

Uncertainty-driven 3D Gaussian Splatting Active Mapping via Anisotropic Visibility Field

Supplementary Material

7. Anisotropic Visibility Field Details

7.1. Derivation of (8)

We represent $\nu(\mathbf{d}; \mathbf{d}_p)$ in the orthonormal basis of spherical harmonics $Y_\ell^m(\mathbf{d})$,

$$\begin{aligned} \nu(\mathbf{d}; \mathbf{d}_p) &:= \zeta \exp(\kappa \mathbf{d} \cdot \mathbf{d}_p) \\ &= \zeta \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_\ell^m(\mathbf{d}), \end{aligned} \quad (12)$$

and derive an analytical expression for the coefficients $a_{\ell m}$.

Since $\exp(\kappa \mathbf{d} \cdot \mathbf{d}_p)$ depends only on the dot-product $x = \mathbf{d} \cdot \mathbf{d}_p$, we first expand in the one-dimensional basis of Legendre polynomials $P_\ell(\mathbf{d} \cdot \mathbf{d}_p)$ [1]:

$$\exp(\kappa \mathbf{d} \cdot \mathbf{d}_p) = \sum_{\ell=0}^{\infty} b_\ell P_\ell(\mathbf{d} \cdot \mathbf{d}_p), \quad (13)$$

By comparing (13) with the Legendre expansion of the exponential [63]:

$$e^{\kappa x} = \sum_{\ell=0}^{\infty} (2\ell+1) i_\ell(\kappa) P_\ell(x).$$

Thus, we can identify the coefficients b_ℓ as

$$b_\ell = (2\ell+1) i_\ell(\kappa). \quad (14)$$

where i_ℓ is the modified spherical Bessel function of the first kind [63].

Then we apply the addition theorem for spherical harmonics [30]:

$$P_\ell(\mathbf{d} \cdot \mathbf{d}_p) = \frac{4\pi}{2\ell+1} \sum_{m=-\ell}^{\ell} Y_\ell^m(\mathbf{d}) Y_\ell^{m*}(\mathbf{d}_p). \quad (15)$$

where $Y_\ell^{m*}(\mathbf{d}_p)$ is the complex conjugate of $Y_\ell^m(\mathbf{d}_p)$.

Substituting (15) and (14) into (13) gives

$$\nu(\mathbf{d}; \mathbf{d}_p) = 4\pi\zeta \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} i_\ell(\kappa) Y_\ell^m(\mathbf{d}) Y_\ell^{m*}(\mathbf{d}_p). \quad (16)$$

Comparing with (7), we identify the coefficients

$$a_{\ell m} = 4\pi i_\ell(\kappa) Y_\ell^{m*}(\mathbf{d}_p). \quad (17)$$

where the modified spherical Bessel function of the first kind $i_\ell(\kappa)$ can be precomputed since κ is a fixed hyperparameter (set to $\kappa = 1$).

Note that $\nu(\mathbf{d}; \mathbf{d}_p) : \mathbb{S}^2 \rightarrow [0, 1]$ is real-valued, so in practice we expand it in a real spherical harmonics basis for implementation.

$$\nu(\mathbf{d}; \mathbf{d}_p) = 4\pi\zeta \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} i_\ell(\kappa) Y_{\ell m}(\mathbf{d}) Y_{\ell m}(\mathbf{d}_p). \quad (18)$$

where $Y_{\ell m}(\mathbf{d})$ is the real spherical harmonics function defined as

$$Y_{\ell m}(\mathbf{d}) = \begin{cases} \sqrt{2} (-1)^m \text{Im} [Y_\ell^{|m|}(\mathbf{d})], & m < 0 \\ Y_\ell^0(\mathbf{d}), & m = 0 \\ \sqrt{2} (-1)^m \text{Re} [Y_\ell^m(\mathbf{d})], & m > 0 \end{cases}$$

and it could be verified that (18) is equivalent to (16).

Therefore, we obtain a closed-form expression for the coefficients representing the directional visibility function $\nu(\mathbf{d}; \mathbf{d}_p)$ in spherical harmonics.

7.2. Details of SH Representation of $V^{(i)}(\mathbf{d})$

In this subsection, we provide further details on why computing the anisotropic visibility field $V^{(i)}(\mathbf{d})$ by directly combining (7) and (5) is impractical due to computational complexity.

We first rewrite (5) in a recursive form:

$$V_{\mathcal{P} \cup \{\mathbf{p}\}}^{(i)}(\mathbf{d}) = V_{\mathcal{P}}^{(i)}(\mathbf{d}) + V_{\mathbf{p}}^{(i)}(\mathbf{d}) - V_{\mathcal{P}}^{(i)}(\mathbf{d}) V_{\mathbf{p}}^{(i)}(\mathbf{d}) \quad (19)$$

starting from $V_{\emptyset}^{(i)}(\mathbf{d}) = 0$, and then recursively computing the anisotropic visibility field over the entire training set. Since spherical harmonics form a linear basis, the sum of two functions expressed in spherical harmonics can be obtained by simply adding their corresponding coefficients $a_{\ell m}$. However, their product poses significant challenges, particularly, the product of two spherical harmonics can be re-expanded as

$$\begin{aligned} Y_{\ell_1}^{m_1}(\mathbf{d}) Y_{\ell_2}^{m_2}(\mathbf{d}) &= \sum_{L=|\ell_1-\ell_2|}^{\ell_1+\ell_2} \sum_{M=-L}^L \sqrt{\frac{(2\ell_1+1)(2\ell_2+1)}{4\pi(2L+1)}} \\ &\quad \times C_{\ell_1 0 \ell_2 0}^{L 0} C_{\ell_1 m_1 \ell_2 m_2}^{L M} Y_L^M(\mathbf{d}). \end{aligned} \quad (20)$$

where $C_{\ell_1 m_1 \ell_2 m_2}^{L M}$ denote the Clebsch–Gordan coefficients [73, 99]. As is shown in the above equation, multiplying two spherical harmonics of degree L produces a

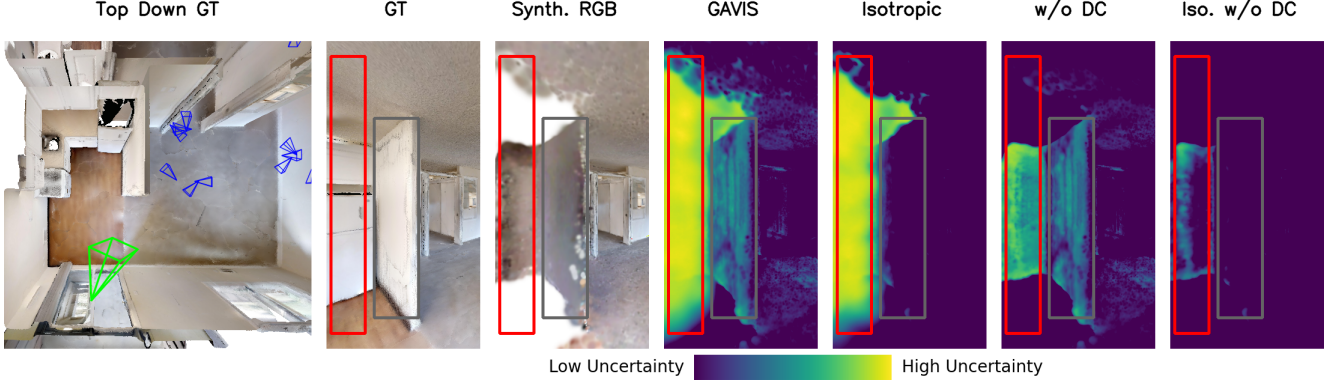


Figure 7. Illustration of the two major components of GAVIS: (1) anisotropic visibility (effects highlighted with gray boxes) and (2) visibility-field density control (effects highlighted with red boxes). From left to right: (i) top-down view of the scene with training views (blue frustums) covering only the right room and the queried view (green frustum) for uncertainty quantification; (ii) synthesized RGB image from a 3DGS model trained only on the partially covered region, leaving the rest of the scene unexplored; (iii) uncertainty estimated by full GAVIS; (iv) GAVIS with isotropic visibility; (v) GAVIS without visibility field density control; (vi) GAVIS with isotropic visibility and without density control. All methods are applied to the same learned 3DGS model.

spherical harmonics expansion of degree up to $2L$. Since the recursion in (5) requires performing the multiplication $|\mathcal{P}|$ times, the resulting degree can grow to $|\mathcal{P}|L$, requiring $O(|\mathcal{P}|^2 L^2)$ parameters to store. Since multiplying two spherical harmonics of degree L can be done in $O(L^3)$ time [53, 102], computing the anisotropic visibility field over the entire training set \mathcal{P} results in an overall complexity of $O(|\mathcal{P}|^4 L^3)$. Therefore, obtaining a closed-form expression for the spherical harmonics coefficients of the anisotropic visibility field is impractical.

7.3. Derivation of (9)

Starting from (5), we have

$$1 - V^{(i)}(\mathbf{d}) = \prod_{\mathbf{p} \in \mathcal{P}} (1 - V_{\mathbf{p}}^{(i)}(\mathbf{d})) \quad (21)$$

Applying the arithmetic-geometric mean (AM-GM) inequality to the right-hand side of (21) yields

$$|\mathcal{P}| \sqrt[|\mathcal{P}|]{\prod_{\mathbf{p} \in \mathcal{P}} (1 - V_{\mathbf{p}}^{(i)}(\mathbf{d}))} \leq \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} (1 - V_{\mathbf{p}}^{(i)}(\mathbf{d})) \quad (22)$$

Therefore, combining (21) and (22) yields the lower bound on $V^{(i)}(\mathbf{d})$.

$$V^{(i)}(\mathbf{d}) \geq 1 - \left(1 - \frac{\sum_{\mathbf{p} \in \mathcal{P}} V_{\mathbf{p}}^{(i)}(\mathbf{d})}{|\mathcal{P}|}\right)^{|\mathcal{P}|}. \quad (23)$$

which is equivalent to (9) since $\tilde{V}(\mathbf{d}) = \sum_{\mathbf{p} \in \mathcal{P}} V_{\mathbf{p}}^{(i)}(\mathbf{d})$.

This bound is tight when all $V_{\mathbf{p}}^{(i)}(\mathbf{d})$ are equal, i.e., when the visibility contributions from all camera views \mathbf{p} are the

same. In practice, this bound provides a computationally efficient way to estimate the overall visibility field $V^{(i)}(\mathbf{d})$ without having to compute the product of spherical harmonics of $V_{\mathbf{p}}^{(i)}(\mathbf{d})$ over all camera poses $\mathbf{p} \in \mathcal{P}$.

7.4. Details of Visibility Field Construction and Query

We provide additional details on constructing the anisotropic visibility field and on efficient querying. Firstly, we describe our efficient algorithm for computing the single-view (isotropic) particle visibility $\Phi_{i,\mathbf{p}} T_{\mathbf{p}}(t_i^{\mathbf{p}})$ for each particle i in training view \mathbf{p} in Alg. 2, which builds on a modified 3DGS rasterizer.

Note that the averaging step at L15 is necessary when 3D Gaussian particles span multiple pixels during rasterization. Since anisotropic visibility is modeled solely as a function of viewing direction, rather than the ray-particle intersection location, similar to view-dependent color in 3DGS [39], different rays from the same camera can interact with different points on the same particle, while sharing the same viewing direction. Therefore, averaging their transmittance contributions provides a consistent way to aggregate them into a single per-view value.

Next, we construct the anisotropic visibility field $V^{(i)}(\mathbf{d})$ by computing the SH coefficients $\gamma_{\ell m}^{(i)}$ for each particle i , accumulating contributions from all training views \mathcal{P} via (11) as outlined in Alg. 3. The single-view visibility terms $\Phi_{i,\mathbf{p}}, T_{\mathbf{p}}(t_i^{\mathbf{p}})$ are obtained using Alg. 2.

Finally, we describe the algorithm for querying the anisotropic visibility field at a direction \mathbf{d} in Alg. 4. Given the stored SH coefficients $\gamma_{\ell m}^{(g)}$ for each particle g , we evaluate the SH basis at the query direction \mathbf{d} to obtain $\hat{V}(\mathbf{d})$,

Algorithm 2 Single-view Particle Visibility

Input: $\mathbf{u}_g \in \mathbb{R}^2$: image-space projection of the Gaussian center to training view \mathbf{p} ; $\Sigma_g \in \mathbb{R}^{2 \times 2}$: image-space conic (precision) matrix; $z_g \in \mathbb{R}$: depth (for front-to-back sorting); $o_g \in [0, 1]$: base opacity; ϵ_T : transmittance cutoff (default 10^{-4});

Output: $\{v_g\}_{g=1}^N$: isotropic particle visibility with respect to training view \mathbf{p}

```
1: Initialize  $v_g \leftarrow 0$ ,  $c_g \leftarrow 0$  for all  $g$ 
2: for each pixel  $\mathbf{x}$  in the image do
3:    $T \leftarrow 1$ 
4:    $\mathcal{G} \leftarrow$  Gaussians overlapping  $\mathbf{x}$  (sorted by increasing  $z$ )
5:   for each  $g \in \mathcal{G}$  do
6:      $\alpha \leftarrow o_g \exp(-(\mathbf{u}_g - \mathbf{x})^\top \Sigma_g (\mathbf{u}_g - \mathbf{x}))$ 
7:      $v_g \leftarrow v_g + T$ ;  $c_g \leftarrow c_g + 1$ 
8:      $T \leftarrow T(1 - \alpha)$ 
9:     if  $T < \epsilon_T$  then break
10:    end if
11:  end for
12: end for
13: for  $g = 1$  to  $N$  do
14:   if  $c_g > 0$  then
15:     $v_g \leftarrow v_g / c_g$   $\triangleright$  Average when a particle overlaps multiple pixels.
16:   end if
17: end for
18: Return  $\{v_g\}_{g=1}^N$ 
```

Algorithm 3 Anisotropic Visibility Field Construction

Input: Training views \mathcal{P} with camera centers $\{\mathbf{x}_p\}$; 3DGS particles \mathcal{G} with centers $\{\mathbf{x}_g\}$; SH degree L ; concentration κ ; scale $\zeta = \exp(-\kappa)$

Output: Coefficients $\gamma \in \mathbb{C}^{|\mathcal{G}| \times (L+1)^2}$

```
1: Precompute  $\{i_\ell(\kappa)\}_{\ell=0}^L$   $\triangleright$  Modified spherical Bessel function of the first kind
2:  $a_\ell \leftarrow 4\pi \zeta i_\ell(\kappa)$  for  $\ell = 0, \dots, L$ 
3: Initialize  $\gamma_{\ell m}^{(g)} \leftarrow 0 \quad \forall g \in \mathcal{G}, 0 \leq \ell \leq L, -\ell \leq m \leq \ell$ 
4: for  $\mathbf{p} \in \mathcal{P}$  do  $\triangleright$  Accumulate per-view contributions
5:    $\mathbf{v} \leftarrow \text{SINGLEVIEWPARTICLEVISIBILITY}(\mathbf{p}, \mathcal{G})$ 
6:    $\mathbf{v}[g] = \Phi_{g, \mathbf{p}} T_{\mathbf{p}}(t_g^{\mathbf{p}})$  via Alg. 2
7:   for all  $g \in \mathcal{G}$  do
8:      $\mathbf{d}_p^{(g)} \leftarrow \frac{\mathbf{x}_g - \mathbf{x}_p}{\|\mathbf{x}_g - \mathbf{x}_p\|}$ 
9:      $\gamma_{\ell m}^{(g)} \leftarrow \gamma_{\ell m}^{(g)} + \mathbf{v}[g] \cdot a_\ell \cdot Y_{\ell m}(\mathbf{d}_p^{(g)})$  for  $\ell = 0:L$ ,  $m = -\ell:\ell$ 
10:  end for
11: return  $\{\gamma_{\ell m}^{(g)}\}$ 
```

Algorithm 4 Query Anisotropic Visibility Field

Input: SH coefficients $\{\gamma_{\ell m}^{(g)}\}$ for particle g ; query direction \mathbf{d} ; SH degree L ; number of training views $|\mathcal{P}|$

Output: Visibility $V^{(g)}(\mathbf{d})$ for particle g in direction \mathbf{d}

```
1:  $y[\ell, m] \leftarrow \text{SPHERICALHARMONICS}(\ell, m, \mathbf{d})$  for  $\ell = 0:L$ ,  $m = -\ell:\ell$ 
2:  $\tilde{V} \leftarrow \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} \gamma_{\ell m}^{(g)} y[\ell, m]$ 
3:  $V^{(g)} \leftarrow 1 - (1 - \frac{\tilde{V}}{|\mathcal{P}|})^{|\mathcal{P}|}$   $\triangleright$  AM-GM lower-bound estimator
4: return  $V^{(g)}$ 
```

and then compute the visibility $V^{(g)}(\mathbf{d})$ using the AM-GM lower-bound estimator in (9). Because we use $L = 2$ in practice, each particle requires only 9 SH coefficients, and the query cost is independent of the number of training views $|\mathcal{P}|$, enabling (per particle) constant-time querying.

Moreover, our method is significantly memory efficient. The anisotropic visibility field requires $\sim 28\%$ additional parameters over standard 3DGS, including SH coefficients and virtual particles (see next subsection). Consequently, our uncertainty quantification method incurs substantially lower memory overhead compared to other 3DGS-based approaches. For example, FisherRF [32] introduces 100% additional parameters to store per-parameter uncertainty, while VIMC [54] requires at least 200% overhead to represent the manifold of radiance field parameters.

7.5. Details of Visibility Field Density Control

We use *virtual particles* to distinguish free space from underexplored regions. Given a trained 3DGS, we uniformly sample virtual particles within the scene. Each particle is initialized with zero opacity, identity rotation, and an isotropic scale s . The total number of virtual particles is

$$N_{\text{vp}} = \rho \cdot V_{\text{scene}},$$

where V_{scene} is the scene volume and $\rho = 100$ is a density hyperparameter specifying the number of virtual particles per unit volume. The scale s is chosen such that the expected total volume of virtual particles per unit volume of the scene slightly exceeds 1, i.e.,

$$\frac{4}{3}\pi s^3 \rho = 1 + \eta,$$

with $\eta = 0.5$ ensuring that the total particle volume slightly exceeds the scene volume, which encourages broad coverage of the space. Note that s is substantially larger than the scale of the original 3DGS particles. This formulation allows a sparse set of virtual particles to approximate coverage of the entire scene, making it significantly more efficient than using a dense grid of particles.

We then combine virtual particles and original particles to compute the visibility of each virtual particle using the

same single-view visibility algorithm as for the original particles, with virtual particles treated as transparent during visibility computation. The multi-view visibility of each virtual particle is then computed as

$$1 - \prod_{p \in \mathcal{P}} (1 - \Phi_{i,p} T_p(t_i^p))$$

Finally, we prune virtual particles using a conservative visibility threshold $\epsilon_v = 0.95$. Particles with visibility greater than ϵ_v are treated as free space and discarded, while those with visibility below ϵ_v are retained as indicators of underexplored regions.

When querying the anisotropic visibility field for virtual particles, we treat all retained virtual particles as invisible in every direction, which is equivalent to setting $\gamma_{\ell m}^{(i)} = 0$ for all ℓ, m . For computational efficiency, however, we simply return $V^{(i)}(\mathbf{d}) = 0$ for all query directions \mathbf{d} .

Note that, unlike density control in 3DGS training [39], which is applied repeatedly throughout optimization, the visibility-field density control is a post-hoc procedure and needs to be executed only once after training. This makes it substantially faster in practice. Details of the algorithm are provided in Alg. 5.

Finally, an illustration of the two major components of GAVIS: (1) anisotropic visibility and (2) visibility-field density control, is provided in Fig. 7.

8. Details of Uncertainty-aware 3DGS Rasterization

In this section, we provide additional details on uncertainty-aware 3DGS rasterization, specifically how to quantify the uncertainty of the observed color along a ray direction \mathbf{d} using the anisotropic visibility field $\{V^{(g)}(\mathbf{d})\}_{g \in \mathcal{G}}$. This section extends the uncertainty-aware volume rendering formulation introduced in NVF [104], originally developed for NeRF, to 3DGS.

The rasterization process for 3DGS (along a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$) is expressed as:

$$\hat{C}(\mathbf{r}) = \sum_i w_i \mathbf{c}_i(\mathbf{d}), \quad (24)$$

where

$$w_i = \alpha_i \prod_{j=0}^{i-1} (1 - \alpha_j),$$

and $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$ is the opacity of the i -th particle, with σ_i its density and δ_i the interval length. In 3DGS, effective α_i is computed by evaluating a 2D Gaussian with covariance Σ_i at the pixel location, multiplied by the learned per-particle opacity o_i , where Σ_i is obtained by projecting the 3D Gaussian onto the image plane [39]. The emitted

Algorithm 5 Visibility Field Density Control

Input: Training views \mathcal{P} ; original 3DGS particles \mathcal{G} with centers $\{\mathbf{x}_g\}$, rotations $\{\mathbf{R}_g\}$, scales $\{s_g\}$, opacities $\{o_g\}$; scene volume $\mathbf{V}_{\text{scene}}$; virtual particle density ρ ; visibility threshold ϵ_v ; hyperparameter η (default 0.5)

Output: Augmented 3DGS $\tilde{\mathcal{G}} = \mathcal{G} \cup \mathcal{G}_{\text{virt}}^{\text{keep}}$ for uncertainty quantification

```

1:  $N_{\text{virt}} \leftarrow \lfloor \rho \cdot |\mathbf{V}_{\text{scene}}| \rfloor$   $\triangleright$  Number of virtual particles
2:  $s \leftarrow \sqrt[3]{\frac{3(1+\eta)}{4\pi\rho}}$   $\triangleright$  Scale of virtual particles
3: Sample virtual centers  $\{\mathbf{x}_j\}_{j=1}^{N_{\text{virt}}}$  uniformly in  $\mathbf{V}_{\text{scene}}$ 
4:  $\mathcal{G}_{\text{virt}} \leftarrow \{(\mathbf{x}_j, \mathbf{I}_3, s, 0)\}_{j=1}^{N_{\text{virt}}}$   $\triangleright$  Virtual particles: identity rotation, scale  $s$ , zero opacity
5:  $\mathcal{G}_{\text{all}} \leftarrow \mathcal{G} \cup \mathcal{G}_{\text{virt}}$ 
6:  $p_g \leftarrow 1 \quad \forall g \in \mathcal{G}_{\text{virt}}$   $\triangleright$  Accumulator for  $\prod_p (1 - \Phi_{g,p} T_p(t_g^p))$ 
7: for all  $p \in \mathcal{P}$  do  $\triangleright$  Accumulate per-view contributions
8:    $\mathbf{v}_p \leftarrow \text{SINGLEVIEWPARTICLEVISIBILITY}(p, \mathcal{G}_{\text{all}})$ 
    $\triangleright \mathbf{v}_p[g] = \Phi_{g,p} T_p(t_g^p)$  via Alg. 2
9:   for all  $g \in \mathcal{G}_{\text{virt}}$  do
10:     $p_g \leftarrow p_g \cdot (1 - \mathbf{v}_p[g])$ 
11:   end for
12: end for
13:  $\mathcal{G}_{\text{virt}}^{\text{keep}} \leftarrow \emptyset$ 
14: for all  $g \in \mathcal{G}_{\text{virt}}$  do
15:    $\tilde{v}_g \leftarrow 1 - p_g$   $\triangleright$  Multi-view visibility
    $1 - \prod_{p \in \mathcal{P}} (1 - \Phi_{g,p} T_p(t_g^p))$ 
16:   if  $\tilde{v}_g \leq \epsilon_v$  then  $\triangleright$  Low visibility  $\Rightarrow$  underexplored region
17:      $\mathcal{G}_{\text{virt}}^{\text{keep}} \leftarrow \mathcal{G}_{\text{virt}}^{\text{keep}} \cup \{(\mathbf{x}_g, \mathbf{I}_3, s, 0)\}$ 
18:   else  $\triangleright$  High visibility  $\Rightarrow$  free space, discard
19:     continue
20:   end if
21: end for
22:  $\tilde{\mathcal{G}} \leftarrow \mathcal{G} \cup \mathcal{G}_{\text{virt}}^{\text{keep}}$ 
23: return  $\tilde{\mathcal{G}}$ 

```

color $\mathbf{c}_i(\mathbf{d})$ is obtained from the spherical harmonics coefficients of particle i , evaluated at direction \mathbf{d} .

To extend (24) to uncertainty-aware rasterization, consider computing the posterior distribution of the observed color $\hat{C}(\mathbf{r})$ along direction \mathbf{d} , denoted as $p(\mathbf{z}_0)$. Each emitted color is modeled as a Gaussian random variable $\mathbf{c}_i(\mathbf{d}) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{c}_i}, \mathbf{Q}_{\mathbf{c}_i})$, where $\boldsymbol{\mu}_{\mathbf{c}_i}$ and $\mathbf{Q}_{\mathbf{c}_i}$ are respectively the mean and covariance. Since density σ_i can be interpreted as the differential probability that a ray terminates at position i [61], the effective opacity α_i represents the probability of ray terminating at that position. The rasterization process can thus be expressed as a Bayesian Network [104]

via the recursion

$$p(\mathbf{z}_i) = \alpha_i p(\mathbf{c}_i) + (1 - \alpha_i) p(\mathbf{z}_{i+1}), \quad (25)$$

where $p(\mathbf{z}_i)$ denotes the distribution of the observed color if the camera were placed at position i along the ray. Equation (25) states that with probability α_i a ray terminates at particle i , in which case the observed color is \mathbf{c}_i ; otherwise, with probability $1 - \alpha_i$, the observed color is inherited from the next position along the ray. The recursion terminates at the last particle, whose emitted color distribution is Gaussian.

By applying the recursive expression in (25) from the last particle to the first, we can compute the posterior of the observed color at the camera position \mathbf{z}_0 , which is the output of the rasterization process.

$$p(\mathbf{z}_0) = \sum_i w_i \mathcal{N}(\boldsymbol{\mu}_{\mathbf{c}_i}, \mathbf{Q}_{\mathbf{c}_i}), \quad (26)$$

which is a Gaussian mixture model (GMM) with N (number of particles along the ray) components.

As noted in NVF [104], directly quantifying uncertainty from the GMM in (26) is limited because the predicted color variance $\mathbf{Q}_{\mathbf{c}_i}$ is often inaccurate due to approximations in the uncertainty estimation process. To address this issue, NVF [104] introduces a visibility-modulated variance correction which ensures particles with lower visibility exhibit higher uncertainty.

Let V_i be a binary random variable indicating whether particle i is visible along the ray, with visibility probability given by the visibility field, $P(V_i = 1) = V^{(i)}(\mathbf{d})$. The distribution of the emitted color is therefore modified to

$$p(\mathbf{c}_i | V_i) = \begin{cases} \mathcal{N}(\boldsymbol{\mu}_{\mathbf{c}_i}, \mathbf{Q}_{\mathbf{c}_i}), & \text{if } V_i = 1, \\ \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{Q}_0), & \text{otherwise,} \end{cases} \quad (27)$$

where $\mathcal{N}(\boldsymbol{\mu}_0, \mathbf{Q}_0)$ is a prior with large variance \mathbf{Q}_0 to reflect the high uncertainty of invisible particles.

Opacity is also visibility-modulated, since opacity predictions are unreliable for particles that are rarely or never observed. Similar to [104], the compensated opacity for 3DGS is

$$\alpha_i^* = (v_i + \beta(1 - v_i)) \alpha_i + o_0(1 - \beta)(1 - v_i), \quad (28)$$

where $v_i := V^{(i)}(\mathbf{d})$ for shorthand, o_0 is the prior opacity for invisible particles, and β is a hyperparameter representing the reliability of the opacity prediction for invisible regions. The first term in (28) captures the case where the particle is visible or the opacity prediction is reliable. The second term introduces a correction when the particle is invisible and the predicted opacity is unreliable, by substituting the predicted opacity in place of a prior opacity o_0 . Note that here we use the per-particle opacity o_0 instead of

the effective opacity that incorporates the Gaussian decay as a function of the distance between the particle center and the pixel location.

Therefore, the visibility-modulated uncertainty-aware 3DGS rasterization process can be expressed as the following Gaussian mixture model:

$$p(\mathbf{z}_0) = \sum_i w_i^* v_i \mathcal{N}(\boldsymbol{\mu}_{\mathbf{c}_i}, \mathbf{Q}_{\mathbf{c}_i}) + \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{Q}_0) \sum_i w_i^* (1 - v_i), \quad (29)$$

where $w_i^* = \alpha_i^* \prod_{j=1}^{i-1} (1 - \alpha_j^*)$ denotes the visibility-compensated weights.

The entropy of the GMM in (29) is then used to quantify the uncertainty of the observed color \mathbf{z}_0 . Because the entropy of a Gaussian mixture lacks a closed-form expression, the following upper bound [28], also adopted in NVF [104], is used as an accurate estimator of the entropy:

$$\mathcal{H}(\mathbf{z}_0) \leq \sum_{i=0}^N \bar{w}_i \left(-\log(\bar{w}_i) + \frac{1}{2} \log((2\pi e)^D |\mathbf{Q}_i|) \right), \quad (30)$$

where we denote $\bar{w}_i := w_i^* v_i$ for $i \geq 1$, and $\bar{w}_0 := \sum_{i=1}^N w_i^* (1 - v_i)$, and $\mathbf{Q}_i := \mathbf{Q}_{\mathbf{c}_i}$ for $i \geq 1$ and \mathbf{Q}_0 for $i = 0$. Here, $D = 3$ corresponds to the RGB color channels.

It is worth noting that the visibility field v_i is estimated using the lower bound in (23). This approximation underestimates visibility and therefore overestimates how much of the mixture should be attributed to invisible regions. Consequently, the RHS in (30) remains a valid upper bound on the entropy computed with exact visibility, since the inequality is preserved under the assumption $\mathbf{Q}_0 \succeq \mathbf{Q}_{\mathbf{c}_i}$ for all i . This condition is enforced by construction, as \mathbf{Q}_0 is chosen to be a large-variance prior.

As demonstrated in [104], when using visibility-modulated uncertainty-aware volume rendering, setting $\mathbf{Q}_{\mathbf{c}_i}$ to a constant already yields strong performance in active mapping, since the visibility field effectively captures uncertainty in unseen regions, which is the dominant factor for active mapping. Incorporating predicted variances $\mathbf{Q}_{\mathbf{c}_i}$ yields only marginal performance gains.

Motivated by this observation, our base GAVIS model adopts a constant variance $\mathbf{Q}_{\mathbf{c}_i} = \sigma_c^2 \mathbf{I}$, which offers a favorable trade-off between performance and computational cost. In addition, existing uncertainty quantification methods for 3DGS, including FisherRF [32] and VIMC [54], could be used to predict $\mathbf{Q}_{\mathbf{c}_i}$, leading to two variants of the base GAVIS model: (1) FisherRF+GAVIS and (2) VIMC+GAVIS. Further details on how these methods are used to obtain $\mathbf{Q}_{\mathbf{c}_i}$ are provided in the Sec. 10, and quantitative experiments of the two variants are included in the Sec. 12, demonstrating that GAVIS can serve as a post-hoc module that significantly improves the performance of existing methods.

Finally, we describe the algorithm for 3DGS uncertainty quantification using the visibility field at a synthesis view \mathbf{p} in Alg. 6. This procedure is efficiently implemented using a modified 3DGS rasterizer, enabling real-time uncertainty quantification at around 200 FPS.

Algorithm 6 3DGS Uncertainty Quantification with Visibility Field

Input: $\mathbf{u}_g \in \mathbb{R}^2$: image-space projection of the Gaussian center to synthesis view \mathbf{p} ; $\Sigma_g \in \mathbb{R}^{2 \times 2}$: image-space conic (precision) matrix; $z_g \in \mathbb{R}$: depth (for front-to-back sorting); $o_g \in [0, 1]$: base opacity; $v_g \in [0, 1]$: queried visibility field value for Gaussian g along direction \mathbf{d}_p ; \mathbf{Q}_{c_g} : predicted color covariance for Gaussian g along direction \mathbf{d}_p ; ϵ_T : transmittance cutoff (default 10^{-4}); β, o_0, \mathbf{Q}_0 : hyperparameters for visibility compensation (default $\beta = 0.5, o_0 = 0.15, \mathbf{Q}_0 = \mathbf{I}_3$).

Output: Entropy map $\mathcal{H} \in \mathbb{R}^{H \times W}$ for synthesis view \mathbf{p} .

```

1: Initialize  $\mathbf{H}[\mathbf{x}] \leftarrow 0$  for all pixels  $\mathbf{x}$  in the image
2: for each pixel  $\mathbf{x}$  in the image do
3:    $T \leftarrow 1$  ▷ current transmittance
4:    $\bar{w}_0 \leftarrow 0$  ▷ accumulated weight of the invisible component
5:    $\mathcal{G} \leftarrow$  Gaussians overlapping  $\mathbf{x}$  (sorted by increasing  $z$ )
6:   for each  $g \in \mathcal{G}$  do
7:      $\alpha \leftarrow o_g \exp(-(\mathbf{u}_g - \mathbf{x})^\top \Sigma_g (\mathbf{u}_g - \mathbf{x}))$ 
8:      $\alpha^* \leftarrow (v_g + \beta(1 - v_g)) \alpha + o_0(1 - \beta)(1 - v_g)$ 
9:      $\triangleright$  Eq. (28)
10:     $w^* \leftarrow \alpha^* T$ 
11:     $h \leftarrow -\log(w^* v_g) + \frac{1}{2} \log |\mathbf{Q}_{c_g}| + \frac{3}{2} \log(2\pi e)$ 
12:     $\mathbf{H}[\mathbf{x}] \leftarrow \mathbf{H}[\mathbf{x}] + w^* v_g h$  ▷ Eq. (30)
13:     $\bar{w}_0 \leftarrow \bar{w}_0 + w^*(1 - v_g)$ 
14:     $T \leftarrow T(1 - \alpha^*)$ 
15:    if  $T < \epsilon_T$  then break
16:    end if
17:  end for
18:   $h \leftarrow -\log \bar{w}_0 + \frac{1}{2} \log |\mathbf{Q}_0| + \frac{3}{2} \log(2\pi e)$ 
19:   $\mathbf{H}[\mathbf{x}] \leftarrow \mathbf{H}[\mathbf{x}] + \bar{w}_0 h$  ▷ Add the contribution of the invisible component
20: end for
21: Return  $\mathbf{H}$ 

```

9. Active Mapping Details

In this section, we provide additional details on our active mapping method, which follows a formulation similar to that in [104].

The goal of active mapping or next-best-view planning is to select the camera pose \mathbf{p} that maximizes the expected information gain (EIG) about the scene, which can be written

as:

$$I(\mathbf{Z}; \boldsymbol{\theta} | \mathbf{p}) = \mathcal{H}[\mathbf{Z} | \mathbf{p}] - \mathcal{H}[\mathbf{Z}_p | \boldsymbol{\theta}, \mathbf{p}], \quad (31)$$

where $I(\cdot)$ denotes mutual information, $\mathcal{H}[\cdot]$ is entropy, \mathbf{Z} is the random variable corresponding to the rendered observation, and $\boldsymbol{\theta}$ represents the radiance-field parameters.

We assume a Gaussian observation model, $p(\mathbf{Z} | \mathbf{p}, \boldsymbol{\theta}) = \mathcal{N}(f(\mathbf{p}; \boldsymbol{\theta}), \Sigma_z)$, where $f(\mathbf{p}; \boldsymbol{\theta})$ is the rendering function and Σ_z is the observation noise covariance. Assuming Σ_z is constant and isotropic, i.e., $\Sigma_z = \sigma_z^2 \mathbf{I}$ then the entropy term $\mathcal{H}[\mathbf{Z} | \boldsymbol{\theta}, \mathbf{p}]$ becomes constant and can be omitted during optimization. Hence, maximizing EIG reduces to maximizing the entropy:

$$\mathbf{p}^* = \arg \max_{\mathbf{p}} \mathcal{H}(\mathbf{Z}_p), \quad (32)$$

where we use $\mathcal{H}(\mathbf{Z}_p)$ as shorthand for $\mathcal{H}[\mathbf{Z}_p | \mathbf{p}]$, which could be obtained from the posterior of the rendered observation. This objective is commonly adopted in radiance-field-based active mapping [46, 54, 104, 107]. Note that uncertainties arising from robot localization or dynamics are not modeled here, as the primary focus of these methods is the uncertainty of the radiance field itself. Moreover, the action τ is restricted to the next camera pose, and the EIG of an entire trajectory is not considered, since efficiently computing the joint entropy over a sequence of observations remains an open challenge [40, 67]. A simplified workaround assumes independence across views [31, 92], though this approximation is not the focus of this work. Extending the GAVIS framework to support trajectory-level EIG is a promising direction for future research, but it lies beyond the scope of this paper.

We adopt the spatial correlation correction term from NVF [104] to recover the joint image-level entropy from per-pixel entropies:

$$\mathcal{H}(\mathbf{Z}_p) = \sum_{m,n} \left(\mathcal{H}(\mathbf{Z}_p^{mn}) - f_{\text{corr}}(\mathcal{H}(\mathbf{Z}_p^{mn}); d_p^{mn}) \right), \quad (33)$$

where \mathbf{Z}_p^{mn} denotes the observed color associated with pixel (m, n) , d_p^{mn} is its expected depth, and $f_{\text{corr}}(\cdot)$ is a depth-dependent correction term that compensates for spatial correlations between neighboring pixels. Additional details can be found in [104].

Finally, we describe the active mapping algorithm with GAVIS in Alg. 7. The algorithm iteratively trains the 3DGS model, constructs the visibility field, and selects the next-best view based on the expected information gain computed from the visibility-aware 3DGS uncertainty quantification.

10. Applying GAVIS as a Post-hoc Module

In this section, we provide additional details on applying GAVIS as a post-hoc module to enhance existing uncertainty quantification methods, including FisherRF [32] and

Algorithm 7 Active Mapping with GAVIS

Input: \mathcal{P} : initial poses; \mathcal{Z} : initial images; T : number of active mapping planning steps.

Output: \mathcal{G} : the trained 3DGS model after active mapping.

```

1: for  $i = 1$  to  $T$  do
2:    $\mathcal{G} \leftarrow \text{train3DGS}(\mathcal{P}, \mathcal{Z})$  ▷ train 3DGS
3:    $\{\gamma_{\ell m}^{(g)}\} \leftarrow \text{VFConstruction}(\mathcal{G}, \mathcal{P})$  ▷ Alg. 3
4:    $\tilde{\mathcal{G}} \leftarrow \text{VFDensityControl}(\mathcal{G}, \mathcal{P})$  ▷ Alg. 5
5:    $\mathcal{P}_c \leftarrow \text{samplePoses}(\mathcal{G})$  ▷ sample candidate poses
6:   for  $p$  in  $\mathcal{P}_c$  do
7:      $v_g \leftarrow \text{VFQuery}(\tilde{\mathcal{G}}, \{\gamma_{\ell m}^{(g)}\}, p)$  ▷ Alg. 4
8:      $\mathbf{Z}_p \leftarrow \text{UQVF}(\tilde{\mathcal{G}}, v_g, p)$  ▷ Alg. 6
9:   end for
10:   $p_i \leftarrow \arg \max_{p \in \mathcal{P}_c} \mathcal{H}(\mathbf{Z}_p)$ 
11:   $\mathcal{P} \leftarrow \{p_i\} \cup \mathcal{P}$ 
12:   $\mathcal{Z} \leftarrow \text{takeImageAt}(\{p_i\}) \cup \mathcal{Z}$  ▷ update training set
13: end for
14: return  $\mathcal{G}$ 

```

VIMC [54]. The key idea is to use the uncertainty estimated by FisherRF or VIMC to refine the variance of emitted color \mathbf{Q}_{c_i} in (3) along ray direction \mathbf{d} , whereas in base GAVIS this quantity is treated as a constant hyperparameter for all particles. We first describe how to integrate GAVIS with FisherRF and VIMC, and then discuss the relationship between GAVIS and these methods.

10.1. Integration with FisherRF

FisherRF [32] approximates the expected information gain of a candidate camera pose by maximizing

$$\arg \max_p \text{tr}(\mathbf{H}''[\mathbf{Z}_p | p, \theta^*] \mathbf{H}''[\theta^* | D^{\text{train}}]^{-1}), \quad (34)$$

where the observation \mathbf{Z}_p is the rendered image for pose p , θ^* are the estimated 3DGS parameters, and the training set is $D^{\text{train}} := \{(p, \mathbf{Z}_p)\}_{p \in \mathcal{P}}$. The observed information is given by:

$$\mathbf{H}''[\theta^* | D^{\text{train}}] = \sum_{(p, \mathbf{Z}_p) \in D^{\text{train}}} \mathbf{H}''[\mathbf{Z}_p | p, \theta^*], \quad (35)$$

where $\mathbf{H}''[\mathbf{Z}_p | p, \theta^*]$ is the Hessian of the negative log-likelihood under the Gaussian observation model

$$-\log p(\mathbf{Z}_p | p, \theta) = \frac{1}{2\sigma_z^2} (\mathbf{Z}_p - f(p; \theta))^T (\mathbf{Z}_p - f(p; \theta)) + C, \quad (36)$$

with rendering model $f(p; \theta)$ and observation noise variance σ_z^2 . FisherRF adopts the Laplace approximation [40], leading to:

$$\mathbf{H}''[\mathbf{Z} | p, \theta^*] \approx \frac{1}{\sigma_z^2} \text{diag}(\nabla_{\theta} f(p; \theta)^T \nabla_{\theta} f(p; \theta)) + \lambda \mathbf{I} \quad (37)$$

Thus the posterior covariance of θ is given by:

$$\Sigma_{\theta} \approx \mathbf{H}''[\theta^* | D^{\text{train}}]^{-1} \quad (38)$$

Further details on these derivations can be found in [23, 32, 40].

We then propagate this parameter uncertainty to the emitted color of each particle. Let $\theta_{\ell, m, k}^{(i)}$ denote the color coefficient of the i -th particle in channel k , parameterized in the spherical harmonics basis. The emitted color in channel k along direction \mathbf{d} is given by

$$c_{i, k}(\mathbf{d}) = \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} \theta_{\ell, m, k}^{(i)} Y_{\ell, m}(\mathbf{d}), \quad (39)$$

which is linear in the coefficients $\theta_{\ell, m, k}^{(i)}$. The variance of the emitted color in channel k is obtained by linear uncertainty propagation:

$$Q_{c_{i, k}}(\mathbf{d}) = \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} Y_{\ell, m}(\mathbf{d})^2 \Sigma_{\ell, m, k}^{(i)}, \quad (40)$$

where $\Sigma_{\ell, m, k}^{(i)}$ is the variance of the coefficient $\theta_{\ell, m, k}^{(i)}$ extracted from the diagonal of Σ_{θ} , as in FisherRF [32] all parameters are assumed independent and the covariance matrix Σ_{θ} is therefore diagonal. Since each color channel is also modeled independently, the covariance of the emitted color vector \mathbf{Q}_{c_i} is diagonal with diagonal entries $Q_{c_{i, k}}$.

10.2. Integration with VIMC

VIMC [54] estimates the distribution of θ from a learned low-dimensional manifold of the radiance field parameters. We draw J parameter samples (with $J = 2$ by default in VIMC). For each sampled parameter θ_j , we compute the emitted color $c_{i, k, j}(\mathbf{d})$ following (39), and estimate the per-channel uncertainty by taking the sample variance across the J samples:

$$Q_{c_{i, k}}(\mathbf{d}) = \frac{1}{J-1} \sum_{j=1}^J (c_{i, k, j}(\mathbf{d}) - \bar{c}_{i, k}(\mathbf{d}))^2, \quad (41)$$

where $\bar{c}_{i, k}(\mathbf{d}) = \frac{1}{J} \sum_{j=1}^J c_{i, k, j}(\mathbf{d})$ is the sample mean. Under the assumption that the three color channels are independent, the resulting emitted-color covariance \mathbf{Q}_{c_i} similarly reduces to a diagonal matrix whose diagonal entries are given by $Q_{c_{i, k}}(\mathbf{d})$.

10.3. Discussion

In principle, an accurate radiance-field uncertainty estimation method should assign high uncertainty to unseen regions, as these areas do not contribute to the training loss. However, existing approaches struggle to capture this behavior.

In FisherRF, the diagonal Laplace approximation considers only the diagonal entries of the Hessian. This ignores the strong correlations among parameters of the same Gaussian particle (e.g., RGB spherical-harmonics coefficients within a color channel), causing the approximation to underestimate uncertainty, especially for viewing directions far from the training views.

In VIMC, uncertainty is approximated from Monte Carlo samples drawn from a learned low-dimensional manifold. With only a small number of samples, variance estimates are noisy and unreliable, while increasing the sample count significantly raises computational cost, making it unsuitable for real-time applications.

Integrating GAVIS provides an efficient way to compensate for these limitations. Since GAVIS reliably assigns higher uncertainty to unseen regions, it improves the accuracy of uncertainty quantification and leads to better active-mapping performance (see Sec. 12).

11. Experimental Details

11.1. Environment Setup

We evaluate our method on four datasets: (1) the standard NeRF Synthetic dataset [57]; (2) a space dataset consisting of the Hubble Space Telescope (HST) [3] and the International Space Station (ISS) [4] for space robotics scenarios; (3) eight indoor environments from the Habitat-Matterport 3D (HM3D) [70] dataset

- 00208-SQqGpSHzfSr
- 00299-bdp1XNEdvmW
- 00321-JWWJBQWHv64
- 00323-yHLr6bvWsVm
- 00415-rBmEe6ab5VP
- 00441-4MRLuLyET6a
- 00446-tL6i2PtktSh
- 00670-mDdyQ6azhVD

and (4) eight scenes from the Gibson [101] dataset (annawan, bremerton, creede, eagerville, eastville, helix, hometown, quantico). NeRF Synthetic and the space dataset are simulated in Blender, whereas HM3D and Gibson scenes are simulated in Habitat-Sim [74]. All images are rendered at a resolution of 512×512 pixels, with both horizontal and vertical fields of view set to 90° . For a fair comparison, all methods are initialized from the same set of closely spaced initial views that cover only a portion of the scene, mimicking the starting condition of a robotic active mapping process. The agent begins with 3 initial views for NeRF Synthetic and the space dataset, and with 5 spherically sampled views from a fixed starting position for HM3D and Gibson. We use 10 active-mapping steps for NeRF Synthetic and the space dataset, 40 steps for Gibson, and 80 steps for HM3D, chosen as the minimal steps at which the strongest methods achieve ac-

ceptable reconstruction quality (PSNR > 24). Each scene is evaluated over three runs with different random seeds, and the mean and standard deviation of all metrics are reported. All experiments for each scene and each method are performed on a single NVIDIA A40 GPU.

11.2. Active Mapping Setup

To ensure a fair comparison of uncertainty quantification methods for active mapping, all methods are evaluated using the same active mapping pipeline, differing only in the uncertainty quantification module, following the setup in NVF [104]. In particular, each method uses the same minimally biased candidate view sampler, which samples only collision-free candidate poses without additional heuristics. This design reflects the principle that an optimal active mapping policy should, in principle, be achievable by relying solely on accurate uncertainty quantification, without the need for heuristic guidance. For the NeRF Synthetic and space datasets, candidate poses are sampled from SE(3) within the scene bounding box. For the Gibson and HM3D datasets, candidate poses are sampled from SE(2) on a plane at positions known to be collision-free, with a fixed height of 1.5 m, mimicking the active mapping process of a mobile robot.

11.3. Training Details

For a fair comparison, we adopt the standard 3DGS training setup from [39] with minimal differences across all 3DGS-based methods. Since FisherRF and GAVIS are post-hoc methods, they share the exact same 3DGS training setup. VIMC uses the same setup as well, with the only difference being an additional loss term used to learn the low-dimensional manifold in the space of 3DGS parameters for uncertainty quantification. NVF is trained following the same configuration as in [104]. For the NeRF Synthetic and space datasets, 3DGS is trained for 3500 iterations at each planning step. For the Gibson and HM3D indoor datasets, 3DGS is trained for 7500 iterations at each planning step, and depth supervision from [55] is incorporated to accelerate convergence for all 3DGS-based methods, using simulator-rendered depth images. For a fair comparison, a similar depth loss [17] is also applied when training NVF. For shorter-horizon datasets (NeRF Synthetic, space, and Gibson), 3DGS is initialized using a subsampled sparse point cloud constructed from all previously observed depth images, to reduce training time. For the longer-horizon HM3D dataset, 3DGS at each planning step is initialized by merging the previous 3DGS model with a subsampled sparse point cloud derived from the most recent depth image.

11.4. Metrics Details

Visual Coverage. For visual coverage evaluation, we adopt the ground-truth visibility (Vis) metric from [104]. This metric assigns each face in the ground-truth mesh a binary visibility value, all faces are initialized with value 0, and then the value of a face is set to 1 (visible) if it is observed by the training views without occlusion. The Vis score is computed as the ratio of the total area of visible faces to the total mesh surface area.

We also visualize the GT visibility map (used in Figs. 6, 9, and 8) by rendering the visibility-annotated mesh from each candidate query pose. Similar to the uncertainty map, darker regions indicate visible areas (value 1), while brighter regions indicate invisible areas (value 0). Ideally, an uncertainty quantification method tailored for active mapping should produce uncertainty maps that correlate with the GT visibility map, assigning high uncertainty (bright) to invisible regions and low uncertainty (dark) to visible regions.

We note that ground-truth (GT) visibility is inherently an isotropic metric that does not account for the view-dependent nature of visibility. In particular, observing a face from a single direction, regardless of viewing distance, may already be considered fully visible. As a result, such metrics do not favor uncertainty quantification methods that encourage revisiting regions from diverse viewpoints. Designing evaluation metrics for visual coverage that account for viewpoint diversity and better align with the active mapping’s objective of acquiring observations from diverse directions remains an open problem.

Mesh Metrics. To assess reconstruction completeness, we also include the completion ratio defined in [88]. Ground-truth 3D points are sampled from the original scene meshes. Predicted points are extracted from a trained 3DGS model using the approach described in [2]. For consistency with prior work [104], we use a completion ratio threshold of 0.01 for the NeRF Synthetic and space datasets. For HM3D and Gibson environments, we adopt a threshold of 0.05.

We additionally report mesh-based metrics, including completion (Comp) and accuracy (Acc). However, we adopt completion ratio (CR) as the primary mesh metric, as it is widely used, easier to interpret, and more robust to outliers compared to Comp [33, 50]. Moreover, Acc is not well aligned with the objective of active mapping under 3DGS, despite its use in NeRF-based active mapping [104]. This is because 3DGS cannot distinguish empty space from under-explored regions, causing reconstructed point clouds to reflect only explored areas. As a result, Acc is biased toward policies that oversample a limited region while penalizing broader exploration in 3DGS active mapping. Empirically, Acc shows minimal improvement with additional views and exhibits no statistically significant differences across methods; accordingly, it is rarely used in 3DGS-based active

mapping [32, 54]. As shown in Tab. 6, Acc varies little across methods and shows weak correlation with other metrics.

Uncertainty Quantification. To quantitatively evaluate the quality of uncertainty quantification, we adopt the Area Under the Sparsification Error curve (AUSE) metric [29, 68], which measures the correlation between predicted uncertainty and actual error. However, the depth-based AUSE metric (AUSE-D) used in [23, 32, 54], which evaluates the correlation between predicted uncertainty and depth error computed from the predicted depth and the ground-truth mesh, can be misaligned with the objective of active mapping. In particular, radiance fields can produce accurate depth predictions in unobserved regions via plausible imagination. As a result, generic uncertainty quantification methods, which are not designed for active mapping and tend to assign low uncertainty to these regions, can still achieve strong AUSE-D scores. In contrast, effective active mapping requires high uncertainty in unobserved regions to encourage exploration. Therefore, a strong AUSE-D score does not necessarily indicate good active mapping performance.

We note that alternative metrics for evaluating uncertainty quantification exist, including negative log-likelihood (NLL) and expected calibration error (ECE) [25]. However, these metrics are less commonly used in the context of radiance fields, since they emphasize the quality of a fully specified and calibrated predictive distribution, which is not necessary for downstream tasks such as active mapping and artifact removal [23]. Moreover, it is unclear how to fairly compare these metrics across different methods, for example, GMM-based approaches (GAVIS&NVF), methods assuming a Gaussian distribution (VIMC), and methods that do not explicitly predict a distribution (FisherRF). Furthermore, similar to AUSE-D, these metrics can also be misaligned with the objective of active mapping.

Therefore, designing uncertainty quantification metrics tailored to active mapping remains an open problem. Toward this direction, we introduce a ground-truth visibility-based variant, denoted as AUSE-V, which measures the correlation between the predicted uncertainty map and the ground-truth visibility map. The ground-truth visibility map is obtained by assigning binary face-level values on the ground-truth mesh and rendering them using a rasterizer. AUSE-V better captures the requirement of active mapping for high uncertainty in unobserved regions, and thus exhibits stronger correlation with both active mapping performance and qualitative behavior. Further results are provided in Sec. 12.1.

However, as discussed above, GT visibility is inherently isotropic and does not account for the view-dependent nature of visibility, so it may not fully capture the benefits of anisotropic visibility modeling. Moreover, a reliable un-

certainty quantification method for active mapping should assign high uncertainty to regions with low GT visibility, but need not always assign low uncertainty to regions with high GT visibility. High uncertainty in observed regions may arise when the test view direction deviates substantially from the training views direction or when model capacity is limited. As a result, predictions can still be inaccurate even in observed regions. Therefore, although AUSE-V is more informative than AUSE-D, it may not fully capture uncertainty quality for active mapping.

12. Additional Results

12.1. Additional Quantitative Results

We present the complete active mapping results for all methods, including the baselines FisherRF, VIMC, and NVF, as well as our proposed methods GAVIS, FisherRF+GAVIS, and VIMC+GAVIS, with all evaluation metrics reported for each dataset in Tab. 6, with additional Comp and Acc metrics included.

As shown in Tab. 6, our method significantly outperforms all 3DGS baselines in all metrics excluding Acc. Moreover, integrating GAVIS as a post-hoc module consistently yields substantial improvements over each corresponding baseline across all datasets and all metrics excluding Acc. Our method also surpasses NVF in active mapping performance while requiring substantially less computation time. GAVIS outperforms NVF on all image-based metrics (PSNR, SSIM, and LPIPS) across all datasets. On the primary mesh metric (CR) and visibility metric (VIS), GAVIS outperforms NVF on all datasets except Gibson. Since these metrics are inherently isotropic and direction agnostic, e.g., observing a region from a single direction may already provide high GT mesh visibility and low point-cloud error (especially when depth loss is used), the performance gains are less pronounced on these metrics.

Among the variants of our method (GAVIS, FisherRF+GAVIS, and VIMC+GAVIS), FisherRF+GAVIS overall outperforms the base GAVIS model on most datasets except the space dataset. This suggests that more accurately modeling Q_{c_i} can further improve performance. The advantage is less pronounced in the space dataset, since color variance tends to be more monotonic across these scenes. VIMC+GAVIS outperforms the base GAVIS model only on the NeRF Synthetic dataset and not on the others. This is because VIMC relies on sampling-based uncertainty estimation, and with only 2 samples (the default setting in [54]), the sampling noise tends to overshadow the benefits introduced by visibility modeling.

Although performance gains can be achieved by more accurately modeling Q_{c_i} , using a constant covariance already yields superior performance, as visibility is the dominant factor in active mapping, which is consistent with

the findings in [104]. Since obtaining more accurate Q_{c_i} with FisherRF+GAVIS or VIMC+GAVIS requires additional processing with substantially higher computational cost, the base GAVIS model provides a more favorable trade-off between performance and efficiency in practice.

We include additional uncertainty quantification results using AUSE-D and AUSE-V in Tab. 5, covering all baseline methods and ablations. GAVIS achieves the best performance in AUSE-V among all baselines and ablations that disable individual components, which is consistent with its superior active mapping performance and qualitative results. GAVIS also attains the best AUSE-D score among all baseline methods. However, it does not outperform the isotropic ablation that disables anisotropic visibility by setting $\nu(\mathbf{d}; \mathbf{d}_p) = 1$. This discrepancy arises from the misalignment between AUSE-D and the objective of active mapping, as discussed in Sec. 11. A simple example can be constructed from a scenario similar to Fig. 7, where the 3DGS model produces plausible imaginations behind a wall, leading to relatively accurate depth predictions in those regions and consequently improving the overall AUSE-D score, even though the model incorrectly assigns low uncertainty to unobserved areas without considering anisotropic visibility.

	GAVIS	NVF	FisherRF	VIMC	Iso.	w/o DC	Iso. w/o DC
AUSE-D ↓	0.224	0.381	0.463	0.504	0.205	0.515	0.413
AUSE-V ↓	0.176	0.231	0.496	0.447	0.292	0.480	0.514

Table 5. Additional uncertainty quantification results

We conduct additional ablation studies on the SH degree L , anisotropy parameter κ , and the density of virtual particles ρ in visibility field density control, compared against the default GAVIS settings ($L = 2$, $\kappa = 1$, $\rho = 100$). The results in Tab. 7, averaged over all datasets, indicate that GAVIS is relatively robust to these hyperparameters. Specifically, for κ , the default $\kappa = 1$ performs best overall. Smaller values modestly reduce rendering quality, while overly large values (e.g., $\kappa = 10$, a sharper $\nu(\mathbf{d}; \mathbf{d}_p)$) degrade visual coverage. For SH degree L , $L = 3$ yields a slight improvement over $L = 2$, but gains plateau at higher degrees; we retain $L = 2$ as the default for efficiency. For virtual particle density ρ , higher values consistently improve performance with diminishing returns, and $\rho = 100$ provides a practical balance between performance and computational cost.

12.2. Additional Qualitative Results

In this subsection, we provide additional comprehensive qualitative results across a wide variety of scenes, demonstrating the generality of GAVIS compared to baseline methods for uncertainty quantification and active mapping.

Fig. 8–9 visualize uncertainty quantification results. For

each scene, all methods are trained on the same set of views, which only partially observe the underlying scene. To enable direct evaluation of uncertainty quantification, we generate ground-truth visibility (GT VIS) maps by marking mesh faces that are invisible from the training views with a bright color (and visible faces with a dark color), and rendering this visibility-annotated mesh from the same queried viewpoints. An accurate uncertainty quantification algorithm should assign high uncertainty to invisible regions, consistent with the GT visibility.

As shown in Figs. 8–9, GAVIS provides uncertainty estimation that aligns most closely with the GT VIS maps, thanks to its analytical anisotropic visibility modeling. In contrast, NVF relies on a black-box neural network approximation of isotropic visibility, leading to a noisy and less accurate uncertainty map. All visibility-aware methods significantly outperform baselines that ignore visibility, which fail to assign high uncertainty to invisible regions. Qualitative results for FisherRF+GAVIS and VIMC+GAVIS are also presented, demonstrating significant improvements over the original method by reliably assigning high uncertainty to invisible regions.

Finally, additional active-mapping reconstruction results for all Gibson and HM3D scenes are shown in Figs. 10 and 11, demonstrating that our method leverages accurate uncertainty quantification to guide more effective exploration, resulting in broader scene coverage and higher-quality reconstructions.

Dataset	Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Acc \downarrow	Comp \downarrow	CR \uparrow	VIS \uparrow
NeRF Synth.	FisherRF	22.34 \pm 0.31	0.870 \pm 0.004	0.119 \pm 0.003	0.014 \pm 2e-4	0.014 \pm 0.001	0.626 \pm 0.011	0.376 \pm 0.010
	VIMC	23.14 \pm 0.25	0.880 \pm 0.003	0.107 \pm 0.003	0.013 \pm 4e-4	0.015 \pm 0.001	0.651 \pm 0.010	0.407 \pm 0.011
	NVF	22.59 \pm 0.26	0.859 \pm 0.005	0.147 \pm 0.004	0.025 \pm 0.002	0.020 \pm 0.001	0.549 \pm 0.014	0.431 \pm 0.005
	GAVIS (ours)	24.26 \pm 0.25	<u>0.894 \pm 0.002</u>	<u>0.097 \pm 0.002</u>	<u>0.014 \pm 2e-4</u>	0.011 \pm 0.001	<u>0.711 \pm 0.009</u>	<u>0.437 \pm 0.006</u>
	FisherRF+GAVIS (ours)	24.55 \pm 0.17	0.898 \pm 0.002	0.092 \pm 0.002	<u>0.013 \pm 3e-4</u>	<u>0.012 \pm 4e-4</u>	<u>0.703 \pm 0.008</u>	0.440 \pm 0.005
	VIMC+GAVIS (ours)	<u>24.30 \pm 0.18</u>	<u>0.894 \pm 0.002</u>	<u>0.096 \pm 0.001</u>	0.014 \pm 2e-4	<u>0.012 \pm 3e-4</u>	0.713 \pm 0.006	<u>0.433 \pm 0.005</u>
Space	FisherRF	24.17 \pm 0.08	0.834 \pm 0.004	0.158 \pm 0.002	0.015 \pm 3e-4	0.020 \pm 0.001	0.547 \pm 0.017	0.474 \pm 0.016
	VIMC	24.56 \pm 0.55	0.841 \pm 0.004	0.150 \pm 0.002	<u>0.015 \pm 4e-4</u>	0.017 \pm 0.003	0.612 \pm 0.008	0.510 \pm 0.009
	NVF	23.76 \pm 0.46	0.796 \pm 0.012	0.202 \pm 0.010	0.037 \pm 0.003	0.015 \pm 0.002	0.499 \pm 0.019	0.564 \pm 0.017
	GAVIS (ours)	26.14 \pm 0.10	0.857 \pm 0.003	<u>0.140 \pm 0.002</u>	<u>0.014 \pm 4e-4</u>	0.013 \pm 0.002	<u>0.630 \pm 0.019</u>	<u>0.582 \pm 0.017</u>
	FisherRF+GAVIS (ours)	<u>25.44 \pm 0.31</u>	<u>0.857 \pm 0.003</u>	0.139 \pm 0.002	0.013 \pm 0.001	<u>0.014 \pm 0.001</u>	<u>0.631 \pm 0.014</u>	<u>0.588 \pm 0.023</u>
	VIMC+GAVIS (ours)	<u>25.47 \pm 0.34</u>	<u>0.857 \pm 0.002</u>	<u>0.141 \pm 0.001</u>	0.016 \pm 0.001	<u>0.013 \pm 2e-4</u>	0.640 \pm 0.008	0.597 \pm 0.009
Gibson	FisherRF	18.11 \pm 0.50	0.720 \pm 0.007	0.419 \pm 0.006	<u>0.038 \pm 0.001</u>	0.891 \pm 0.139	0.431 \pm 0.031	0.469 \pm 0.035
	VIMC	15.70 \pm 0.20	0.668 \pm 0.003	0.465 \pm 0.003	<u>0.046 \pm 0.002</u>	1.006 \pm 0.067	0.337 \pm 0.013	0.366 \pm 0.014
	NVF	23.29 \pm 0.12	0.798 \pm 0.001	0.402 \pm 0.002	0.029 \pm 3e-4	0.033 \pm 0.001	0.880 \pm 0.002	0.915 \pm 0.002
	GAVIS (ours)	<u>24.42 \pm 0.14</u>	<u>0.812 \pm 0.003</u>	<u>0.323 \pm 0.004</u>	<u>0.040 \pm 0.001</u>	<u>0.044 \pm 0.002</u>	<u>0.831 \pm 0.005</u>	<u>0.890 \pm 0.003</u>
	FisherRF+GAVIS (ours)	24.58 \pm 0.12	0.815 \pm 0.002	0.319 \pm 0.003	<u>0.041 \pm 0.001</u>	<u>0.044 \pm 0.002</u>	<u>0.836 \pm 0.006</u>	<u>0.890 \pm 0.007</u>
	VIMC+GAVIS (ours)	<u>24.04 \pm 0.19</u>	<u>0.807 \pm 0.002</u>	<u>0.324 \pm 0.003</u>	0.041 \pm 3e-4	0.075 \pm 0.006	0.772 \pm 0.009	0.834 \pm 0.010
HM3D	FisherRF	18.32 \pm 0.44	0.693 \pm 0.009	0.446 \pm 0.009	<u>0.040 \pm 0.001</u>	0.503 \pm 0.097	0.447 \pm 0.025	0.558 \pm 0.030
	VIMC	17.15 \pm 0.26	0.645 \pm 0.006	0.477 \pm 0.006	<u>0.052 \pm 0.001</u>	0.178 \pm 0.038	0.476 \pm 0.017	0.618 \pm 0.020
	NVF	22.69 \pm 0.14	0.760 \pm 0.002	0.434 \pm 0.002	0.036 \pm 0.001	0.040 \pm 0.001	<u>0.819 \pm 0.003</u>	<u>0.873 \pm 0.001</u>
	GAVIS (ours)	<u>23.97 \pm 0.07</u>	<u>0.791 \pm 0.001</u>	<u>0.338 \pm 0.002</u>	<u>0.043 \pm 4e-4</u>	<u>0.040 \pm 3e-4</u>	<u>0.820 \pm 0.003</u>	<u>0.876 \pm 0.001</u>
	FisherRF+GAVIS (ours)	24.23 \pm 0.07	0.797 \pm 0.001	0.330 \pm 0.002	<u>0.043 \pm 3e-4</u>	<u>0.040 \pm 3e-4</u>	0.821 \pm 0.002	0.877 \pm 0.001
	VIMC+GAVIS (ours)	<u>23.04 \pm 0.13</u>	<u>0.774 \pm 0.002</u>	<u>0.347 \pm 0.002</u>	0.043 \pm 5e-4	0.057 \pm 0.002	0.750 \pm 0.008	0.826 \pm 0.005

Table 6. **Quantitative results.** Active mapping performance across all datasets and methods. Best results are shown in **bold**; second and third best are underlined.

Setting	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Acc \downarrow	Comp \downarrow	CR \uparrow	VIS \uparrow
GAVIS (default)	24.70 \pm 0.08	0.839 \pm 0.001	0.224 \pm 0.001	0.028 \pm 2e-4	0.027 \pm 0.001	0.748 \pm 0.006	0.697 \pm 0.005
$\kappa = 0$	24.22 \pm 0.14	0.832 \pm 0.001	0.223 \pm 0.001	0.027 \pm 3e-4	0.027 \pm 0.001	0.744 \pm 0.005	0.666 \pm 0.005
$\kappa = 0.1$	24.26 \pm 0.18	0.834 \pm 0.001	0.222 \pm 0.001	0.027 \pm 4e-4	0.026 \pm 0.001	0.747 \pm 0.005	0.671 \pm 0.005
$\kappa = 0.3$	24.39 \pm 0.13	0.836 \pm 0.002	0.222 \pm 0.001	0.028 \pm 3e-4	0.026 \pm 4e-4	0.753 \pm 0.006	0.686 \pm 0.003
$\kappa = 3$	24.71 \pm 0.11	0.843 \pm 0.001	0.216 \pm 0.001	0.026 \pm 2e-4	0.043 \pm 0.004	0.725 \pm 0.007	0.676 \pm 0.006
$\kappa = 10$	24.04 \pm 0.16	0.835 \pm 0.002	0.223 \pm 0.002	0.024 \pm 4e-4	0.112 \pm 0.017	0.662 \pm 0.009	0.610 \pm 0.013
$L = 1$	24.61 \pm 0.10	0.837 \pm 0.002	0.224 \pm 0.002	0.027 \pm 3e-4	0.028 \pm 0.001	0.749 \pm 0.005	0.694 \pm 0.004
$L = 3$	24.90 \pm 0.09	0.844 \pm 0.002	0.217 \pm 0.001	0.027 \pm 2e-4	0.027 \pm 0.001	0.761 \pm 0.003	0.702 \pm 0.006
$L = 4$	24.73 \pm 0.10	0.841 \pm 0.001	0.220 \pm 0.001	0.027 \pm 2e-4	0.027 \pm 0.001	0.758 \pm 0.004	0.700 \pm 0.006
$L = 5$	24.74 \pm 0.15	0.840 \pm 0.003	0.220 \pm 0.002	0.028 \pm 3e-4	0.027 \pm 0.001	0.755 \pm 0.008	0.697 \pm 0.008
$\rho = 10$	24.53 \pm 0.08	0.837 \pm 0.001	0.225 \pm 0.001	0.028 \pm 3e-4	0.033 \pm 0.002	0.738 \pm 0.005	0.685 \pm 0.006
$\rho = 25$	24.54 \pm 0.12	0.836 \pm 0.001	0.224 \pm 0.001	0.028 \pm 2e-4	0.031 \pm 0.001	0.734 \pm 0.003	0.683 \pm 0.003
$\rho = 50$	24.54 \pm 0.11	0.839 \pm 0.001	0.222 \pm 0.001	0.028 \pm 3e-4	0.029 \pm 0.001	0.750 \pm 0.003	0.693 \pm 0.003
$\rho = 200$	24.99 \pm 0.08	0.844 \pm 0.001	0.216 \pm 0.001	0.027 \pm 2e-4	0.026 \pm 2e-4	0.765 \pm 0.003	0.703 \pm 0.004
$\rho = 400$	24.95 \pm 0.09	0.844 \pm 0.001	0.215 \pm 0.001	0.027 \pm 2e-4	0.025 \pm 3e-4	0.766 \pm 0.005	0.701 \pm 0.004
$\rho = 1000$	25.07 \pm 0.10	0.847 \pm 0.001	0.212 \pm 0.001	0.026 \pm 2e-4	0.030 \pm 0.004	0.763 \pm 0.004	0.700 \pm 0.005

Table 7. **Additional ablation studies.** Active mapping performance of GAVIS with different hyperparameter settings. GAVIS uses $\kappa = 1$, $L = 2$ and $\rho = 100$ by default.

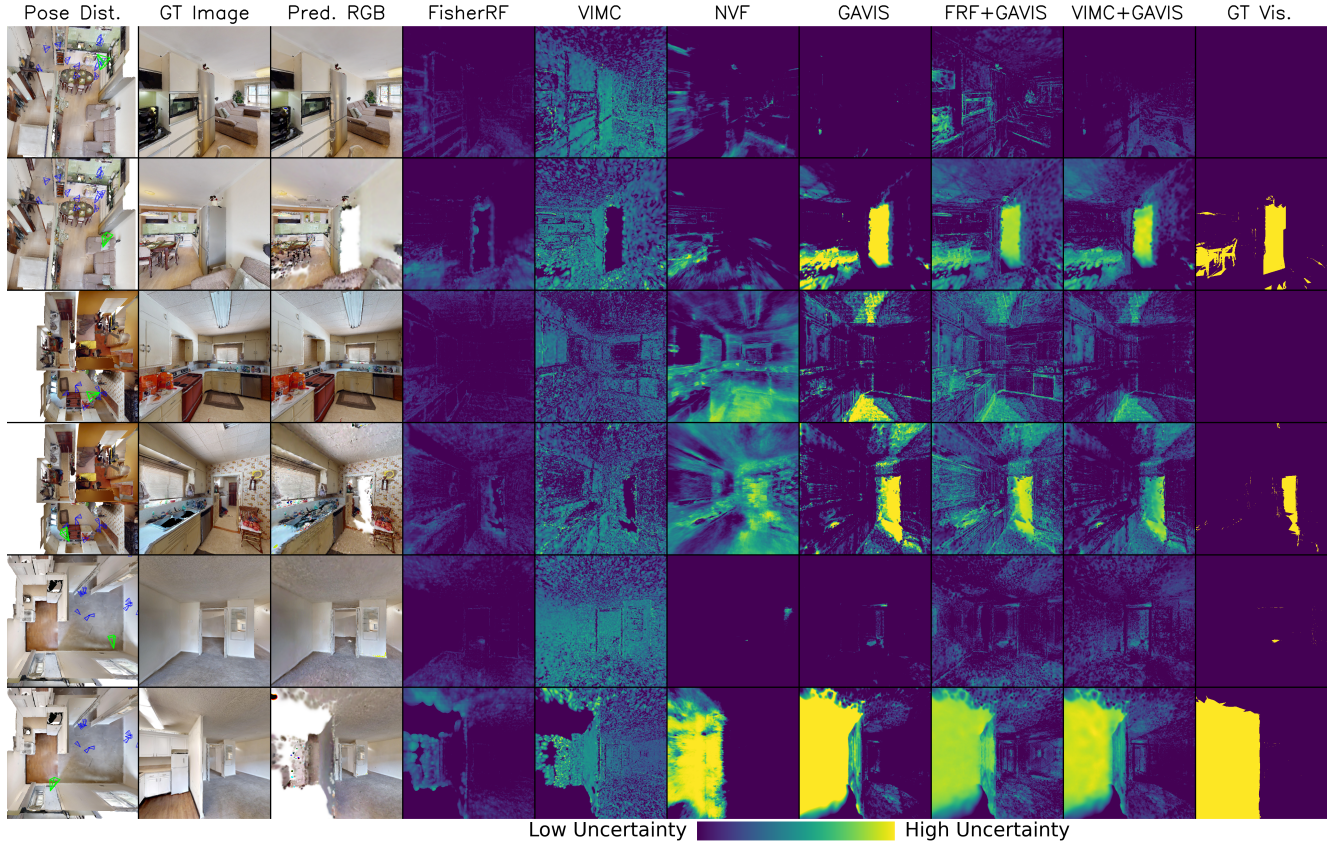


Figure 8. **Qualitative results for uncertainty quantification** in indoor scenes (Gibson and HM3D). From left to right: (1) Top-down view of the scene with training views (blue frustums) that cover only part of the room, and the queried view for uncertainty evaluation (green frustum); (2) Ground-truth RGB image rendered from the scene mesh; (3) Synthesized RGB image from a 3DGS model trained on the partial-view dataset; Uncertainty maps produced by (4) FisherRF, (5) VIMC, (6) NVF, (7) GAVIS, (8) FisherRF+GAVIS, (9) VIMC+GAVIS; (10) Ground-truth visibility map used as reference, where bright regions indicate invisible areas and dark regions indicate visible areas. An accurate uncertainty quantification method should closely align with the GT visibility by reliably assigning high uncertainty to invisible regions. All methods are trained using the same set of training views.

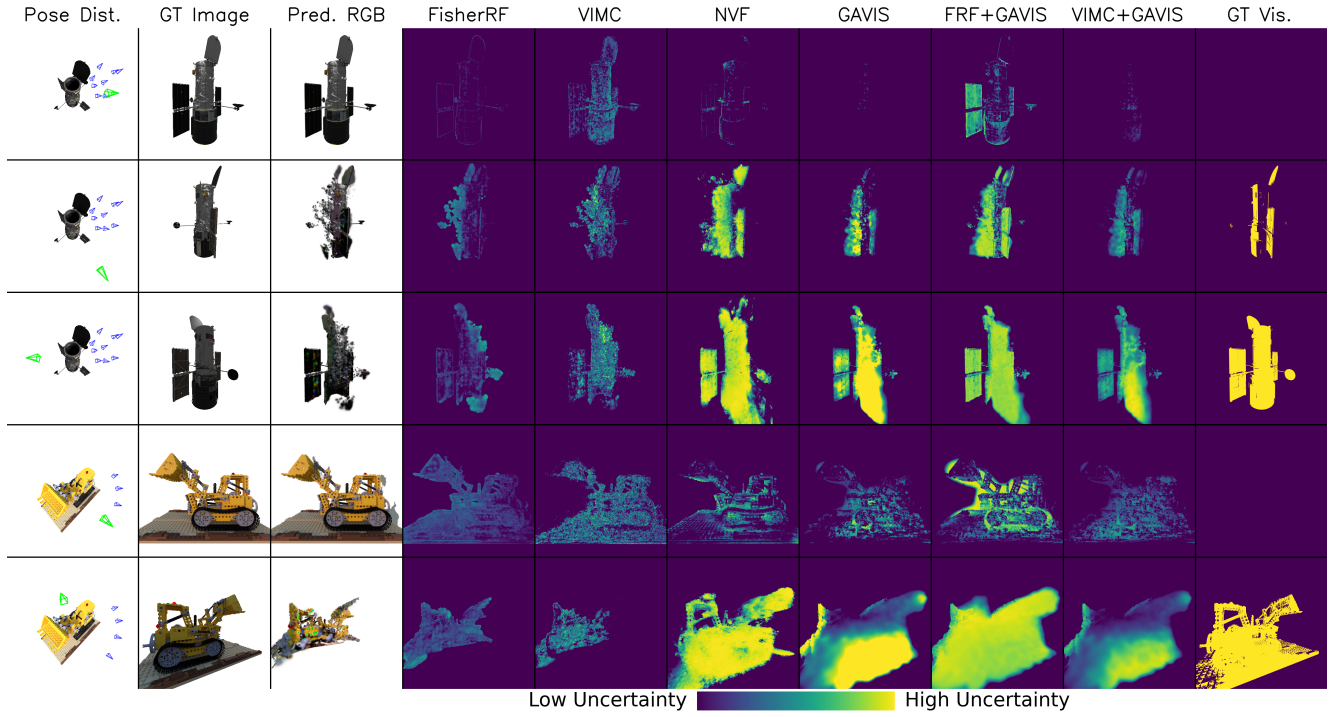


Figure 9. **Qualitative results for uncertainty quantification** in HST (from space dataset) and Lego (from NeRF Synthetic dataset) scenes. From left to right: (1) Isometric view of the object with training views (blue frustums) that cover only part of the object geometry, and the queried view for uncertainty evaluation (green frustum); (2) Ground-truth RGB image rendered from the scene mesh; (3) Synthesized RGB image from a 3DGS model trained on the partial-view dataset; Uncertainty maps produced by (4) FisherRF, (5) VIMC, (6) NVF, (7) GAVIS, (8) FisherRF+GAVIS, (9) VIMC+GAVIS; (10) Ground-truth visibility map used as reference, where bright regions indicate invisible areas and dark regions indicate visible areas. An accurate uncertainty quantification method should closely align with the GT visibility by reliably assigning high uncertainty to invisible regions. All methods are trained using the same set of training views.

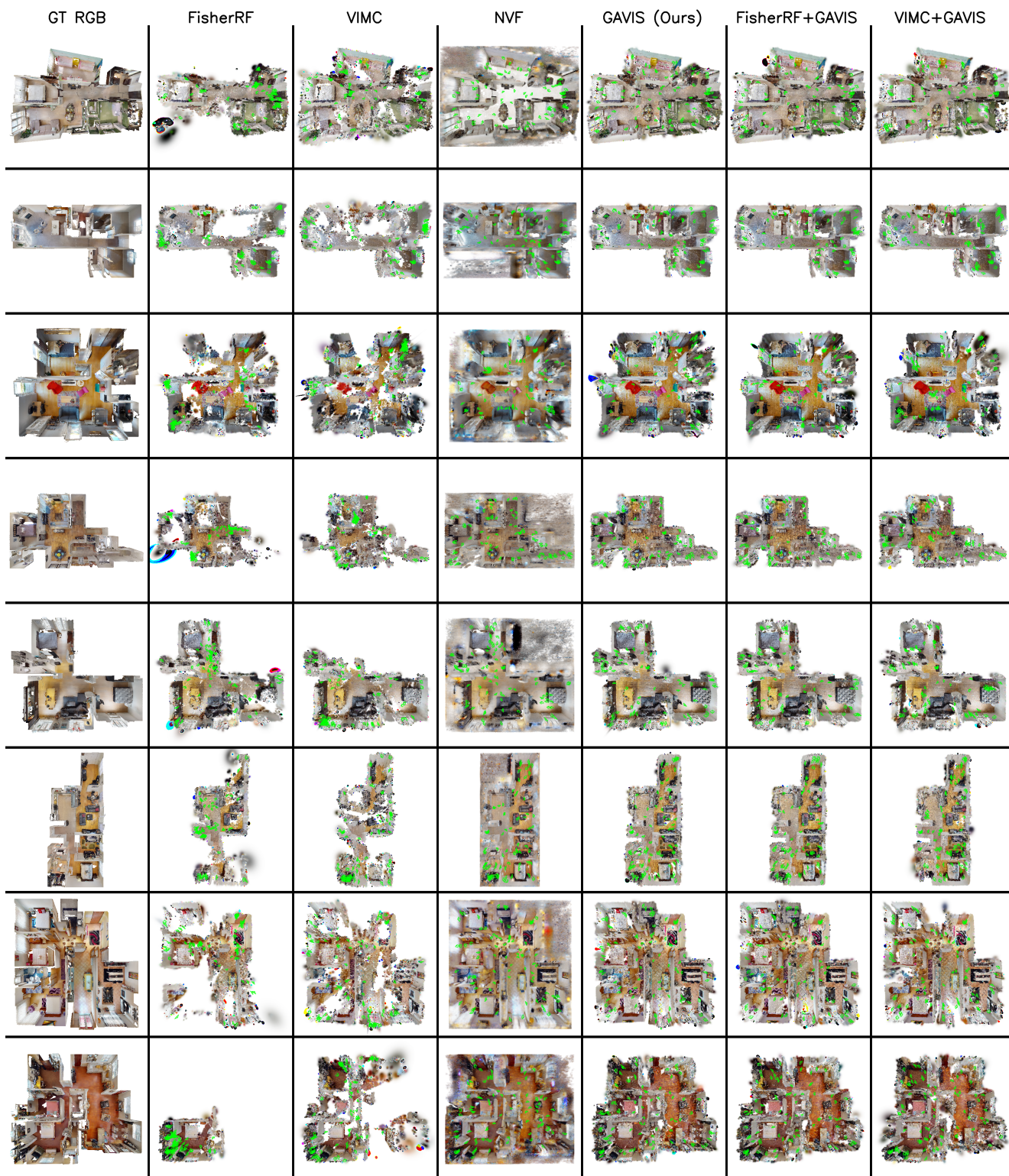


Figure 10. **Qualitative results for active mapping** in all HM3D scenes From left to right: (1) Ground-truth top-down view; Reconstruction results from (2) FisherRF, (3) VIMC, (4) NVF, (5) GAVIS, (6) FisherRF+GAVIS, (7) VIMC+GAVIS. Planned camera poses are shown as green frustums.

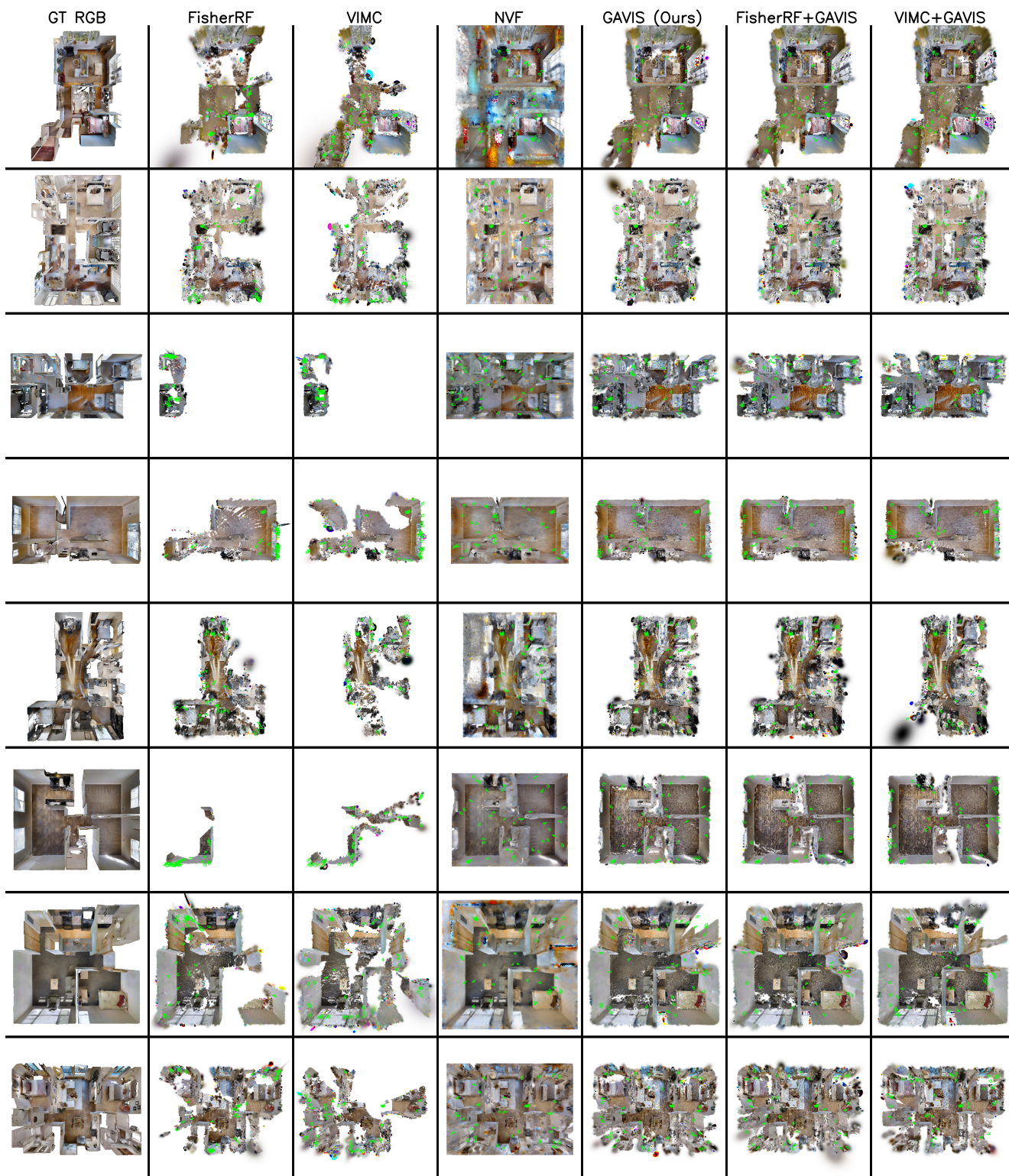


Figure 11. **Qualitative results for active mapping** in all Gibson scenes From left to right: (1) Ground-truth top-down view; Reconstruction results from (2) FisherRF, (3) VIMC, (4) NVF, (5) GAVIS, (6) FisherRF+GAVIS, (7) VIMC+GAVIS. Planned camera poses are shown as green frustums.