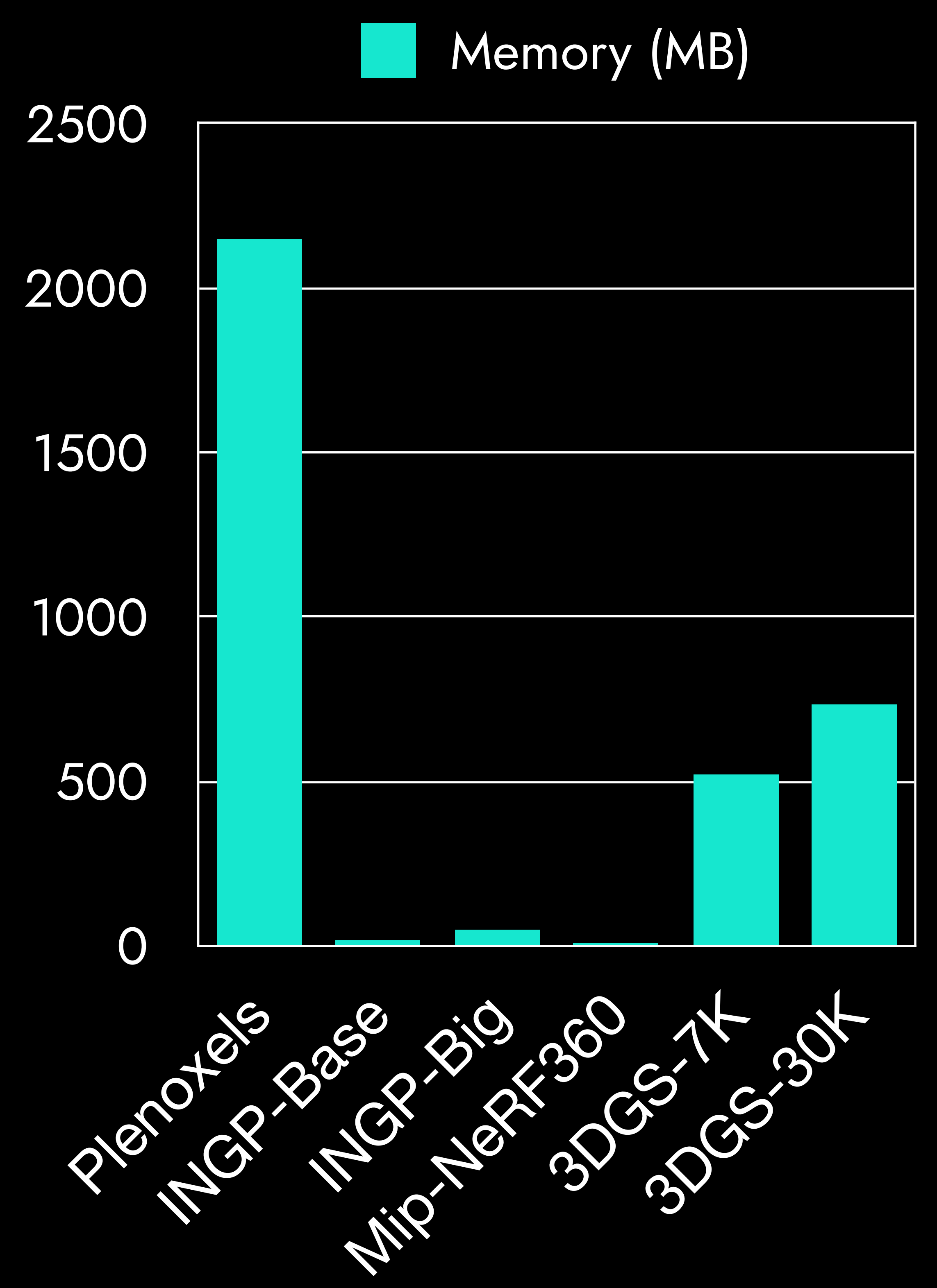
$$\left\| \frac{\partial L}{\partial \mathrm{u}} \right\|_2 = \left\| \sum_{i \in \mathscr{P}} \frac{\partial L}{\partial \mathrm{p}_i} \frac{\partial \mathrm{p}_i}{\partial \mathrm{u}} \right\|_2 > \tau$$

where $\mathscr{P}$ is a set of pixel indices in an image, p_i is the i-th pixel, and τ is a hyperparameter that controls ADC depending on input gradients.

Apple-to-apple comparisons







* The numbers are adopted from their work, for the Mip-NeRF360 dataset.
** Müller et al. (2022) reported 60 FPS on HD resolutions using their adaptive samplings.

NAVER AI LAB

Limitations

- Handcrafted heuristics for densification
  - Revisiting Densification in Gaussian Splatting (Bulo, 2024)
  - 3D Gaussian Splatting as Markov Chain Monte Carlo (Shakiba, 2024)

The limitations come from the slide of Kopanas (2024).

Limitations

- Handcrafted heuristics for densification
  - Revisiting Densification in Gaussian Splatting (Bulo, 2024)
  - 3D Gaussian Splatting as Markov Chain Monte Carlo (Shakiba, 2024)
- Popping artifacts because of the *mean-based sorting*
  - StopThePop: Sorted Gaussian Splatting (Radl et al., 2024)

The limitations come from the slide of Kopanas (2024).

Limitations

- Handcrafted heuristics for densification
  - Revisiting Densification in Gaussian Splatting (Bulo, 2024)
  - 3D Gaussian Splatting as Markov Chain Monte Carlo (Shakiba, 2024)
- Popping artifacts because of the *mean-based sorting*
  - StopThePop: Sorted Gaussian Splatting (Radl et al., 2024)
- Memory complexity
  - 3DGS: 350-700 MB for 3-6 M of Gaussians
  - INGP: 15-50MB, Mip-NeRF360: 8.6 MB
  - Reducing the Memory Footprint of 3D Gaussian Splatting (Papantonakis et al., 2024)

The limitations come from the slide of Kopanas (2024).

Effective Rank Analysis and Regularization for Enhanced 3D Gaussian Splatting

Junha Hyung* (KAIST), Susung Hong* (Korea Univ.), Sungwon Hwang (KAIST), Jaeseong Lee (KAIST),
Jaegul Choo† (KAIST), Jin-Hwa Kim† (NAVER AI Lab, SNU AIIS)

NeurIPS 2024

† Work done during an internship at NAVER AI Lab. ^ Co-corresponding authors.

Mesh extraction

- Mesh representation has enormous advantages when real-time rendering using *WebGL*, relighting, editing, compatibility, etc.
- Mesh extraction from 3DGS is challenging due to suboptimal *needle-like* artifacts and inaccurate normals caused by anisotropic Gaussians.
  - SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering (Guédon & Lepetit, 2023)
  - 2D Gaussian Splatting for Geometrically Accurate Radiance Fields (Huang et al., 2024)
  - Effective Rank Analysis and Regularization for Enhanced 3D Gaussian Splatting (Hyung et al., 2024)

Singular value distribution

- The singular value decomposition (SVD) obtains $A = \mathrm{U}\Sigma\mathrm{V}^\intercal$ where $\mathrm{U} \in \mathbb{R}^{M\times M}$, $\mathrm{V} \in \mathbb{R}^{N\times N}$, and $\Sigma \in \mathbb{R}^{M\times N}$. The truncated diagonal matrix Σ_r where $r = \min(M, N)$ contains the largest singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$.
- The *singular value distribution* is given by $q_i = \sigma_i / \|\sigma\|_1$ for $i = 1,2,\ldots, r$, where $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_r)^\intercal$ and $\|\cdot\|_1$ represents the l1-norm (sum of absolutes).
- This operation transforms the singular value distribution into a *probability distribution*.

Effective rank

Definition 1 (Effective rank)**.** The effective rank of the matrix A is concisely defined as the exponential of the Shannon entropy as follows:

$$\mathrm{erank}(\mathrm{A}) = \exp\big(H(q_1, q2, \ldots, q_r)\big), \quad \text{where} \quad H(q_1, q_2, \ldots, q_r) = -\sum_{i=1}^{r} q_i \log q_i.$$

- **Property 1.** It holds that $1 \leq \mathrm{erank}(\mathrm{A}) \leq \mathrm{rank}(\mathrm{A}) \leq r$
  - where the first inequality holds with equality if and only if $\sigma = (\|\sigma\|_1, 0, \ldots, 0)^\intercal$
  - and the second one if and only if $\sigma = (\|\sigma\|_1/r, \ldots, \|\sigma\|_1/r, 0, \ldots, 0)^\intercal$.

The Effective Rank: a Measure of Effective Dimensionality (Roy and Vetterli, 2007)

Effective rank regularization

- The goal is to maintain the effective rank of 3D Gaussians to encourage planar shapes while *penalizing ranks close to 1* to reduce needle-like artifacts.
- Although disk-like Gaussians with an effective rank of ~2 are preferred, shapes with a rank <2 are also crucial for complex geometries.
- Hyung et al. (2024) introduced an effective rank regularization term that grows exponentially as the effective rank approaches 1, heavily penalizing those Gaussians.

$$\mathscr{L}_{\text{erank}} = \lambda_{\text{erank}} \sum_k \max\left(-\log\left(\text{erank}(s_1^{(k)}, s_2^{(k)}, s_3^{(k)}) - 1 + \epsilon\right), 0\right) + s_3^{(k)}$$

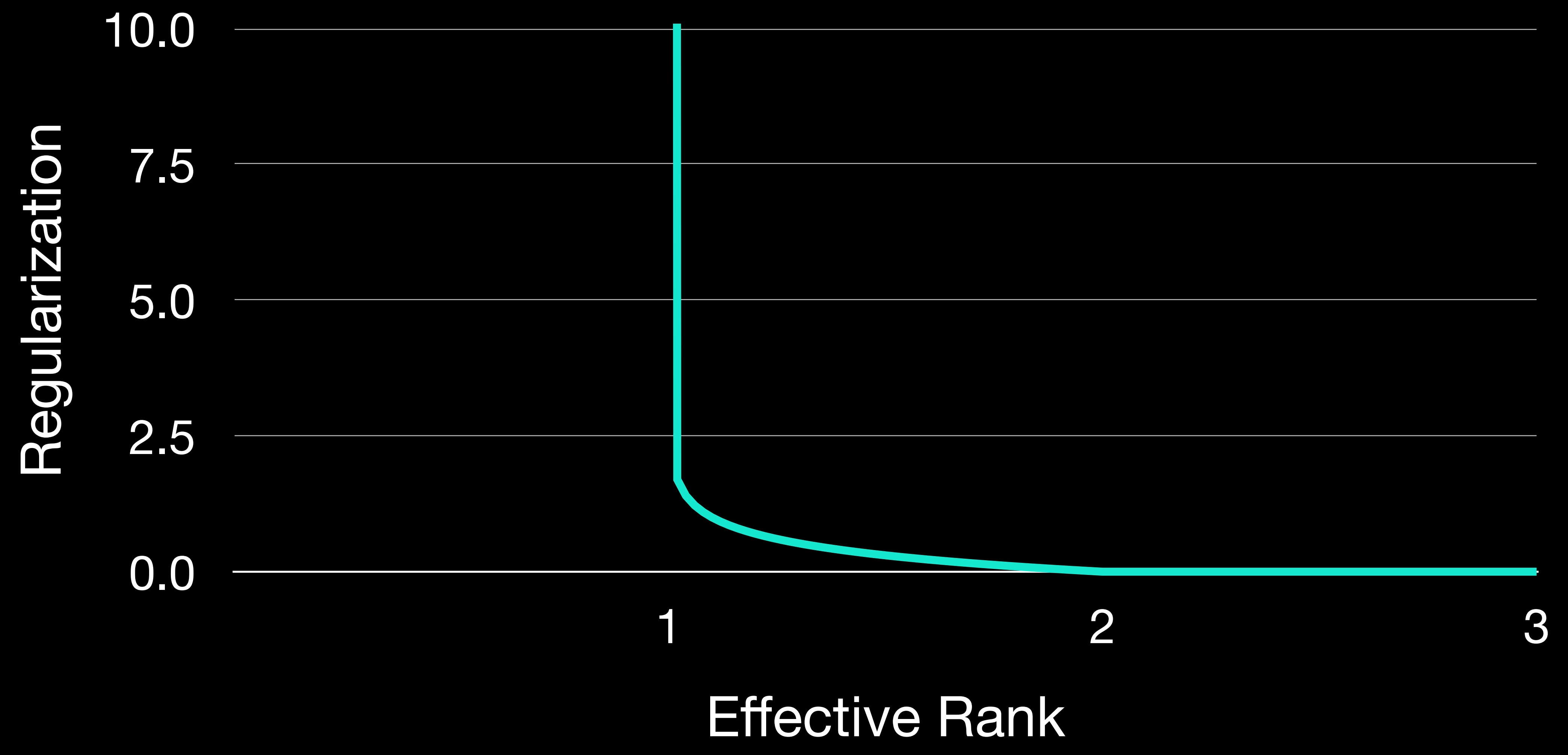
where $s_1^{(k)}, s_2^{(k)}$, and $s_3^{(k)}$ are the scale parameters* of the *k*-th Gaussian.

*Sorted in a descending order and l1-normalized.

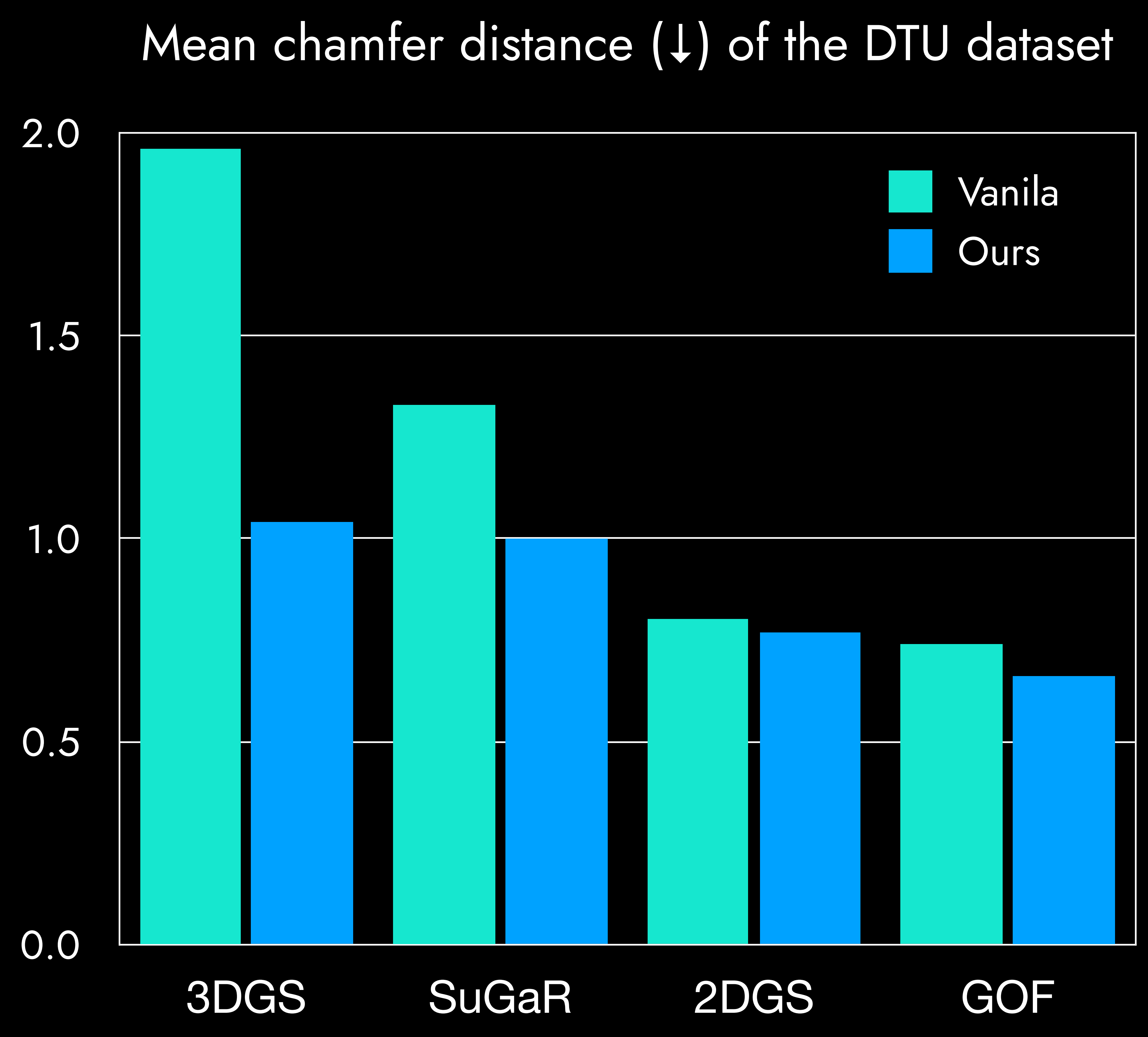
Effective rank regularization

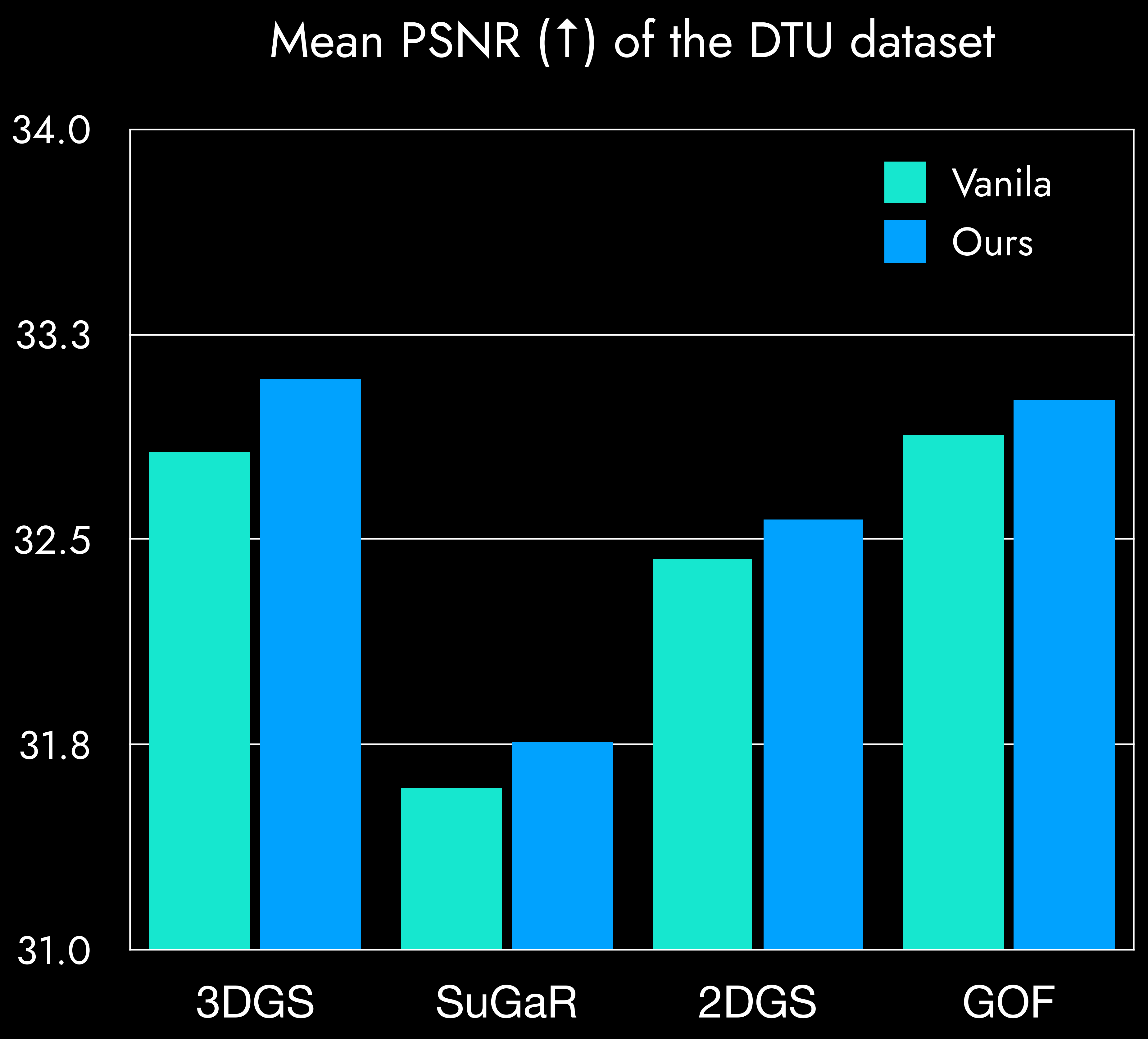
- Hyung et al. (2024) introduced an effective rank regularization term that grows *exponentially* as the effective rank approaches 1, heavily penalizing those Gaussians.

$$\mathscr{L}_{\text{erank}} = \lambda_{\text{erank}} \sum_{k} \max\left(-\log\left(\text{erank}(s_1^{(k)}, s_2^{(k)}, s_3^{(k)}) - 1 + \epsilon\right), 0\right) + s_3^{(k)}$$



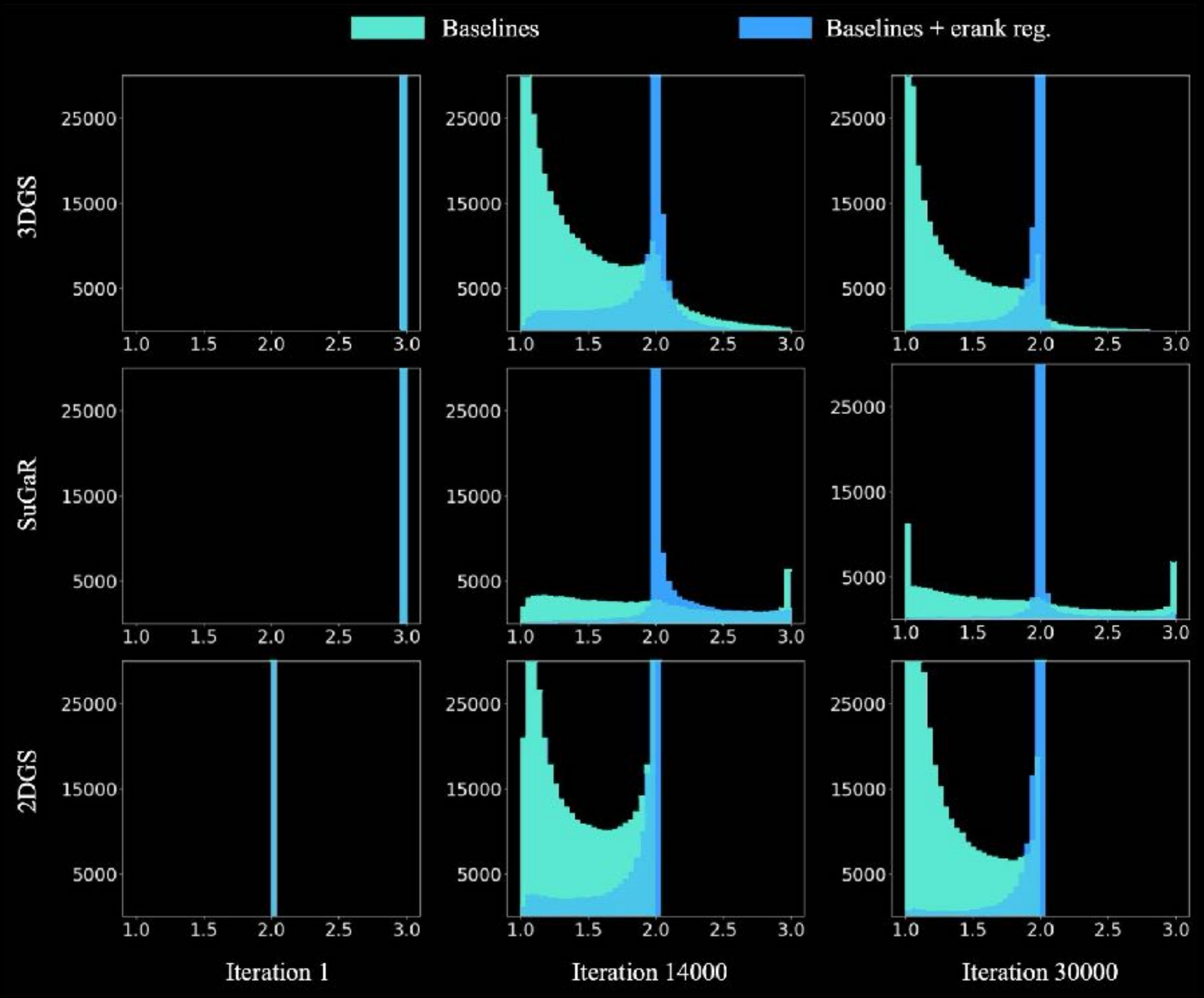
Quantitative results





Mean over 15 scenes of the DTU dataset

Effective rank analysis w/ histogram

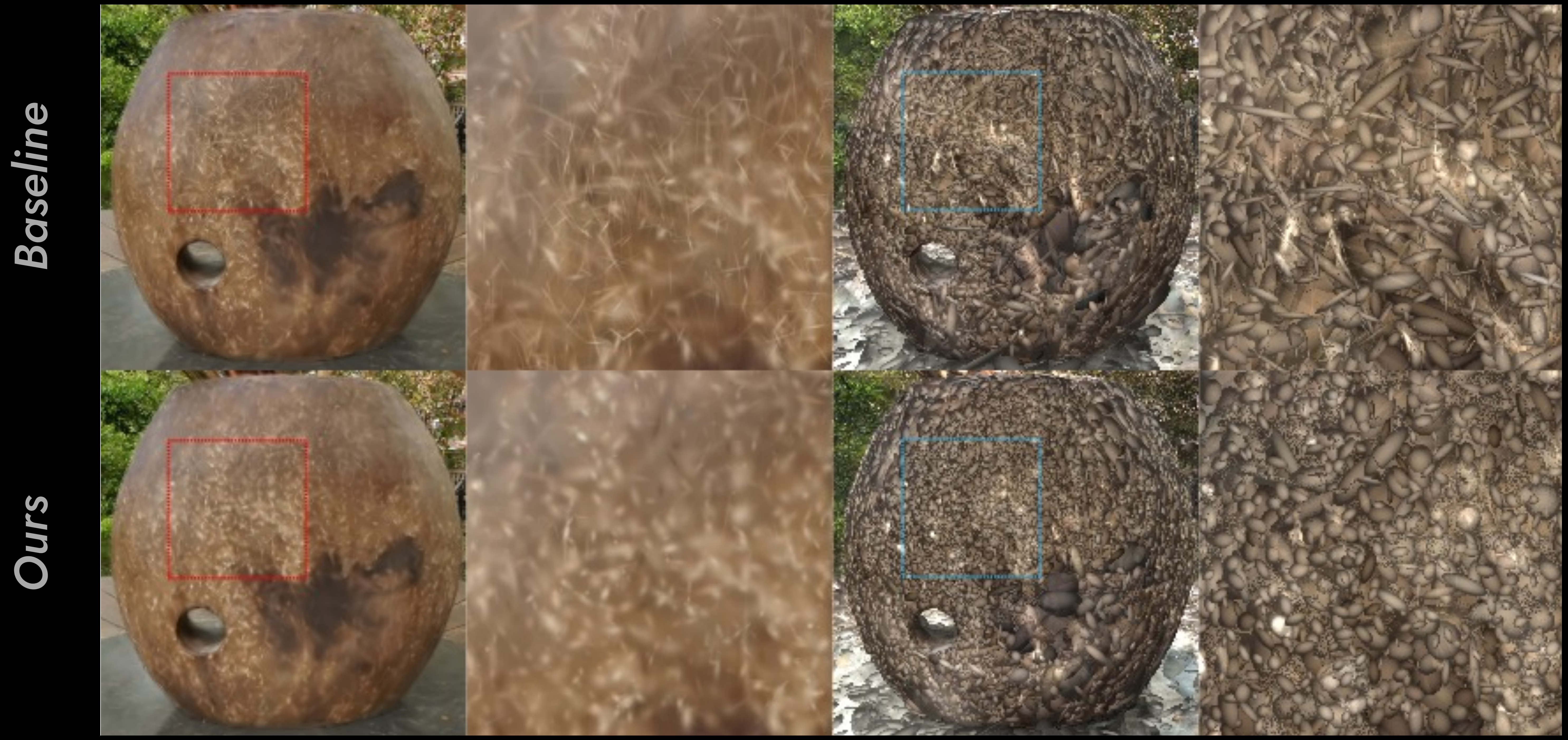


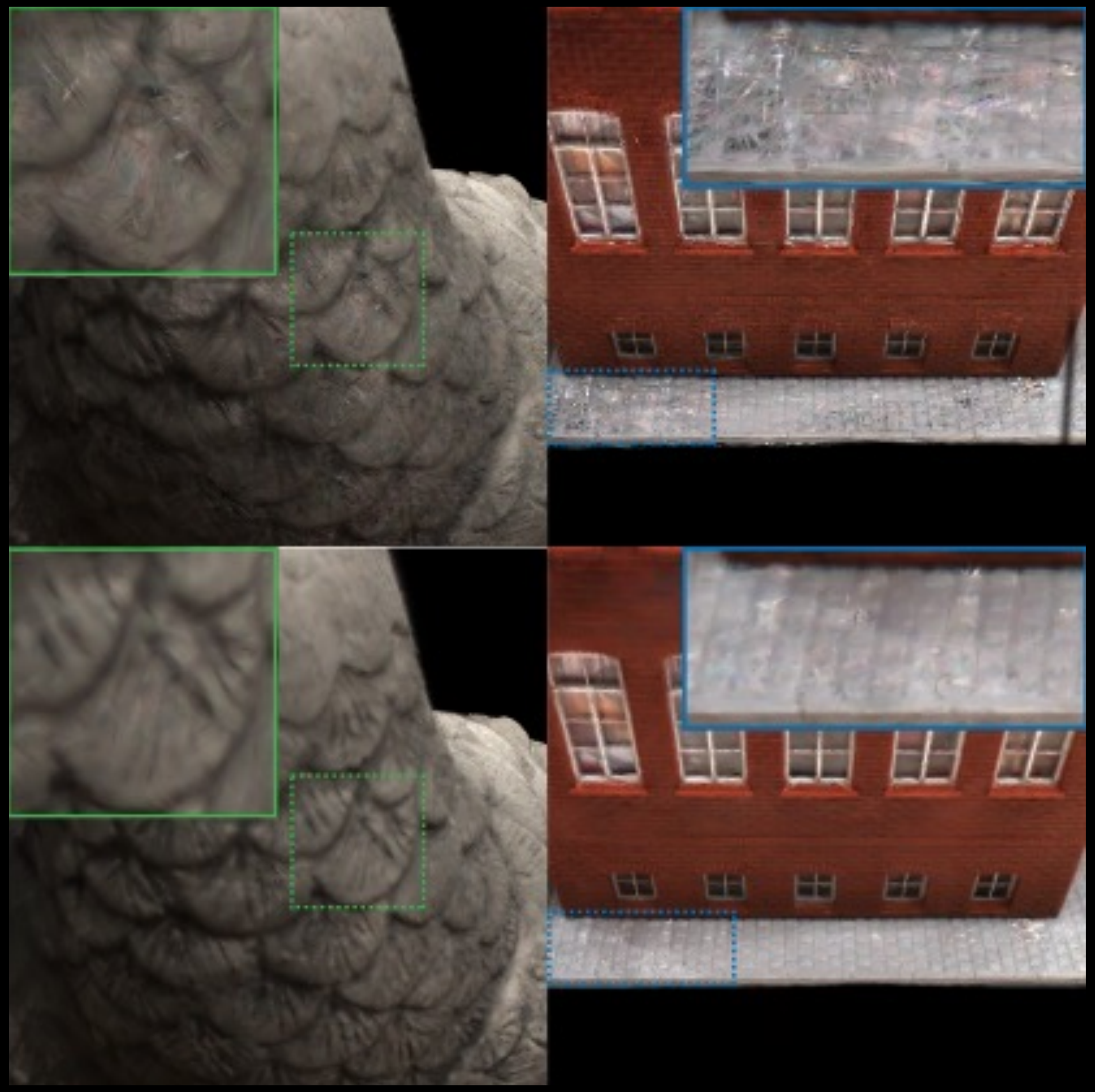
The x and y axes represent the effective rank and the corresponding population of Gaussians.

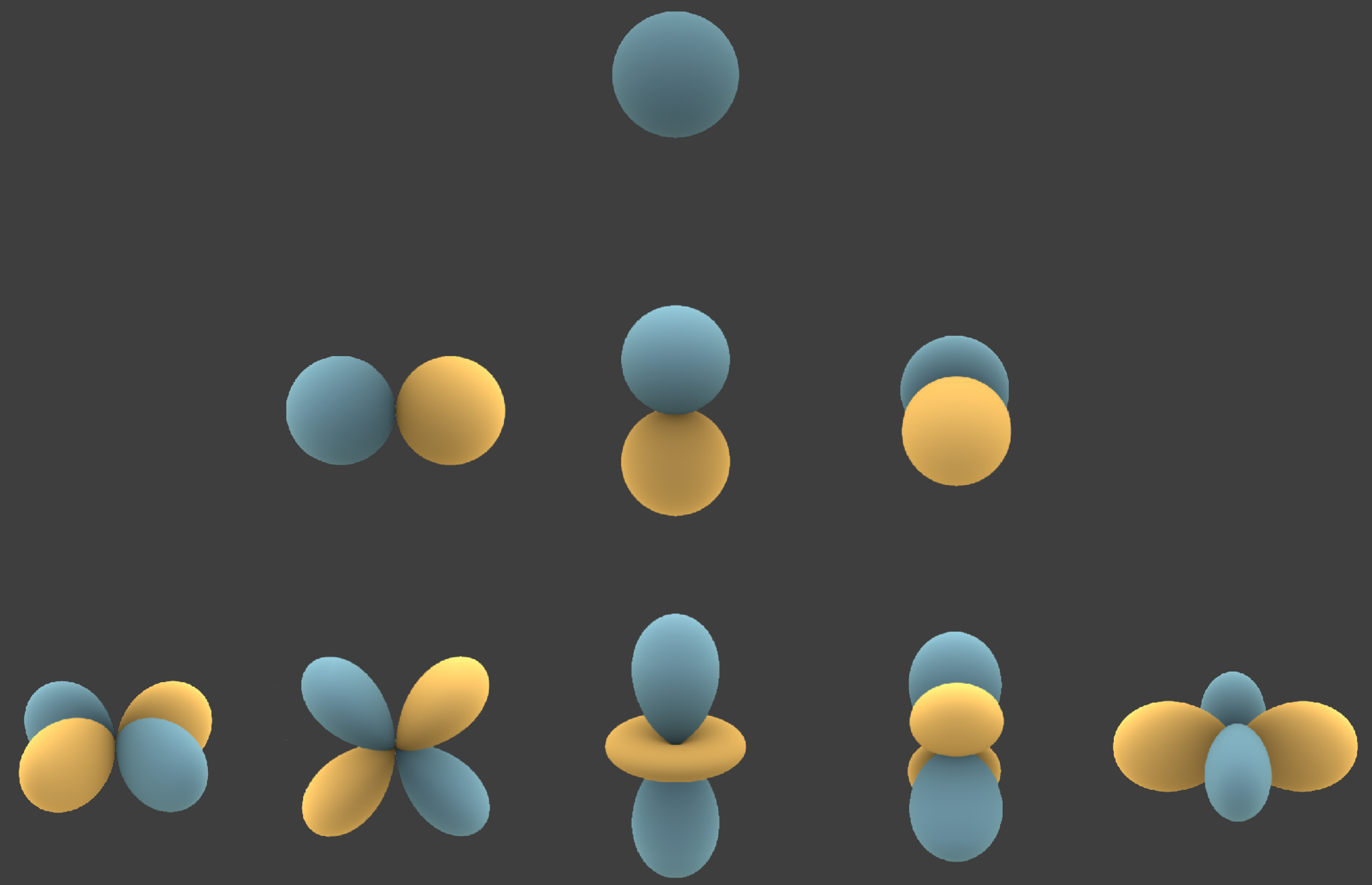
Visualizations

Mip-NeRF360 (Barron et al., 2022)

DTU (Jensen et al., 2014)







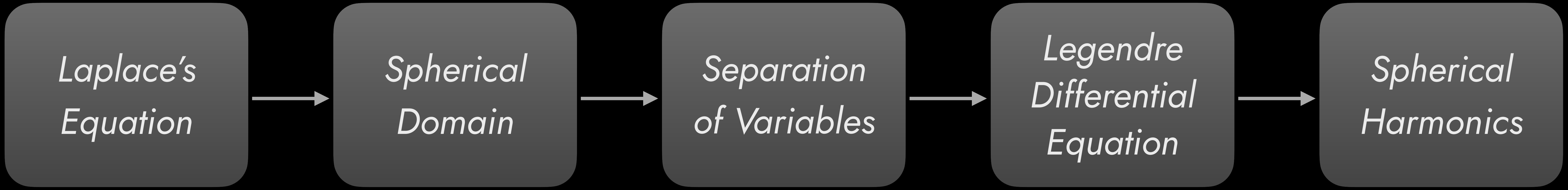
Spherical Harmonics

and viewing-dependent colors

NAVER AI LAB

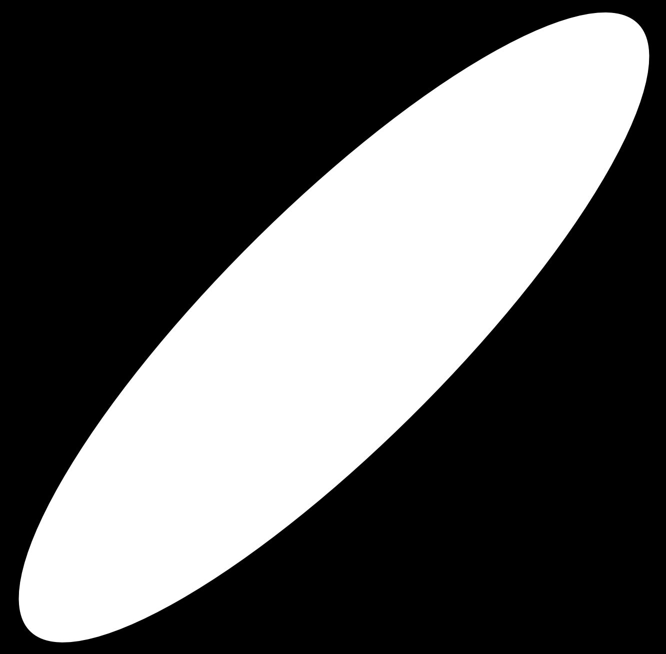
Introduction

- Special functions defined on the surface of a sphere
- Spherical harmonics (SH) form an orthonormal basis — a function defined on the surface can be represented as a sum of the spherical harmonics.
  - Similarly, periodic functions defined on a circle can be represented as a sum of sines and cosines via Fourier series.
- Spherical harmonics are solutions to *Laplace's equation* on spherical domains.



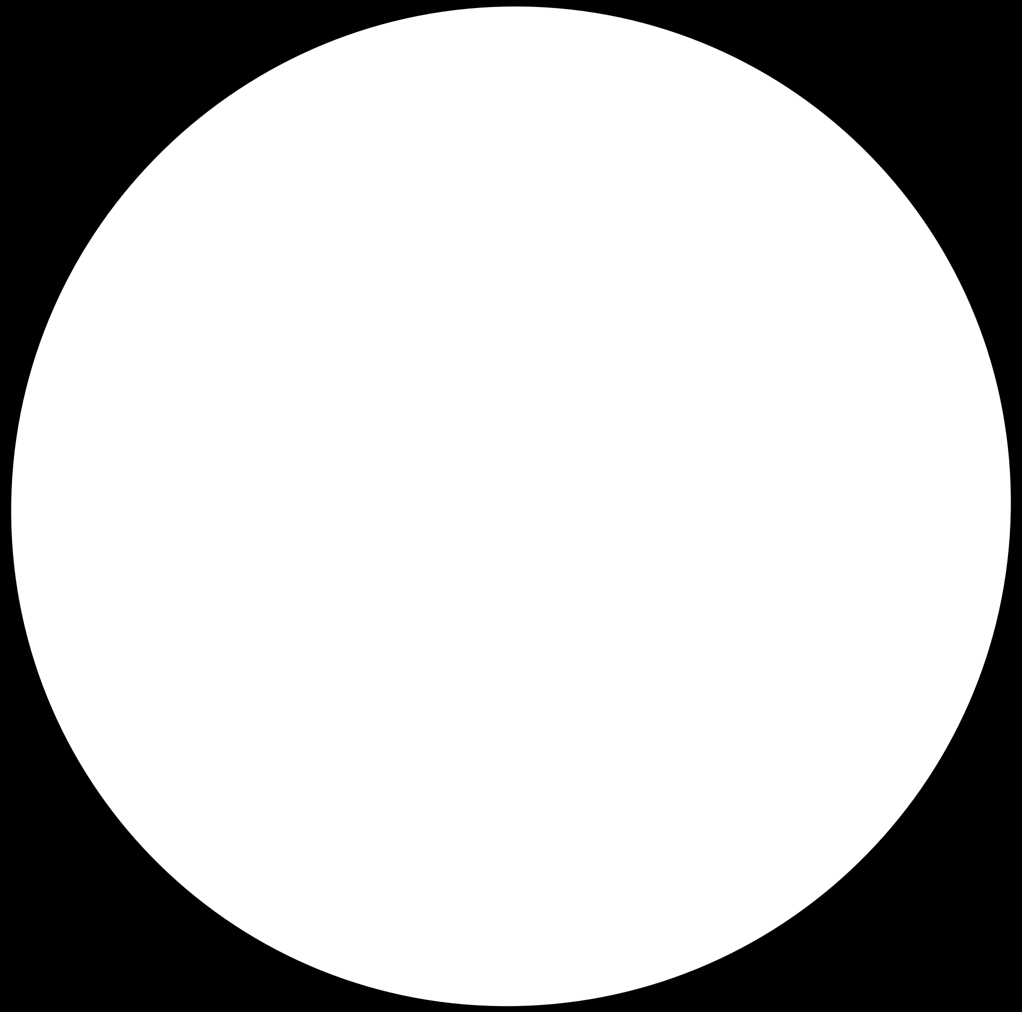
Limitations of sinusoidal encoding

- Lack of *compactness*
  - Angular signals like reflectance lobes are naturally localized and often smooth.
  - Sinusoidal features require many high-frequency terms to represent sharp lobes or directional highlights.
    - To represent such a narrow spike using a sum of sine and cosine waves, you need many high-frequency components.
  - It leads to *overfitting* or *instability* during training due to high-dimensional representation.



Limitations of sinusoidal encoding (Cont'd)

- Lack of *Isotropy*
  - Sinusoidal encoding is not 3D rotation-equivariant:
    - Encodings of directions depend on the coordinate axis ($e.g.$, $\sin(2^k \cdot \theta_x), \cos(2^k \cdot \theta_y)$)
    - This breaks symmetry — the same angular shape rotated gives different representations.
  - Bad for view-dependent effects that are naturally *rotationally symmetric*
    - A specular lobe or BRDF is often rotationally symmetric about the normal.



Origin of spherical harmonics

- To understand spherical harmonics, we begin with Laplace's equation in cartesian coordinates:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} = 0 \quad \text{(Laplace's equation)}$$

where f is a twice-differentiable real-valued function and ∇ is the gradient operator.

- In spherical domains, we get*:

$$\nabla^2 f = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial f}{\partial r}\right) + \frac{1}{r^2 \sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta \frac{\partial f}{\partial \theta}\right) + \frac{1}{r^2 \sin^2\theta}\frac{\partial^2 f}{\partial \varphi^2} = 0$$

using $x = r\sin\theta\cos\phi$, $y = r\sin\theta\sin\phi$, and $z = r\cos\theta$.

* Griffiths, Introduction to Electrodynamics, 2013

Origin of spherical harmonics (Cont'd)

- Assume a separable solution, $f(r, \theta, \varphi) = R(r)Y(\theta, \varphi)$ leading* to separation into radial and angular parts after rearrangement of terms. For some λ,

$$\lambda = \frac{1}{R}\frac{\partial}{\partial r}\left(r^2\frac{\partial R}{\partial r}\right),$$

$$-\lambda = \frac{1}{Y\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial Y}{\partial\theta}\right) + \frac{1}{Y\sin^2\theta}\frac{\partial^2 Y}{\partial\varphi^2}.$$

- The second equation can be simplified further under the assumption that $Y(\theta, \varphi) = \Theta(\theta)\Phi(\varphi)$ to apply separation of variables again.

* Both sides of Laplace's equation are multiplied by $r^2/(RY)$.

Origin of spherical harmonics (Cont'd)

- For a complex constant m,

$$-m^2 = \frac{1}{\Phi}\frac{\partial^2\Phi}{\partial\varphi^2}$$

$$m^2 = \lambda\sin^2\theta + \frac{\sin\theta}{\Theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial\Theta}{\partial\theta}\right).$$

- To specify m, Φ must be a periodic function whose period is evenly divided by 2π. Thus, $\Phi(\varphi) = \Sigma_{m=-\infty}^{\infty} c_m e^{im\varphi}$ in the Fourier series.
- The *eigenfunctions** take the form of $e^{im\varphi}$ which makes the above first equation.

* Eigenfunction is a type of eigenvector.

Origin of spherical harmonics (Cont'd)

- The solution function $Y(\theta, \varphi)$ is regular at the *poles* of the sphere. Since when $\theta = 0$, φ is necessarily π.
- Imposing this regularity at the boundary points of our spherical domain is a well-known *Sturm-Liouville theory** (not discussed in our class.)
- The solution leads to that $\lambda = \ell(\ell + 1)$ for some non-negative integer with $\ell \geq |m|$.
- Then, substituting $x = \cos\theta$ rearranges into the Legendre differential equation.

Use this: $\frac{\partial}{\partial\theta} = \frac{\partial}{\partial x}\frac{\partial x}{\partial\theta} = -\sin\theta\frac{\partial}{\partial x}$

* https://en.wikipedia.org/wiki/Sturm–Liouville_theory

Associated Legendre polynomials

- The *associated* Legendre polynomial has nonzero and nonsingular solutions on [-1, 1], which is satisfied by $\cos\theta$, where ℓ and m are integers with $0 \leq |m| \leq \ell$.

$$(1-x^2)\frac{d^2P_\ell^m(x)}{dx^2} - 2x\frac{dP_\ell^m(x)}{dx} + \left[\ell(\ell+1) - \frac{m^2}{1-x^2}\right]P_\ell^m(x) = 0$$

- The Legendre ODE is frequently encountered in physics and other technical fields.
- Thus, the total number of spherical harmonics up to a given degree ℓ is $(\ell+1)^2$.
- 🤔 Laplace's equation guarantees it behaves smoothly and predictably.

Origin of spherical harmonics (Cont'd)

- The angular part satisfies the associated Legendre differential equation*. Its known *ordinary differential equation (ODE)* solutions are the spherical harmonics.

$$Y_\ell^m(\theta, \phi) = \sqrt{\frac{(2\ell + 1)}{4\pi} \cdot \frac{(\ell - m)!}{(\ell + m)!}} P_\ell^m(\cos \theta) \cdot e^{im\phi}$$

where $f(r, \theta, \phi) = R(r)\Theta(\theta)\Phi(\phi) = R(r)Y_\ell^m(\theta, \phi)$; ℓ, m, and P_ℓ^m denotes degree, order, and associated Legendre polynomial, respectively.

- The normalization constant $N(\ell, m)$ comes from the orthogonality of spherical harmonics — the integral of the product of two harmonics is one if they are the same.

* You can prove using the fact that Φ is a twice-differentiable real-valued function! [Proof]

Origin of spherical harmonics (Cont'd)

- We use the spherical harmonics

$$Y_\ell^m(\theta, \phi) = \sqrt{\frac{(2\ell+1)}{4\pi} \cdot \frac{(\ell-m)!}{(\ell+m)!}} P_\ell^m(\cos\theta) \cdot e^{im\phi}$$

to represent a function on the surface of a unit sphere or a directional vector.

- For fine-grained approximations, we exploit a sum of spherical harmonics over varying degrees ℓ and orders m.

Associated Legendre polynomials

- *Associated* Legendre polynomials are defined as:

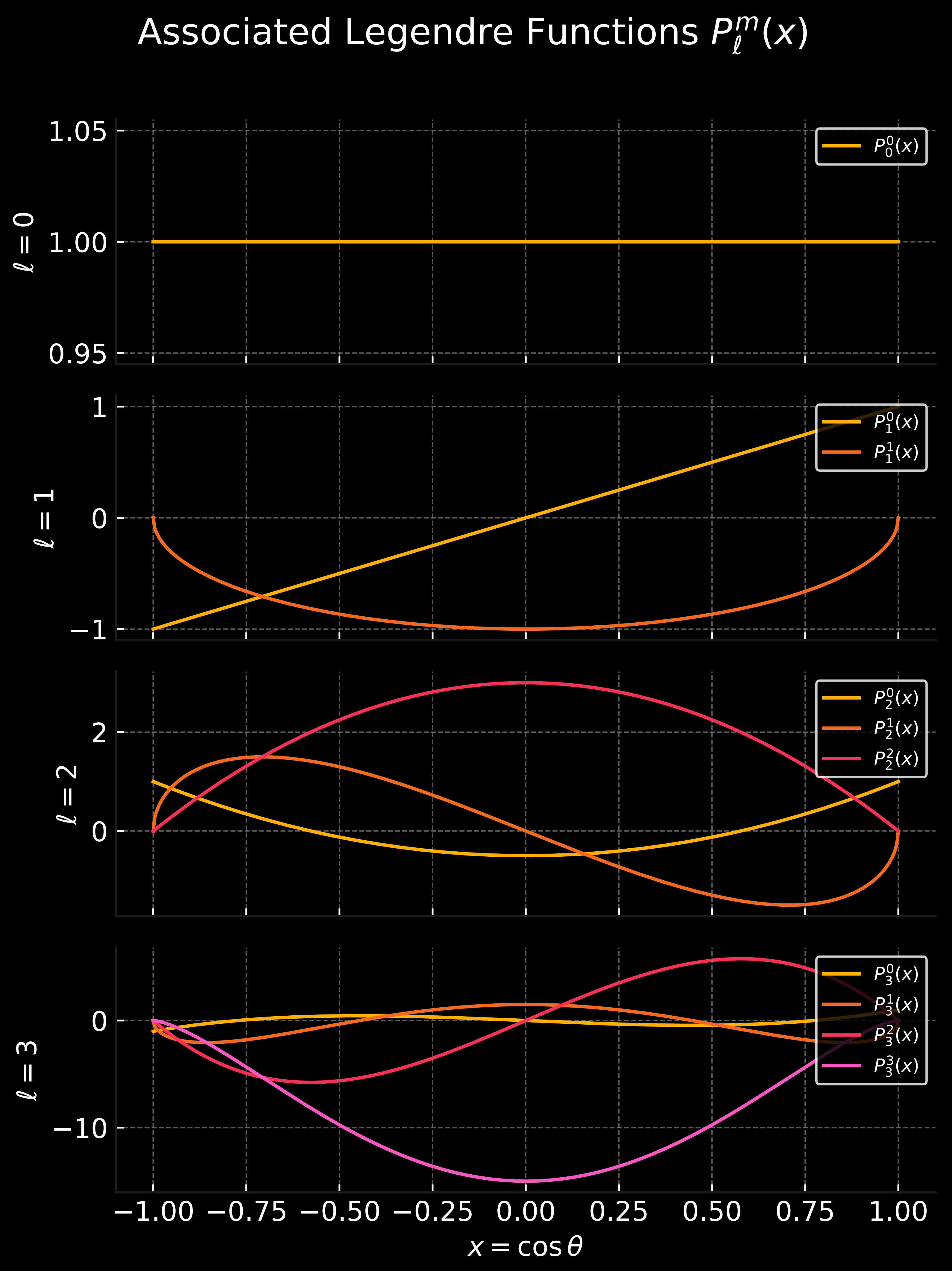
$$P_\ell^m(x) = (-1)^m(1-x^2)^{m/2}\frac{d^m}{dx^m}P_\ell(x).$$

- *Rodrigues' formula** generates the *ordinary* Legendre polynomials as:

$$P_\ell(x) = \frac{1}{2^\ell \ell!}\frac{d^\ell}{dx^\ell}\left(x^2-1\right)^\ell.$$

* Olinde Rodrigues (1816), Sir James Ivory (1824) and Carl Gustav Jacobi (1827).

Associated Legendre polynomials (Vis')



Rotation equivariance

- Let $f(\mathbf{v})$ be a function on the sphere (e.g., a BRDF lobe), and let's express it in SH:

$$f(\mathbf{v}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} c_\ell^m Y_\ell^m(\mathbf{v})$$

- If we rotate the function by rotation R, the rotated function is: $(Rf)(\mathbf{v}) = f(R^{-1}\mathbf{v})$.
- Then, the SH coefficients of the rotated function are linear transforms of the original:

$$c_\ell^{m'\prime} = \sum_{m=-\ell}^{\ell} D_{m'm}^{(\ell)}(R) \cdot \mathbf{c}_\ell^m$$

where $D_{m'm}^{(\ell)}(R)$ is the Wigner D-matrix* for rotation R at degree ℓ, It's a known unitary matrix that depends only on R and ℓ.

* Eugene Wigner (1927)

Angular frequency

- Each spherical harmonic of degree ℓ oscillates on the sphere roughly ℓ times around the great circle (ℓ-zero crossings).
- So the angular wavelength — the angle between peaks — is roughly:

$$\text{Wavelength} \approx \frac{360^\circ}{2\ell} = \frac{180^\circ}{\ell}.$$

- One full cycle of a wave has two lobes: positive and negative.
- Later, we will discuss the Laplace-Beltrami eigenvalue of spherical harmonics and it suggests that the frequency is approximately $\sqrt{\ell(\ell+1)}$.

SH in 3D gaussian splatting

- To efficiently model view-dependent color per Gaussian, each Gaussian stores 48 SH coefficients (16 per RGB channel when $\ell \leq 3$).
- The color from a viewing direction $\mathbf{v}$ is:

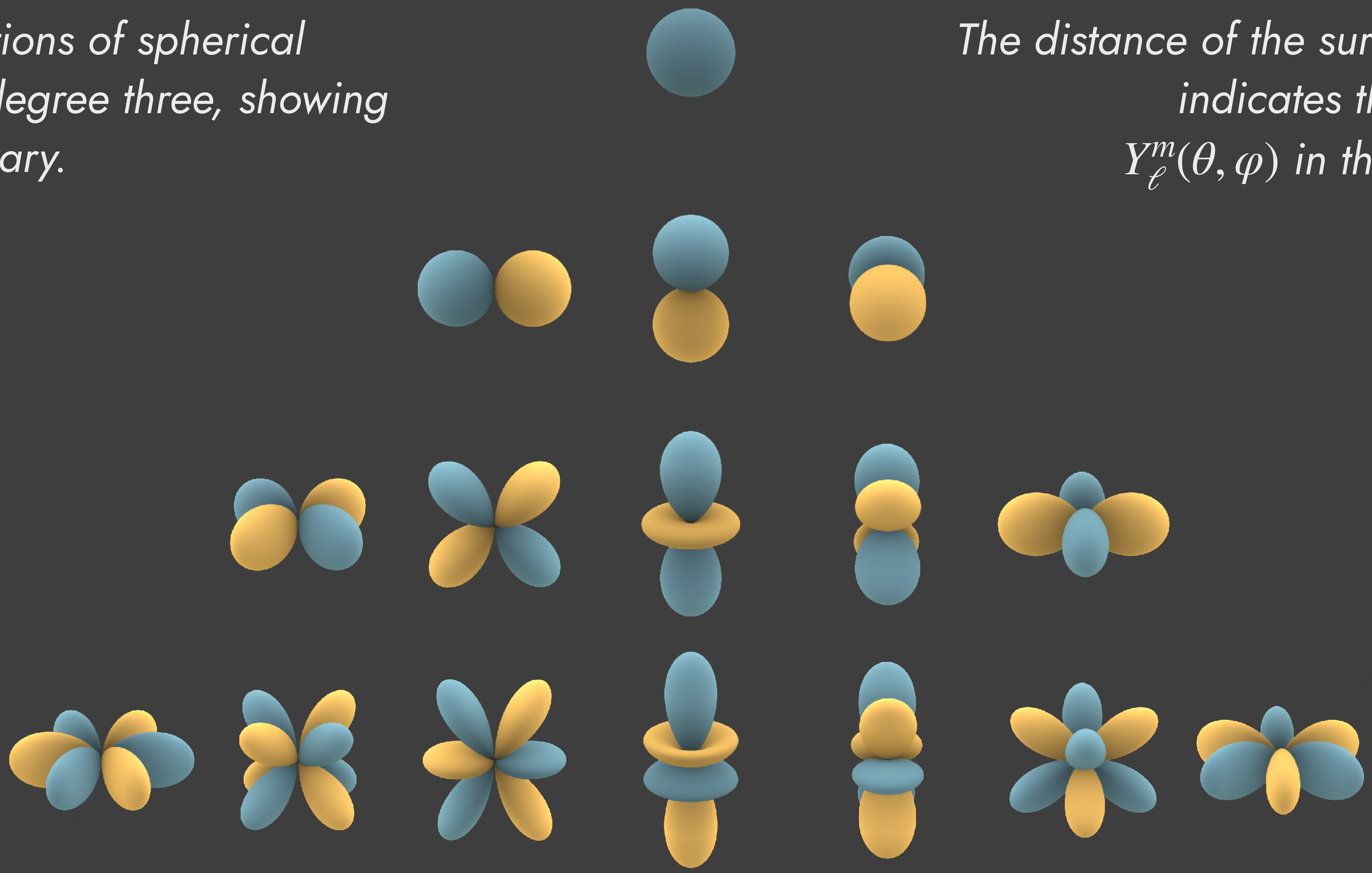
$$c(\mathbf{v}) = \sum_{l=0}^{3} \sum_{m=-\ell}^{l} c_{\ell,m} \cdot Y_\ell^m(\mathbf{v})$$

where $c_{\ell,m}$ denotes learned *scalar* coefficients.

- It captures low-frequency angular variation and is fast to evaluate during rasterization.

Visual representations of spherical harmonics up to degree three, showing how their values vary.

The distance of the surface from the origin indicates the absolute value of $Y_\ell^m(\theta, \varphi)$ in the angular direction.



Blue portions represent regions where the function is positive, and yellow portions represent where it is negative.

Atomic orbitals shaped by SH

- Atomic orbitals are the exact solutions to the Schrödinger equation for hydrogen-like atoms—systems with a single electron bound by a Coulomb potential.

$$\left[-\frac{\hbar^2}{2m}\nabla^2+V(x)\right]\psi(x)=E\psi(x)$$

where $\hbar$: reduced Planck constant, m: particle mass, $V(x)$: potential energy, E: total energy, and $\psi(x)$: a wavefunction.

- In the Schrödinger equation, Laplace's equation $\nabla^2\psi=0$ arises when potential and energy vanish (no external forces), describing a free particle.
- Whose solutions are the spherical harmonics that define the shape of the orbitals.

3DGS Efficiency

Scaffold-GS and Octree-GS

Anchor points of Scaffold-GS

- *Anchor points* are used to predict the view-dependent neural Gaussians in their corresponding voxels following Scaffold-GS (Lu et al., 2023).
- Each *anchor point* has a context feature $\hat{\mathrm{f}}_v$, a location x_v, a scaling factor l_v, and k -learnable offset $\{\mathcal{O}_0, \ldots, \mathcal{O}_{k-1}\}$. The position of neural Gaussians is the sum of the anchor position and learning offsets as follows:

$$\{\mu_0, \ldots, \mu_{k-1}\} = x_v + \{\mathcal{O}_0, \ldots, \mathcal{O}_{k-1}\} \cdot l_v.$$

- Opacities, scales, rotations, and colors are decoded from the anchor features through corresponding MLPs. For example, the opacities are using MLP F_α as follows:

$$\{\alpha_0, \ldots, \alpha_{k-1}\} = \mathrm{F}_\alpha(\hat{\mathrm{f}}_v, \Delta_{vc}, \tilde{d}_{vc})^*.$$

* Δ_{vc} is the relative viewing distance, while $\tilde{d}_{vc}$ is the direction to the camera (opposite viewing direction).

Anchor building of Octree-GS

- Based on Scaffold-GS, Octree-GS (Ren & Jiang et al., 2024) initializes the position of anchor points from a set of sparse SfM points P, based on the Level of Detail (LoD).
- To determine the *global LoD K*, we find the largest and smallest distances* between the camera centers of training images and SfM points as d_{max} and d_{min}, respectively:

$$K = \lfloor \log_2(\hat{d}_{max}/\hat{d}_{min}) \rceil + 1$$

where $\lfloor \cdot \rceil$ denotes the round operator.

- For each LoD layer, through 0 to K-1, the anchors** of each layer L are as follows:

$$\mathrm{V}_L = \left\{ \left\lfloor \frac{\mathrm{P}}{\delta/2^L} \right\rceil \cdot \delta/2^L \right\}$$

where the base voxel size δ for the coarsest layer corresponds to LoD 0.

*One can use a percentile of 99.9% to exclude outliers. **Duplicated anchors at the coarse level may be discarded.

Anchor selection of Octree-GS

- In training and inference, we subsample the Gaussian primitives based on LoD.
- For the observation distance d_{ij} between the camera center i and an anchor j, the maximum LoD $\hat{L}_{ij}$ is defined as follows:

$$\hat{L}_{ij} = \lfloor L^{\star}_{ij} \rfloor = \lfloor \Phi(\log_2(d_{max}/(d_{ij} \cdot s))) + \Delta L_j \rfloor$$

where Φ is a clamping function to be in $[0, K-1]$, s is a focal scale factor adjusting camera intrinsic variation, and ΔL is a learnable LoD bias.

- Finally, the anchor j is selected for rendering if its LoD level $L_j \leq \hat{L}_{ij}$ is met, where L_j (implicitly) denotes the anchor j's LoD.

Opacity blending

- To ensure smooth rendering transitions between different LoD levels, one-degree finer LoD anchors are also used where $L_j = \hat{L}_{ij} + 1$.
- The Gaussian primitives' opacity of these anchors are scaled by $L_{ij}^{\star} - \hat{L}_{ij}$, which is bounded by $0 \leq L_{ij}^{\star} - \hat{L}_{ij} < 1$ due to the floor operation in $\hat{L}_{ij} = \lfloor L_{ij}^{\star} \rfloor$.

Anchor growing

- Similar to Scaffold-GS (Lu et al., 2023), they calculate the average accumulated gradient of the spawned Gaussian primitives ∇_g, checking if it exceeds a threshold.
- It will be promoted and converted into a new anchor in the next-level LoD voxel if it is located in an empty voxel at every T iteration.

  🤔 *"Is the position of the Gaussian primitive used for the new anchor in the next-level LoD, while the previous Gaussian primitive remains as the output of the MLP?"*

- To prevent aggressive growth into higher LoD levels, we set the threshold as follows: $\tau_g^L = \tau_g \cdot 2^{\beta L}$ where τ_g and β are hyperparameters*.

*The default values for τ_g and β are 2e-4 and 2e-1, respectively.

Anchor pruning

- The average opacity of Gaussians generated over T training iterations is the input to a pruning criterion for the anchor, like Scaffold-GS (Lu et al., 2023).

Progressive training

- They gradually activate finer levels of LoD during training, allowing all LoD levels to try their best to represent the scene.
- Starting from $\left\lfloor \frac{K}{2} \right\rfloor$ level, iteratively activate an additional LoD level after N_L iterations.
- Coarse-grained anchors are trained longer as $N_{i-1} = \omega N_i$, while $\omega \geq 1$.

FLoD: Flexible Level of Detail

Seo & Choi et al., SIGGRAPH 2025

Flexible level of detail

- Compared to Octree-GS (Ren & Jiang et al., 2024), several points differ:
  - They don't use the octree structure, while FLoD (Seo & Choi et al., 2025) imposes each level *independently to capture* the overall scene.
  - Progressive training starts by cloning the previous LoD, while lowering the 3D scale constraint to capture finer-grained LoDs.
- However, it can be seamlessly integrated with Scaffold-GS.

3D scale constraint

- For each level ℓ where $\ell \in [1, L_{max}]$, the minimum 3D scale $s_{min}^{(\ell)}$ is defined as:

$$s_{min}^{(\ell)} = \begin{cases} \lambda \times \rho^{1-\ell} & \text{if } 1 \leq \ell < L_{max} \\ 0 & \text{otherwise} \end{cases}$$

  while λ is the initial 3D scale constraint, and ρ is the factor by which it is *progressively reduced* at each subsequent level.

- There is *no maximum* for the scale as: $s^{(\ell)} = e^{s_{opt}} + s_{min}^{(\ell)}$, where only s_{opt} is learnable.

The initial 3D scale constraint λ = 0.2 and the scale factor ρ = 4, which makes $s_{min}^{(3)} = 0.2 \times 4^{-2} = 1/80$.

Exponential parameterization

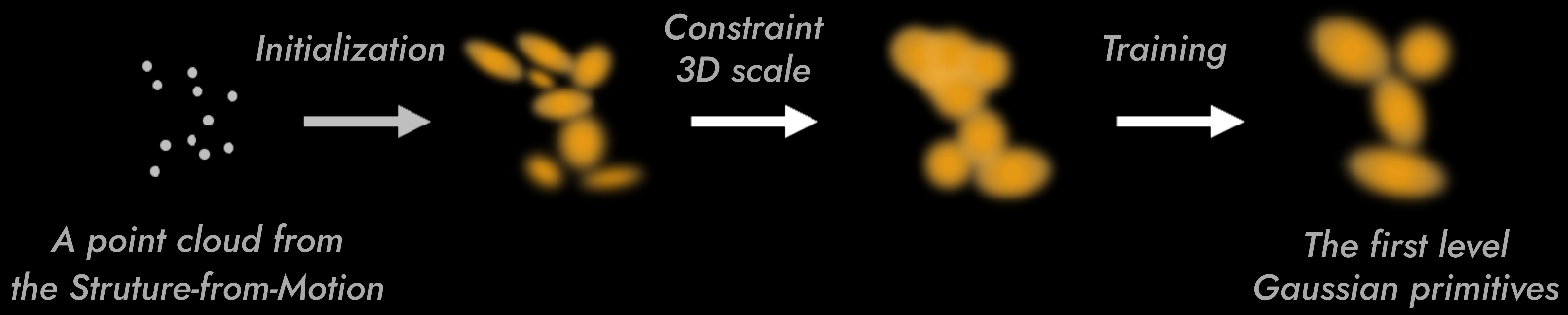
- Scale must *be positive*, *i.e.*, $\exp(x) > 0$, ensuring *numerical stability*.
- *Multiplicative by nature*:
  - Including *scale*, *radii*, *distances*, and *amplitudes* also grow multiplicatively.

$$\exp(x + \Delta) = \exp(x) \cdot \exp(\Delta)$$

  - A small linear step for x corresponds to multiplying the scale by a factor.
  - That's why people say exponential is the *natural coordinate* for scale.
- At times, it is necessary to ensure that x remains within a valid range, thereby avoiding *overflow* or *underflow*.

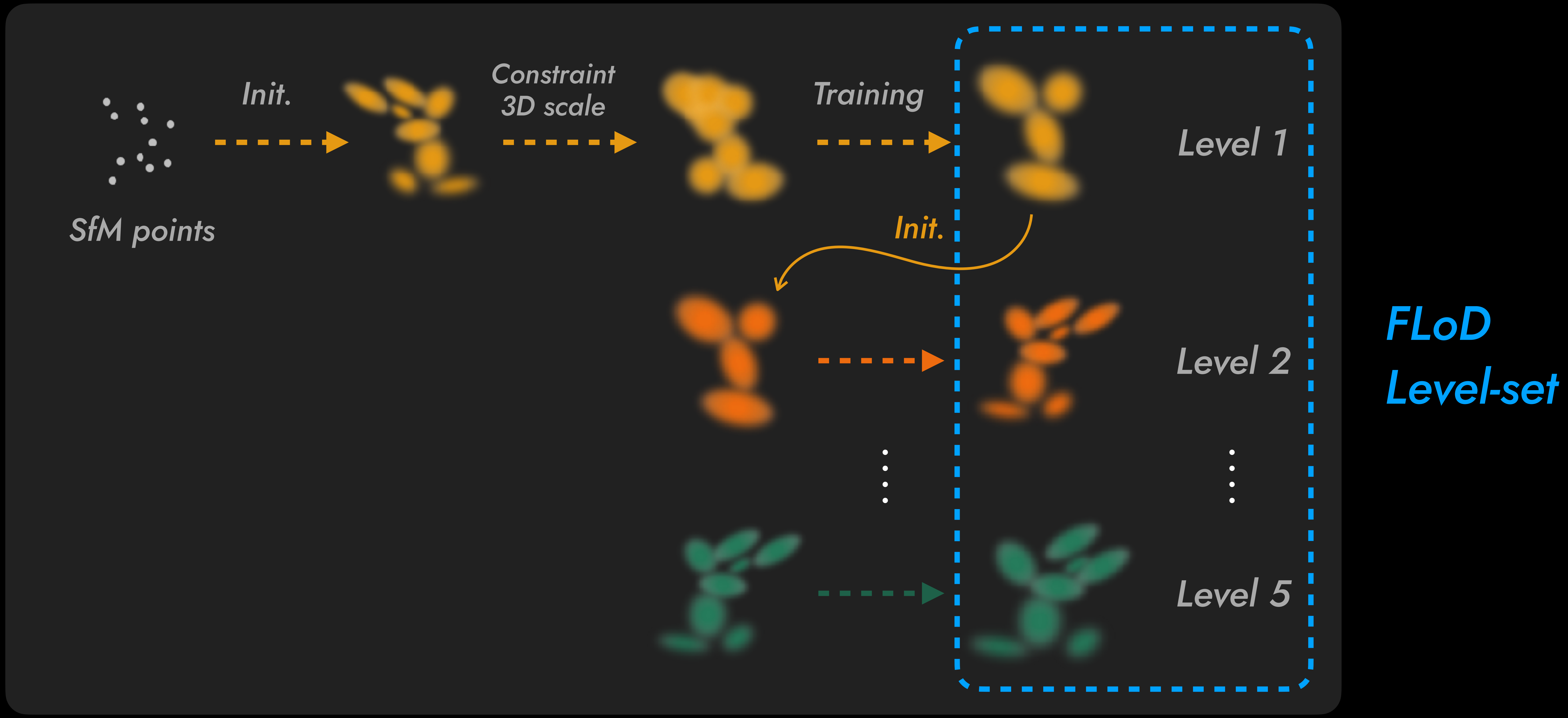
Training and pruning

- The first level starts from the SfM points with the first level-specific 3D scale constraint.
- The Gaussians whose average distance to their three nearest neighbors is less than half of the 3D scale constraint $s_{min}^{(\ell)}$ are pruned to further reduce the memory footprint.



Progressive training

- Save the trained Gaussian primitives at the current level ℓ.
- The copy initializes the next level $\ell + 1$, *seamlessly* with a smaller-scale constraint.



Seamless next-level constraint

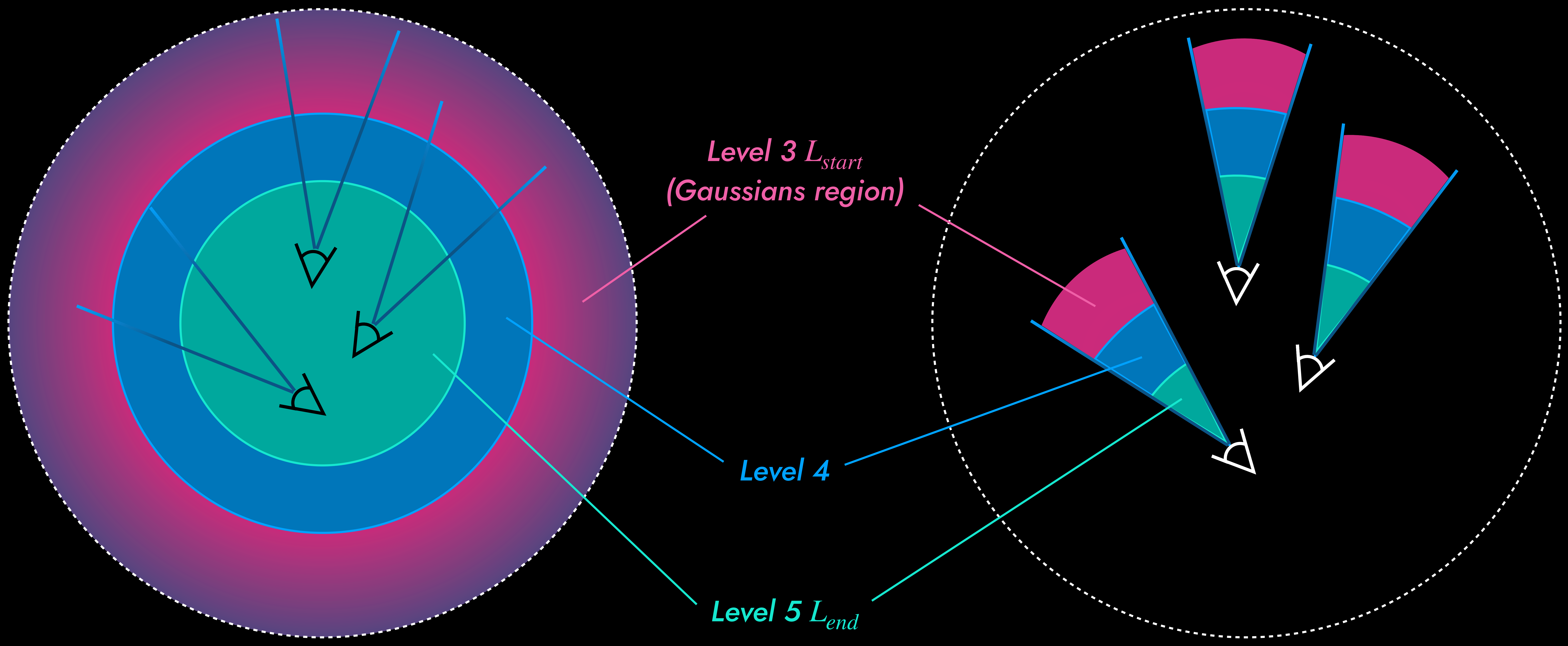
- The initialized Gaussians of the next level $\ell + 1$ have learnable parameters so that:

$$\mathrm{s}_{opt}^{(\ell+1)} = \log\left(\mathrm{s}^{(\ell)} - s_{min}^{(\ell+1)}\right),$$

  which makes $\mathrm{s}_{init}^{(\ell+1)} = \mathrm{s}^{(\ell)}$ having a new $s_{min}^{(\ell+1)}$.

- This prevents abrupt fluctuations in the initial training loss by preserving the computed scale values.

Two LoD options



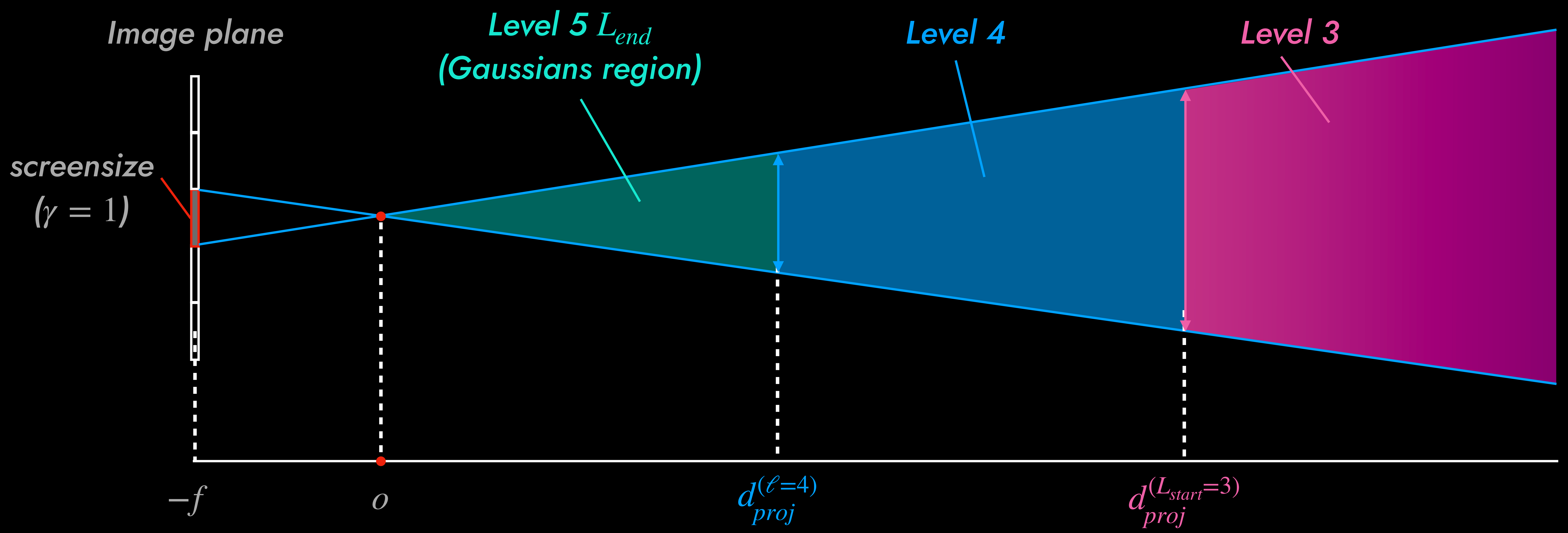
(a) Pre-determined

(b) Per-view

Selective rendering

- Each Gaussian level range is determined by $d_{proj}^{(\ell)}$ defined as follows:

$$d_{proj}^{(\ell)} = \frac{s_{min}^{(\ell)}}{\gamma} \times f.$$



The unit of γ and f is a pixel, while d_{proj} and s_{min} are in the world coordinate system.

Single-level rendering evaluation

Level 3 := $\Sigma_{\ell=1}^{3}$ level ℓ

Only level 3!

Selective rendering evaluation

Bonsai in the Mip-NeRF 360 Dataset with levels 5 and 4 of the FLoD.

Selective rendering evaluation

Treehill in the Mip-NeRF 360 Dataset with levels 5, 4, 3, and 2 of the FLoD.

Model efficiency

Compared with Hierarchical-3DGS (Kerbl et al., 2024).

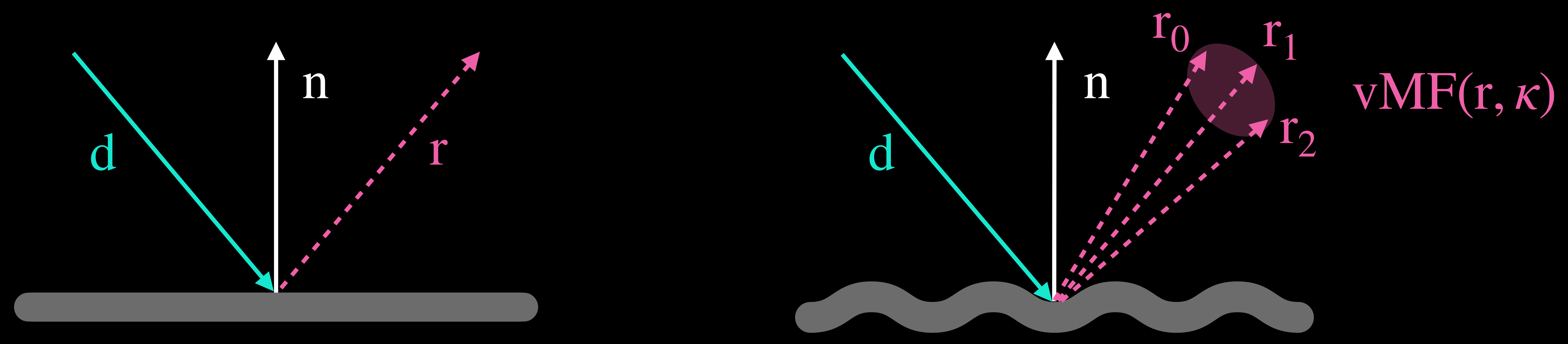
Surface Roughness & Integrated Encoding

NAVER AI LAB

Surface roughness matters

Surface roughness

- *Surface roughness*, or simply *roughness*, is the quality of a surface that is not smooth.
- From a mathematical perspective, it is related to the spatial variability structure of surfaces, and it is inherently a multiscale property, ranging from fine to coarse scales.
- However, a simple model often approximates it using the von Mises–Fisher (vMF) distribution, where the concentration parameter κ acts as the *inverse of roughness*.



von Mises–Fisher distribution

- The von Mises–Fisher distribution $\mathrm{vMF}(\mu, \kappa)$ models the directional spread of vectors on the unit sphere, similar to the Gaussian distribution on a plane.
- It is defined by a mean direction $\mu \in \mathbb{S}^{d-1}$ and a concentration parameter $\kappa \geq 0$, which controls how tightly directions cluster around μ.
- Its probability density function* is given by:

$$f(\mathrm{x}; \mu, \kappa) = C_d(\kappa)\exp(\kappa\mu^\top \mathrm{x})$$

where $C_d(\kappa)$ is a normalization constant and x is a unit vector.

* It has an association with multivariate Gaussian distribution, and remind that $\mu^\top\mu = \mathrm{x}^\top\mathrm{x} = 1$.

From vMF to IDE

- The von Mises–Fisher (vMF) distribution captures directional uncertainty using a single concentration parameter κ, modeling roughness as the angular spread.
- In Ref-NeRF (Verbin et al., 2021), *integrated directional encoding (IDE)* is used to approximate *the convolution of a vMF lobe with spherical harmonics*, enabling efficient reflection modeling under varying roughness.
- This bridges statistical directional modeling (vMF) with spectral angular representations (SH), where κ controls the bandwidth and smoothness of reflected light in the SH domain.

Closed-form solution of IDE

Integrated directional encoding. The expected value of any spherical harmonic under a vMF distribution has the following simple closed-form expression:

$$\mathbb{E}_{\mu\sim\mathrm{vMF}(\mu_r,\kappa)}[Y_\ell^m(\mu)] = A_\ell(\kappa)Y_\ell^m(\mu_r) \quad \text{where} \quad A_\ell(\kappa) \approx \exp\left(-\frac{\ell(\ell+1)}{2\kappa}\right).$$

- We assume vMF blurring attenuates each degree ℓ similarly to Gaussian decay in Fourier space.
- It's the eigenvalue decay for the Laplace–Beltrami operator on the sphere. On the sphere, the Laplacian eigenvalues for SH of degree l are: $\Delta_{\mathbb{S}^2} Y_l^m = -l(l+1)Y_l^m$.
- Notice that, in the Fourier domain, Gaussian convolution corresponds to multiplying Fourier coefficients by a Gaussian: $\hat{f}(\xi) = \exp\left(-\xi^2\sigma^2/2\right) \cdot f(\xi)$. *ref.* Mip-NeRF (Barron et al., 2021)

*Based on the analysis of Laplacian eigenvalue, the frequency of SH can be assumed as $\sqrt{\ell(\ell+1)} \approx \ell$.

Laplace–Beltrami operator

- A generalization of the Laplace operator to functions defined on submanifolds in Euclidean space, more generally, on Riemannian manifolds.
- Named after Pierre-Simon Laplace and Eugenio Beltrami.
- The Laplace–Beltrami operator is the *divergence* of the gradient on a Riemannian manifold $\Delta f = \mathrm{div}(\nabla f)$.
- In spherical coordinates, when restricted to $\mathbb{S}^2$, it becomes the angular part of the full Laplacian.

Divergence (in vector calculus)

- In vector calculus, the *divergence* is a vector operator that operates on a vector field, producing a *scalar* field that indicates the rate at which the vector field alters the volume in an infinitesimal neighborhood of each point.
- In Cartesian coordinates*,

$$\mathrm{div}\mathbf{F} = \nabla \cdot \mathbf{F} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right) \cdot \left(F_x, F_y, F_z\right) = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}.$$

- In spherical coordinates,

$$\mathrm{div}\,\mathbf{F} = \nabla \cdot \mathbf{F} = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 F_r\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta\, F_\theta\right) + \frac{1}{r\sin\theta}\frac{\partial F_\varphi}{\partial \varphi}.$$

*Notice that it is invariant under rotations.

Laplacian eigenvalue of SH

- Laplacian eigenvalue $-\lambda$ satisfies $\Delta f = -\lambda f$.
- Previously, we have:

$$-\lambda = \frac{1}{Y\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial Y}{\partial\theta}\right) + \frac{1}{Y\sin^2\theta}\frac{\partial^2 Y}{\partial\varphi^2}$$

$$-\lambda Y = \frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial Y}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2 Y}{\partial\varphi^2}$$

$$-\lambda Y = \Delta_{\mathbb{S}^2} Y$$

where $-\lambda = -\ell(\ell+1)$ from the *Sturm–Liouville theory**.

* https://en.wikipedia.org/wiki/Sturm–Liouville_theory

Laplacian eigenvalue of Fourier series

- In the 1D Fourier series, the eigenvalue equation is:

$$\frac{d^2}{dx^2}e^{inx} = -n^2 e^{inx}$$

- Here, the eigenvalue is $-n^2$, and the frequency is n.
- The frequency is often the square root of its negative.

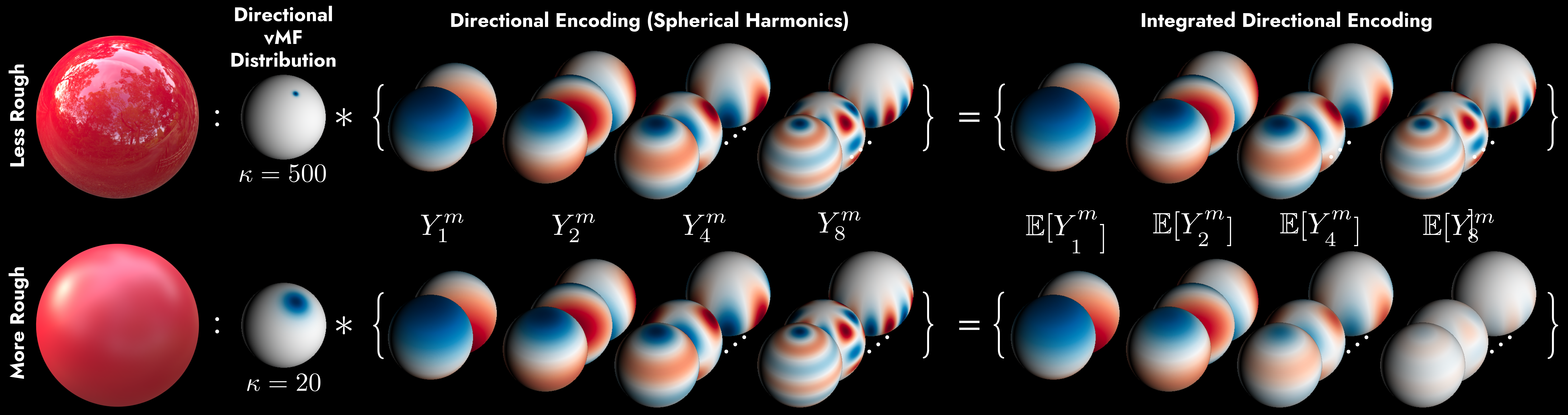
* https://en.wikipedia.org/wiki/Sturm–Liouville_theory

Recap

- The function of spherical harmonics takes a directional vector and gives 16 channels for the degree ℓ of 3.
- 3DGS assigns 16 learnable coefficients to each color component before performing linear combinations, whereas some NeRFs utilize an MLP for more complex cases.
- When we consider the roughness ρ, the output of the ℓ-th degree spherical harmonics is attenuated by multiplying $\exp\left(-\ell(\ell+1)\cdot\rho\right)$.
  - *The element with the higher degree (higher frequency) received the higher penalty.*
  - *The higher the roughness, the higher the penalty on the exponential scale.*

IDE Visualization

- Each component is a spherical harmonic convolved with the vMF distribution with κ.
- Less rough locations receive higher-frequency encodings (top), while rougher regions have attenuated high frequencies (down).



Reproduced Figure 3 from Verbin et al. (2021).

GT Ours

Our customized model with PSNR of 35.35 and SSIM of 0.984.

Thank you for your attention!

Any questions?