

Active Next-Best-View Optimization for Risk-Averse Path Planning

Amirhossein Mollaei Khass, Guangyi Liu, Vivek Pandey, Wen Jiang, Boshu Lei,
Kostas Daniilidis, Nader Motee

Abstract—Safe navigation in uncertain environments requires planning methods that integrate risk aversion with active perception. In this work, we present a unified framework that refines a coarse reference path by constructing tail-sensitive risk maps from Average Value-at-Risk statistics on an online-updated 3D Gaussian-splat Radiance Field. These maps enable the generation of locally safe and feasible trajectories. In parallel, we formulate Next-Best-View (NBV) selection as an optimization problem on the $SE(3)$ pose manifold, where Riemannian gradient descent maximizes an expected information gain objective to reduce uncertainty most critical for imminent motion. Our approach advances the state-of-the-art by coupling risk-averse path refinement with NBV planning, while introducing scalable gradient decompositions that support efficient online updates in complex environments. We demonstrate the effectiveness of the proposed framework through extensive computational studies.

I. INTRODUCTION

Autonomous navigation in unknown or partially observable environments remains a fundamental challenge in robotics, particularly for applications requiring real-time mapping, localization, and risk-averse trajectory planning. In such settings, a robot must simultaneously build a model of its environment, localize itself within that model, and plan safe, goal-directed paths that account for uncertainty and potential hazards [1]–[4]. At the same time, it must actively acquire new observations to refine its internal map and improve decision-making.

While traditional SLAM techniques have made significant strides [5], [6], many systems—such as AG-SLAM [4]—treat exploration, perception, and safe planning as decoupled components. Active approaches typically focus on maximizing visibility or reducing global map [7], [8], without explicitly considering downstream

task objectives like reaching a specific goal under safety constraints.

Recent advancements in 3D Gaussian Splatting (3DGS) [9] have introduced promising tools for efficient and high-fidelity scene reconstruction. These have been integrated into SLAM pipelines [10] and extended to active perception frameworks like AG-SLAM [4], which combines 3DGS with next-best-view strategies. However, such methods predominantly support unconstrained exploration and do not address the scenario where a robot must navigate purposefully toward a goal. In contrast, safety-aware methods such as RaEM [1] emphasize risk-aware viewpoint selection but often rely on predefined viewpoint sets and do not integrate real-time motion planning with onboard sensory feedback. Recent work such as Splat-Nav [2] leverages pretrained Gaussian Splatting scene representations to plan safe navigation corridors.

Despite progress, a key gap remains: in realistic missions, robots are frequently required to follow a task-specific trajectory from a start to a goal position—where significant deviations from this path are undesirable or infeasible. Existing view planning and exploration algorithms are not well suited to this constrained setting, where safety must be ensured along a predefined or planned route, not in free space. This motivates our formulation, which tightly integrates information gathering with trajectory-aware, risk-averse planning.

In this work, we present a unified framework for risk-averse, perception-driven trajectory planning that tightly couples safe motion with informative sensing in unknown 3-D environments. A coarse reference path is refined by constructing conservative risk maps using Average Value-at-Risk (AV@R) over an online-updated 3D Gaussian-splat Radiance Field, from which safe lattice points are extracted and searched with A^* . Concurrently, next-best-views are optimized on the $SE(3)$ pose manifold via Riemannian gradient descent on a Fisher-information objective to reduce uncertainty most critical for imminent motion. Our key contributions are: (i) a risk-averse replanning framework that integrates local A^* with real-time AV@R-based safety filtering using 3D Gaussian Splat Radiance Fields; (ii) a Riemannian

This work was supported partly by the ONR N00014-23-1-2779.

A.M. Khass, V. Pandey and N. Motee are with the Department of Mechanical Engineering and Mechanics, Lehigh University. {ammb23, vkp219, motee}@lehigh.edu.

G. Liu is with Amazon Robotics. gyliu@amazon.com. This paper is independent of his position at Amazon and does not relate to his employment there.

W. Jiang, B. Lei, and K. Daniilidis are with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104, USA. {wenjiang, leiboshu}@seas.upenn.edu, kostas@cis.upenn.edu.

optimization scheme on SE(3) for next-best-view computation, maximizing information gain under geometric and task-specific heuristics; and (iii) computational enhancements such as decomposable gradients that enable stochastic or mini-batch updates for scalability.

II. RELATED WORKS

Safe navigation in visually rich environments has gained renewed interest with the emergence of efficient scene representations like Gaussian Splatting (GSplat). These representations enable real-time rendering and dense mapping, making them well-suited for onboard planning and control. Splat-Nav [2] introduces a two-part framework for safe navigation using pretrained GSplat scenes: Splat-Plan, which computes safe corridors via a fast ellipsoid-ellipsoid collision test, and Splat-Loc, which performs robust localization using RGB images through a PnP algorithm. Although their method could, in principle, be applied to dynamic scenes, the authors evaluate Splat-Nav only in preconstructed static environments, without real-time updates of the 3DGS, noting that existing GSplat SLAM algorithms do not yet run in real time. On the other hand, AG-SLAM [4] extends GSplat to active SLAM, planning trajectories that balance information gain and localization uncertainty using Fisher Information. Unlike earlier exploration methods that assume perfect tracking, AG-SLAM accounts for pose uncertainty and improves robustness in challenging conditions. Still, it does not explicitly enforce safety during navigation. SAFER-Splat [3] proposes a real-time safety filter based on a novel Control Barrier Function (CBF) that operates directly on GSplat primitives. SAFER-Splat assumes that the underlying GSplat map is already reasonably accurate and does not leverage active perception to update the environment. In contrast, our approach employs risk-aware safety measures and actively selects next-best-views to maximize expected information gain, enabling continuous map refinement in the vicinity of the robot’s trajectory during exploration. Due to space limitations, we restrict our discussion to recent works only.

III. PROBLEM FORMULATION

We consider an autonomous robot equipped with a forward-facing RGB-D camera and onboard computation that must traverse an unknown or partially known 3-D environment under sensing, safety, and resource constraints. The robot is provided a coarse reference trajectory and maintains an online 3D Gaussian-splat (3DGS) map updated from acquired views. The objectives are to

(i) ensure collision-free operation, (ii) produce a near-optimal short trajectory, and (iii) selectively learn only the scene regions required for safe navigation to limit computation and communication. To this end, the planner alternates short-horizon, risk-aware path optimization with NBV computation: each waypoint is treated as a local subgoal, a forward-looking risk mask is formed from the current 3DGS risk field, and a proximity-weighted Expected-Information Gain (EIG) objective is optimized to produce NBV poses that maximally reduce uncertainty relevant to imminent motion. Executing the resulting control and sensing actions for the local path segment and assimilating the new images closes the loop, yielding a real-time, information-driven strategy for safe navigation in complex 3-D scenes.

IV. TECHNICAL PRELIMINARIES

In this section, we establish the technical foundations of our approach.

A. Learning 3D Scenes via Gaussian Splatting

In this work, We represent environment as 3D Gaussian Splatting (3DGS) map [9]. Volumetric rendering [11] forms the foundation of modern 3D scene reconstruction approaches such as NeRF [12], Plenoxels [13], and 3D Gaussian Splatting [9]. The discrete volumetric rendering equation in 3D Gaussian Splatting (3DGS) [14] produces the pixel color as $\hat{C} = \sum_{i=1}^{N_s} T_i \rho_i \mathbf{c}_i$, and $T_i = \prod_{j=1}^{i-1} (1 - \rho_j)$, where T_i is the accumulated transmittance at pixel location $x_i \in \mathbb{R}^2$, ρ_i is the blending coefficient of the i -th Gaussian, and \mathbf{c}_i is its color. The blending coefficient is defined as $\rho_i = \epsilon_i \exp(-\frac{1}{2}(x_i - \mu_i)^\top \Sigma_i^{-1}(x_i - \mu_i))$, where ϵ_i is the opacity, and μ_i, Σ_i define the 2D projected position and shape of the Gaussian in screen space. Depth rendering is computed analogously. The discrete depth rendering equation to produce \hat{D} is given by $\hat{D} = \sum_{i=1}^{N_s} T_i \rho_i d_i$, where d_i is the z -depth of the i -th Gaussian in camera space. The model is optimized jointly using RGB and depth supervision:

$$\mathcal{L} = \sum_{i=1}^{N_s} (\mathcal{L}_1(C, \hat{C}) + \psi \mathcal{L}_1(D, \hat{D})), \quad (1)$$

where C and D are the ground-truth RGB and depth, \hat{C} and \hat{D} are the rendered predictions, \mathcal{L}_1 is the L_1 loss, and $\psi \in [0, 1)$ balances depth relative to RGB.

B. Quantifying Collision Risk in 3D Gaussian Splatting

Since our ultimate objective is to plan a risk-averse path through an uncertain environment while simultaneously updating the scene representation, we begin by introducing a principled measure of safety. In this work,

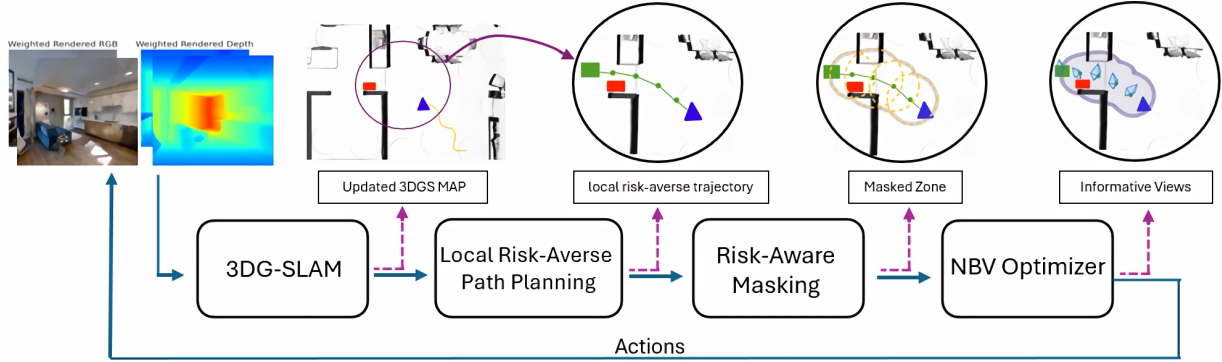


Fig. 1: Block diagram of the proposed risk-averse trajectory planning as in Algorithm 1. Each cycle the system updates the 3D Gaussian point cloud via SLAM, computes a risk field, synthesizes a safe local path, masks the environment around that path, selects high-risk splats, and evaluates candidate NBV poses over this focused subset. Executing the control and sensing actions steers the robot toward its goal while simultaneously gathering informative observations.

we employ the Average Value-at-Risk, formally defined as follows.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ denote the underlying probability space. For a continuous random variable $y : \Omega \rightarrow \mathbb{R}$, the Average Value-at-Risk (AV@R) at level $\varepsilon \in (0, 1)$ is defined as conditional expected value [15], [16]:

$$\text{AV@R}_\varepsilon(y) := \mathbb{E}[y \mid y < \text{V@R}_\varepsilon(y)], \quad (2)$$

where the corresponding Value-at-Risk (V@R) is given by $\text{V@R}_\varepsilon(y) := \inf\{z \in \mathbb{R} \mid \mathbb{P}(y < z) > \varepsilon\}$.

We aim to construct a risk field over the environment, quantifying collision risk at every point in space. We begin by discretizing the workspace into a rectilinear grid with vertices $q = q_{ijk} = (\alpha_i, \beta_j, \gamma_k) \in \mathbb{R}^3$, and denote the full set of vertices by \mathcal{Q} . The 3D environment itself is modeled as a radiance field represented by Gaussian splats, where each splat is $x_i \sim \mathcal{N}(\mu_i, \Sigma_i)$ with the full collection $\mathcal{X}_t = \{x_i\}$. To evaluate the local risk at a vertex q , we first measure its relationship to each Gaussian x_i . This is captured through the signed distance:

$$d(q, x_i) := \left\langle x_i - q, \frac{\mu_i - q}{\|\mu_i - q\|_2} \right\rangle, \quad (3)$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product [1], and $\|\cdot\|_2$ denotes the 2-norm. Assuming isotropic covariance $\Sigma_i = \sigma_i^2 I_3$, this signed distance itself becomes a Gaussian random variable $d(q, x_i) \sim \mathcal{N}(\|\mu_i - q\|_2, \sigma_i^2)$. We then quantify safety in terms of the Average Value-at-Risk (AV@R) of this distance distribution, which measures the expected value of the worst ε -fraction of outcomes:

$$\text{AV@R}_\varepsilon(d) = \|\mu_i - q\|_2 - \frac{\sigma_i}{\sqrt{2\pi}} \cdot \frac{1}{\varepsilon \exp(\iota^2)}, \quad (4)$$

where $\iota = \text{erf}^{-1}(2\varepsilon - 1)$, and $\text{erf}^{-1}(\cdot)$ is the inverse error function. Finally, to obtain a conservative measure

of risk at each vertex q , we take the minimum AV@R value across all splats:

$$\alpha(q) := \min_{x_i \in \mathcal{X}_t} \text{AV@R}_\varepsilon(d(q, x_i)). \quad (5)$$

Repeating this computation across all vertices $q \in \mathcal{Q}$ yields risk a scalar field, which we interpret as the risk field of the environment. This risk field serves as the foundation for risk-averse planning and the NBV optimization.

C. Risk-Aware Environment Masking for NBV

Traditional NBV methods evaluate a finite pool of candidate camera poses and select the pose that maximizes information gain over the entire scene [17]. In our setting, the camera is rigidly mounted to the robot, so the robot pose and the camera orientation together determine the observed view. Rather than optimizing a global information objective, we weight information by its relevance to future risk along the planned trajectory by prioritizing observations that most effectively reduce collision uncertainty in regions that will be traversed by the robot. This risk-weighted NBV criterion concentrates sensing resources on the regions that directly affect the robot's near-term motions, deliberately de-emphasizing distant parts of the scene that do not influence imminent decisions. By prioritizing measurements inside the forward-looking mask, the planner ignores far-range scenes unless and until it becomes relevant, which yields safer, lower-latency, and more computationally efficient exploration.

We construct a risk-driven masked environment and restrict the NBV search to this focused region. Given a coarse reference trajectory between two consecutive subgoals, represented by a sequence of 3D waypoints $\mathcal{Z} = \{z_1, z_2, \dots, z_N\}$ with $z_i \in \mathbb{R}^3$, each waypoint z_k

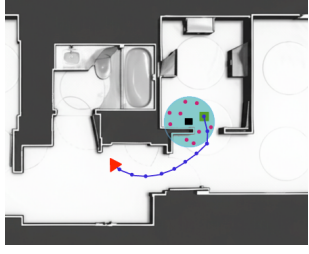


Fig. 2: This shows a safe proxy subgoal selection as explained in Subsection V-A. A safe subgoal \bar{z}_{j+1} (■) is chosen within a ball centered at the current unsafe subgoal z_{j+1} (■). Robot is depicted by ▲ and the connected blue dots show the planned local trajectory \mathcal{Z}_j^s .

is assigned a masking radius

$$r_{\text{mask}}(z_k) = \beta_1 e^{-\beta_2 \alpha(z_k)}, \quad \beta_1, \beta_2 > 0,$$

where $\alpha(z_k)$ denotes the risk field evaluated at z_k . The forward-looking masked region around \mathcal{Z} is defined as the union of Euclidean balls

$$\Pi_{\mathcal{Z}} = \bigcup_{z_k \in \mathcal{Z}} \mathcal{B}(z_k, r_{\text{mask}}(z_k)). \quad (6)$$

Thus, $\Pi_{\mathcal{Z}}$ collects the scene elements that most strongly influence the robot's near-term collision risk; see Fig. 2.

The next-best view is then selected by maximizing a proximity-weighted Fisher-information Expected Information Gain (EIG) over this masked region. Restricting the NBV search to $\Pi_{\mathcal{Z}}$ concentrates sensing and model updates on areas that directly affect imminent motion, thereby sparsifying acquisitions and reducing computation compared with global strategies.

V. OPTIMIZING NBV FOR RISK-AVERSE PLANNING

Our proposed framework is depicted in Fig. 1 and summarized in Algorithm 1. Building upon the concepts introduced in Subsection IV-C, we establish the foundations of our method and provide formal safety guarantees. Consider a robot deployed in an unknown and uncertain environment, tasked with following a coarse reference trajectory $\mathcal{Z} = \{z_1, z_2, \dots, z_N\}$. Since no safety assurances are inherent to this reference path, some waypoints may fall within unsafe regions. To address this challenge in real-time safety-critical navigation, we propose an *adaptive, localized, risk-averse* trajectory planning scheme. As illustrated in Fig. 2, the scheme adjusts the next immediate waypoint (subgoal) to guarantee safety, while simultaneously identifying the most informative regions of the environment along the reference trajectory. These regions are used to update the 3DGS-based scene model, ensuring that the resulting trajectory remains provably safe.

A. Safety Verification for Local Trajectory Planning

In dynamically changing or partially known environments, the reference trajectory may not guarantee global safety. To ensure safe execution, we adopt a local replanning strategy guided by the localized risk profile. When the robot is located at waypoint $z_j \in \mathcal{Z}$, the subsequent waypoint z_{j+1} is treated as a *subgoal*. The task is then to compute a risk-averse path from z_j to z_{j+1} using only local environmental information.

To this end, we define a localized partition of the workspace, $\mathcal{Q}_j \subseteq \mathcal{Q}$, which contains all grid points between z_j and z_{j+1} . For each grid point $q \in \mathcal{Q}_j$, the worst-case risk metric (5) is evaluated. The subset of safe candidate points is then obtained as

$$\mathcal{Q}_j^f = \{q \in \mathcal{Q}_j \mid \alpha_q \geq \gamma\}, \quad (7)$$

where $\gamma \in \mathbb{R}_+$ denotes a user-specified risk tolerance. A collision-free, risk-averse path from z_j to z_{j+1} is computed over the filtered grid \mathcal{Q}_j^f using the standard A^* algorithm.

If the nominal waypoint z_{j+1} lies outside the safe set \mathcal{Q}_j^f , it is refined by selecting the safest feasible alternative within a neighborhood of z_{j+1} . The proxy subgoal is formally defined as

$$\bar{z}_{j+1} := \arg \max_{z \in \mathcal{Q}_j^f \cap \mathcal{B}(z_{j+1}, \delta(z_{j+1}))} \text{AV@R}(z), \quad (8)$$

where $\mathcal{B}(z_{j+1}, \delta(z_{j+1}))$ denotes the ball of radius $\delta(z_{j+1})$ centered at z_{j+1} . In rare cases where no feasible proxy subgoal exists, the robot continues to explore within the safe region \mathcal{Q}_j^f until a suitable goal can be identified. An illustration of this procedure is provided in Fig. 2.

The locally replanned, risk-averse trajectory segment from z_j to either the refined proxy subgoal \bar{z}_{j+1} or the nominal waypoint z_{j+1} (when feasible) is denoted as

$$\mathcal{Z}_j^s = \{r_j^1, r_j^2, \dots, r_j^{K_j}\}, \quad (9)$$

with $r_j^1 = z_j$, $r_j^{K_j} = z_{j+1}$ or \bar{z}_{j+1} , and K_j the number of points in the locally planned segment.

B. NBV Optimization via Proximity-Weighted EIG

Classical NBV planning strategies [17], [18] are effective for maximizing Expected Information Gain (EIG) in unconstrained exploration. However, they are less suitable when the robot must follow a prescribed trajectory from start to goal while adhering to safety constraints. In such task-driven scenarios, the robot cannot deviate significantly from its intended path.

To address this limitation, we propose a trajectory-aware NBV strategy that builds on the principles in

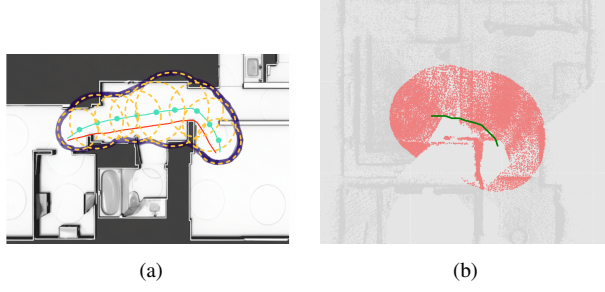


Fig. 3: Illustration of risk-aware environment masking to guide NBV selection, focusing on locally relevant Gaussians to maximize expected information gain and improve map accuracy. (a) Schematic of the masking process, showing Gaussian points near the trajectory in the masked region. (b) Subset of 3D Gaussian points selected within each safe zone to maximize information gain.

Subsection IV-C and operates along the locally safe trajectory segment \mathcal{Z}_j^s . At each waypoint r_j^k , the robot selects the most informative view, thereby balancing information acquisition with path feasibility. Formally, the NBV problem at each $r_j^k \in \mathcal{Z}_j^s$ can be written as

$$T_{k,j}^* = \arg \max_{T_{k,j} \in \text{SE}(3)} \mathcal{I}(T_{k,j}; \Pi_{\mathcal{Z}_j^s}), \quad (10)$$

where $\mathcal{I}(T_{k,j}; \Pi_{\mathcal{Z}_j^s})$ is the EIG as a function of the robot pose $T_{k,j}$ at waypoint r_j^k , parameterized by the 3D Gaussian points in $\Pi_{\mathcal{Z}_j^s}$. Following [17], the information gain for the masked region $\Pi_{\mathcal{Z}_j^s}$ is expressed as

$$\mathcal{I}(T; \Pi_{\mathcal{Z}_j^s}) = \text{tr}(\mathbf{H}''[\Phi | T, \mathbf{w}^*] \mathbf{H}''[\mathbf{w}^*]_{\text{prior}}^{-1}), \quad (11)$$

where $\text{tr}(Y)$ denotes the trace of matrix Y , and $\mathbf{H}''[\mathbf{w}^*]_{\text{prior}}^{-1}$ is obtained by aggregating Hessians of the 3DGS model parameters across all training views. The Hessian contribution of a candidate view is approximated as

$$\mathbf{H}''[\Phi | T, \mathbf{w}^*] \approx \text{diag}(J^\top J) + \lambda I, \quad (12)$$

where $J = \nabla_{\mathbf{w}} f(T, \mathbf{w}^*)$ is the Jacobian of the rendering model $f(T, \mathbf{w})$ with respect to Gaussian splat parameters \mathbf{w} , $\text{diag}(J^\top J)$ extracts the diagonal of $J^\top J$, and λ is a regularization constant.

To capture spatial relevance, we introduce a *proximity weighting function* that emphasizes nearby Gaussian splats relative to distant ones in $\Pi_{\mathcal{Z}_j^s}$. This reflects the intuition that nearby Gaussian splats carry greater importance for safe navigation, since the robot encounters them before more distant structures. We define an exponential decay weight function with respect to the graph distance between the robot and splat center:

$$v_T(\mu_i) = \alpha e^{-\beta \|p_T - \mu_i\|_g}, \quad (13)$$

where $\alpha, \beta > 0$ are design parameters, $\|\cdot\|_g$ denotes graph (grid-based) distance, p_T is the camera position

Algorithm 1: Risk-Averse Trajectory Planning

Initialize: Coarse reference trajectory from start to goal $\mathcal{Z} = \{z_1, \dots, z_N\}$

for $j = 1 : N$ **do**

Find safe filtered grid $\mathcal{Q}_j^f = \{q \in \mathcal{Q}_j \mid \alpha(q) \geq \gamma\}$

if $z_{j+1} \notin \mathcal{Q}_j^f$ **then**

Find safe proxy subgoal \bar{z}_{j+1} of z_{j+1}

$$\bar{z}_{j+1} := \arg \max_{z \in \mathcal{Q}_j^f \cap \mathcal{B}(z_{j+1}, \delta(z_{j+1}))} \text{AV@R}(z),$$

end if

Find shortest path \mathcal{Z}_j^s from z_j to \bar{z}_{j+1} using A^*

Find risk-aware masked environment $\Pi_{\mathcal{Z}_j^s}$

for $k = 1 : K_j$ **do**

Find NBV for $r_j^k \in \mathcal{Z}_j^s$

$$\psi^* = \arg \max_{\psi \in \mathbb{S}^1} \bar{\mathcal{I}}(T; \Pi_{\mathcal{Z}_j^s}),$$

Update 3D Gaussian Splatting

end for

end for

Return: Safe Path and Updated 3D Gaussian Map

from pose T , and μ_i is the mean of the i th Gaussian splat.

The weighting function is incorporated into the Fisher Information Matrix by modifying the Hessian approximation in (12):

$$\bar{\mathbf{H}}''[\Phi | T, \mathbf{w}^*] \approx [\bar{V} \text{diag}(J^\top J) + \lambda I], \quad (14)$$

where $\bar{V} = \text{diag}(V_1, \dots, V_{|\mathcal{X}_T|})$, with $V_i = v_T(\mu_i) \cdot I_{|w_i|}$, scales the contribution of each Gaussian according to its proximity to the viewpoint, and w_i is the parameter vector for splat $x_i \in \mathcal{X}$. The resulting proximity-weighted information gain is defined as

$$\bar{\mathcal{I}}(T; \Pi_{\mathcal{Z}_j^s}) = \text{tr}(\bar{\mathbf{H}}''[\Phi | T, \mathbf{w}^*] \mathbf{H}''[\mathbf{w}^*]_{\text{prior}}^{-1}), \quad (15)$$

which can be expressed equivalently as

$$\bar{\mathcal{I}}(T; \Pi_{\mathcal{Z}_j^s}) = \sum_{i=1}^{|\mathcal{X}|} v_T(\mu_i) \mathcal{I}_i(T; \Pi_{\mathcal{Z}_j^s}), \quad (16)$$

where $\mathcal{I}_i(T; \Pi_{\mathcal{Z}_j^s}) = \text{tr}(\bar{\mathbf{H}}''[\Phi | T, w_i^*] \mathbf{H}''[w_i^*]_{\text{prior}}^{-1})$ denotes the information gain associated with parameter w_i and $|\mathcal{X}|$ is the cardinality of set \mathcal{X} . The full pose optimization in (10) considers all six degrees of freedom in $\text{SE}(3)$. In the trajectory-aware NBV formulation, however, the robot is constrained to a fixed sequence of positions \mathcal{Z}_j^s . The translational component of T is thus fixed, and the orientation is restricted to yaw rotations due to platform or sensor field-of-view constraints. This

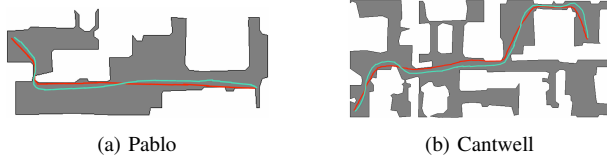


Fig. 4: Shortest path (—) and risk-averse path (—) shown in two representative environments; the risk-averse trajectory avoids high-risk regions at the expense of additional length.

reduces the optimization to a single parameter, the yaw angle $\psi \in \mathbb{S}^1$:

$$\arg \max_{\psi \in \mathbb{S}^1} \bar{\mathcal{I}}(T; \Pi_{\mathcal{Z}_j^s}), \quad (17)$$

where ψ represents the viewing angle at each waypoint.

We solve (17) using gradient ascent on the manifold \mathbb{S}^1 . The objective $\bar{\mathcal{I}}(\cdot)$ is differentiable with respect to ψ under mild assumptions, enabling first-order methods.

The process is repeated until convergence or until a maximum number of iterations is reached. In practice, only a few gradient steps are sufficient to yield an informative viewpoint policy.

C. Efficient Implementation of NBV Optimization

While the information-theoretic optimization is effective, computing the full gradient at each iteration can be costly in large-scale environments. To improve runtime, we exploit the additive structure of the information gain objective in (16), reformulating (17) as

$$\arg \max_{\psi \in \mathbb{S}^1} \sum_{i=1}^{|\mathcal{X}|} v_T(\mu_i) \mathcal{I}_i(T; \Pi_{\mathcal{Z}_j^s}). \quad (18)$$

By linearity of the gradient operator, the update becomes a weighted sum of per-splat gradients:

$$\psi^{(l+1)} \leftarrow \psi^{(l)} + \eta \sum_{i=1}^{|\mathcal{X}|} v_T(\mu_i) \nabla_{\psi} \mathcal{I}_i(T; \Pi_{\mathcal{Z}_j^s}). \quad (19)$$

where $\eta > 0$ is the step size, and the gradient $\nabla_{\psi} \bar{\mathcal{I}}(\cdot)$ is obtained via automatic differentiation through the trajectory parameterization. This decomposition enables parallelization and supports efficient variants such as stochastic and mini-batch gradient ascent, which trade exactness for scalability and faster convergence, which is important for real-time and resource-constrained implementation.

Theorem 1: Consider two prior information matrices $\mathbf{H}''[\mathbf{w}^*]_{\text{prior}}$ and $\mathbf{H}''[\mathbf{v}^*]_{\text{prior}}$, corresponding to parameters \mathbf{w}^* and \mathbf{v}^* of the rendering model. If the prior satisfies the Loewner ordering on the cone of positive semi-definite matrices:

$$\mathbf{H}''[\mathbf{w}^*]_{\text{prior}} \succeq \mathbf{H}''[\mathbf{v}^*]_{\text{prior}},$$



Fig. 5: Results from the Swormville scene (Gibson dataset). (a) Top-down map showing the ground-truth shortest path (—) and the executed risk-averse trajectory (—). (b) 3-D Gaussian-splat reconstruction of the scene; only the forward-looking masked region relevant to the task was used for learning, illustrating the sparse, task-focused sensing.

then the proximity-weighted EIG satisfies

$$\bar{\mathcal{I}}_{\mathbf{w}^*}(T; \Pi_{\mathcal{Z}_j^s}) \leq \bar{\mathcal{I}}_{\mathbf{v}^*}(T; \Pi_{\mathcal{Z}_j^s}),$$

where

$$\bar{\mathcal{I}}_{\square}(T; \Pi_{\mathcal{Z}_j^s}) = \text{tr}(\bar{\mathbf{H}}''[\Phi | T, \square^*] \mathbf{H}''[\square^*]_{\text{prior}}^{-1}),$$

for $\square \in \{\mathbf{w}, \mathbf{v}\}$.

Theorem 1 (Proof in Appendix) shows that EIG exhibits diminishing returns: views added to a well-trained model yield less information than those added earlier. We exploit this as a principled early-stopping rule for (19). Once the expected information gain falls below a threshold, further updates contribute negligibly to uncertainty reduction. Unlike fixed iteration limits or gradient-norm criteria, this stopping rule adapts to the informativeness of new views, ensuring efficient use of computation without sacrificing performance.

VI. EXPERIMENTS

In this section, we provide a brief overview of our experimental setup and discuss the results obtained from our algorithm 1.

A. Experimental Setup

The environment is simulated using the Habitat simulator [19] with scenes from the Gibson dataset [20]. All experiments are conducted on a machine with an Intel Core i9-13900K CPU and an NVIDIA RTX A2000 GPU, implemented in PyTorch 2.2 with CUDA 12. Mask parameters are fixed to $(\beta_1, \beta_2) = (0.2, 1.1)$, with a tolerance of $\gamma = 0.10$.

B. Evaluation Metrics

We evaluate our method across two key dimensions: trajectory safety and environment reconstruction quality.

Environment	PSNR \uparrow			SSIM \uparrow			LPIPS \downarrow			Depth MAE \downarrow		
	R=1	R=2	R=3	R=1	R=2	R=3	R=1	R=2	R=3	R=1	R=2	R=3
Denmark	16.71	16.53	16.23	0.65	0.64	0.63	0.47	0.49	0.50	0.31	0.32	0.34
Pablo	15.79	15.40	14.20	0.64	0.62	0.60	0.47	0.48	0.51	0.43	0.48	0.53
Cantwell	15.41	14.87	14.04	0.57	0.54	0.52	0.55	0.56	0.58	0.22	0.25	0.30
Eudora	15.77	14.84	13.66	0.71	0.66	0.60	0.46	0.50	0.52	0.15	0.32	0.40
Swormville	15.46	15.04	14.40	0.66	0.63	0.62	0.49	0.51	0.53	0.50	0.49	0.51
Elmira	14.51	13.32	12.97	0.61	0.59	0.57	0.50	0.51	0.58	0.35	0.38	0.42

TABLE I: Comparison of environment reconstruction metrics across 3 corridor radii ($R = 1, 2, 3$) around robot’s trajectory. Higher PSNR and SSIM and lower LPIPS and MAE indicate better reconstruction quality. As expected, quality degrades with increasing distance from the robot’s trajectory.

To quantify safety, we use the Average Value at Risk AV@R metric, which quantifies safety along a trajectory. A lower AV@R indicates a higher concentration of risk, whereas a higher AV@R implies that the robot is consistently avoiding high-risk regions. Thus, higher AV@R values reflect safer trajectories in our context.

For assessing environment reconstruction, we employ standard image-based metrics: PSNR (Peak Signal-to-Noise Ratio) [21], SSIM (Structural Similarity Index) [22], LPIPS (Learned Perceptual Image Patch Similarity) [23], and MAE (Mean Absolute Error). These metrics evaluate how closely the rendered outputs from our Gaussian Splatting scene representation match ground-truth views, measuring both pixel-wise accuracy and perceptual quality.

While we also report path length for completeness, we note that safety-oriented planning naturally leads to longer trajectories compared to shortest-path baselines (e.g., A^*). Therefore, path length is not used as a primary metric, but rather as a reference for analyzing the trade-off between safety and efficiency.

C. Effectiveness of our Algorithm

1) *Path Planning*: We evaluate our risk-aware planning algorithm across diverse environments, as illustrated in Fig. 4. The green trajectories represent paths generated by our method, while the red trajectories correspond to the shortest paths computed using A^* [24].

As visually evident from the figures, our planner consistently steers the robot away from high-risk areas, maintaining greater clearance from obstacles compared to the shortest-path baseline. This qualitative observation highlights the effectiveness of our approach in producing safer, more robust trajectories.

Quantitatively, Table III reports the path lengths and corresponding AV@R values. While our method results in longer trajectories, as expected in risk-averse navigation, it achieves significantly higher AV@R scores, affirming its ability to minimize exposure to high-risk regions and prioritize safety over raw efficiency.

2) *Environment Reconstruction Quality*: We evaluate the fidelity of environment reconstruction using standard image quality metrics—PSNR, SSIM, LPIPS, and depth MAE—reported in Table I. Our method employs a risk-aware masking strategy that emphasizes regions near the robot’s trajectory when quantifying information gain. This prioritization ensures that areas critical for safe navigation are reconstructed with higher fidelity, aligning with our objective of localized, risk-sensitive mapping.

To further assess reconstruction performance, we evaluate the metrics within varying radii around the robot’s path, effectively forming a spherical corridor centered on the trajectory. As the radius increases, PSNR and SSIM values consistently decrease, while LPIPS and depth MAE values increase indicating that perceptual and geometric accuracy diminish in regions farther from the robot. This trend is not only expected, but also desirable, as it confirms that our algorithm focuses reconstruction quality where it matters most: in the immediate vicinity of the robot’s trajectory. The consistent behavior across all metrics supports the effectiveness of our risk-aware strategy, and all observations are quantitatively supported by the results presented in Table I.

We evaluate our NBV optimization (Eq. (19)) by measuring the percentage increase in information gain achieved by the optimized view compared to a nominal one. As shown in Table II, the optimized NBV consistently yields higher information gain. While both the safety measure (Table III) and expected information gain (EIG) (Table II) show an increasing trend, they are not strongly correlated, as EIG is influenced by the inherent complexity of each environment.

VII. CONCLUSION

In this work, we tackled the problem of risk-averse navigation in unknown 3D environments. Our framework refines a coarse global path into a safe, perception-driven trajectory by maintaining an online 3D Gaussian Splatting (3DGS) map and actively reducing uncertainty

Environment	Denmark	Pablo	Cantwell	Eudora	Swormville	Elmira
% EIG \uparrow	41.6	28.0	35.2	20.21	33.60	20.83

TABLE II: The whole table: % increase in Expected Information Gain (EIG) from optimized NBV compared to a nominal view.

Environment	Path Length (m)			Safety Measure (m) \uparrow		
	Shortest	Ours	% loss	Shortest	Ours	% gain
Denmark	5.72	7.79	36.1	0.27	0.37	37.0
Pablo	5.81	6.68	15.0	0.82	0.96	17.1
Cantwell	8.60	11.71	36.2	0.34	0.38	11.7
Eudora	6.48	7.69	18.7	0.18	0.23	28.7
Swormville	9.27	12.14	30.9	0.31	0.42	35.5
Elmira	5.51	7.15	29.7	0.28	0.46	64.2

TABLE III: Comparison of path lengths—Shortest (—) as in Fig. 5b) and Ours (—) as in Fig. 5b—and corresponding AV@R values across environments.

through a NBV strategy optimized on the SE(3) manifold. By quantifying risk directly from the 3DGS representation, the robot can avoid unsafe regions while continuously replanning around local subgoals. This tight integration of mapping, risk assessment, SE(3)-based viewpoint optimization, and planning enables navigation that is both safe and efficient, while preserving reconstruction quality where it matters most. Our experiments demonstrate that the framework delivers reliable, high-fidelity navigation in complex environments, highlighting its potential for real-world deployment.

REFERENCES

- [1] G. Liu, W. Jiang, B. Lei, V. Pandey, K. Daniilidis, and N. Motee, “Beyond uncertainty: Risk-aware active view acquisition for safe robot navigation and 3d scene understanding with fisherrf,” *arXiv preprint arXiv:2403.11396*, 2024, v2.
- [2] T. Chen, O. Shorinwa, J. Bruno, A. Swann, J. Yu, W. Zeng, K. Nagami, P. Dames, and M. Schwager, “Splat-nav: Safe real-time robot navigation in gaussian splatting maps,” *IEEE Transactions on Robotics*, vol. 41, pp. 2765–2784, 2025.
- [3] T. Chen, A. Swann, J. Yu, O. Shorinwa, R. Murai, M. K. III, and M. Schwager, “Safer-splat: Safety with control barrier functions in online gaussian splatting maps,” 2024.
- [4] W. Jiang, B. Lei, and K. Daniilidis, “Ag-slam: Active gaussian splatting slam,” *arXiv preprint arXiv:2410.17422*, 2024.
- [5] Y. Chen, S. Huang, and R. Fitch, “Active slam for mobile robots with area coverage and obstacle avoidance,” *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1182–1192, 2020.
- [6] H. Carrillo, I. Reid, and J. A. Castellanos, “On the comparison of uncertainty criteria for active slam,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2080–2087.
- [7] M. L. Rodríguez-Arévalo, J. Neira, and J. A. Castellanos, “On the importance of uncertainty representation in active SLAM,” *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 829–834, 2018.
- [8] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, “Gaussian Splatting SLAM,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- [9] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 42, no. 4, 2023.
- [10] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, “Gaussian splatting SLAM,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [11] J. T. Kajiya and B. P. Von Herzen, “Ray tracing volume densities,” *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 165–174, 1984.
- [12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 405–421.
- [13] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, “Plenoxels: Radiance fields without neural networks,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5501–5510.
- [14] M. Turkulainen, X. Ren, I. Melekhov, O. Seiskari, E. Rahtu, and J. Kannala, “Dn-splatter: Depth and normal priors for gaussian splatting and meshing,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.
- [15] R. T. Rockafellar and S. Uryasev, “Conditional value-at-risk for general loss distributions,” *Journal of Banking & Finance*, vol. 26, no. 7, pp. 1443–1471, 2002.
- [16] S. Sarykalin, G. Serraino, and S. Uryasev, “Value-at-risk vs. conditional value-at-risk in risk management and optimization,” in *State-of-the-Art Decision-Making Tools in the Information-Intensive Age*. INFORMS, 2008, pp. 270–294.
- [17] W. Jiang, B. Lei, and K. Daniilidis, “Fisherrf: Active view selection and mapping with radiance fields using fisher information,” in *European Conference on Computer Vision (ECCV)*. Springer, 2024.
- [18] J. Wilson, M. Almeida, S. Mahajan, M. Labrie, M. Ghaffari, O. Ghasemalazadeh, M. Sun, C.-H. Kuo, and A. Sen, “Pop-gs: Next best view in 3d-gaussian splatting with p-optimality,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [19] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, “Habitat: A platform for embodied AI research,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9339–9347.
- [20] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, “Gibson Env: Real-world perception for embodied agents,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9068–9079.
- [21] A. Hore and D. Ziou, “Image quality metrics: Psnr vs. ssim,” in *20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 2366–2369.
- [22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [23] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 586–595.
- [24] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. SSC-4, no. 2, pp. 100–107, 1968.