

# Provably Safe Stein Variational Clarity-Aware Informative Planning

**Kaleb Ben Naveed\***

*Department of Robotics, University of Michigan, Ann Arbor*

KBNAVEED@UMICH.EDU

**Utkrisht Sahai\***

*Department of Robotics, University of Michigan, Ann Arbor*

USAHAI@UMICH.EDU

**Anouck Girard**

*Embry-Riddle Aeronautical University*

GIRARDA3@ERAU.EDU

**Dimitra Panagou**

*Department of Robotics, Department of Aerospace Engineering, University of Michigan, Ann Arbor*

DPANAGOU@UMICH.EDU

**Editors:** G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

## Abstract

Autonomous robots are increasingly deployed for information-gathering tasks in environments that vary across space and time. Planning informative and safe trajectories in such settings is challenging because information decays when regions are not revisited. Most existing planners model information as static or uniformly decaying, ignoring environments where the decay rate varies spatially; those that model non-uniform decay often overlook how it evolves along the robot’s motion, and almost all treat safety as a soft penalty. In this paper, we address these challenges. We model uncertainty in the environment using clarity, a normalized representation of differential entropy from our earlier work that captures how information improves through new measurements and decays over time when regions are not revisited. Building on this, we present Stein Variational Clarity-Aware Informative Planning, a framework that embeds clarity dynamics within trajectory optimization and enforces safety through a low-level filtering mechanism based on our earlier *gatekeeper* framework for safety verification. The planner performs Bayesian inference-based learning via Stein variational inference, refining a distribution over informative trajectories while filtering each nominal Stein informative trajectory to ensure safety. Hardware experiments and simulations across environments with varying decay rates and obstacles demonstrate consistent safety and reduced information deficits. [Paper Website]<sup>1</sup>.

**Keywords:** Safe Informative Planning, Variational Inference

## 1. Introduction

Autonomous robots are increasingly used in environmental monitoring (Hitz et al., 2017; Naveed et al., 2025b), search and rescue (Li et al., 2023), and 3D reconstruction (Kompis et al., 2021), where they must actively gather information to estimate unknown fields such as wind, temperature, or gas concentration. Unlike passive sensing, this requires *informative planning*, a sequential process that determines where and when to sample to reduce uncertainty. The main challenge is to plan trajectories that (i) account for spatially varying information decay rates in the environment, and (ii) remain provably safe with respect to obstacles while exploring informative regions. In this work, we focus on environments characterized by such spatially varying information decay, referred to as *stochastic spatiotemporal environments*.

---

\* Equal contribution.

1. Hardware and simulations experiment videos: [https://usahai18.github.io/stein\\_clarity/](https://usahai18.github.io/stein_clarity/)

Existing informative planning methods, including orienteering-based (Bottarelli et al., 2019), submodular (Meliou et al., 2007), sampling-based (Moon et al., 2025), and GP-based planners (Chen et al., 2022), maximize information gain but often assume static fields or use a constant global decay parameter. Ergodic exploration (Mathew and Mezić, 2011; Dressel and Kochenderfer, 2019) instead matches visitation frequency to a target distribution, but most variants ignore temporal evolution (Seewald et al., 2024; Dong et al., 2025; Lee et al., 2024). The spatiotemporal extension in (Naveed et al., 2024) accounts for temporal effects, yet computes the target distribution independently of the robot’s motion and thus misses uncertainty evolution along the trajectory. Finally, existing methods generally lack provable safety guarantees: penalty-based approaches (Lee et al., 2024) do not ensure constraint satisfaction, while CBF-based methods (Ames et al., 2016; Dong et al., 2025), though formally safe, are difficult to design and tune in high-dimensional systems.

We address these challenges with a provably safe, clarity-aware Stein variational informative planning framework. Clarity, introduced in our earlier work (Agrawal and Panagou, 2023), rescales differential entropy to  $[0, 1]$ , making uncertainty easier to interpret and more computationally tractable. Using clarity, we model how information decays and regenerates over time, enabling the robot to reason about the spatiotemporal evolution of uncertainty. We then define a differentiable clarity-based objective and optimize it within a Stein variational inference formulation (Liu and Wang, 2016; Lambert et al., 2020), a particle-based Bayesian learning method that operates in function space to approximate the posterior distribution over trajectories through gradient flows. This formulation allows multiple trajectory candidates to evolve in parallel, enabling the planner to learn and maintain a diverse set of informative motion strategies. To ensure safety, we integrate the `gatekeeper` framework (Agrawal et al., 2024; Naveed et al., 2025a), where each Stein particle generates a nominal trajectory concatenated with a backup trajectory to form a *candidate trajectory*. A candidate is committed if forward propagation verifies that all states remain in the safe set and the backup terminates in the backup set. Among safe candidates, the lowest-cost one is executed. Compared with existing methods, our **contributions** are twofold: **(i)** We develop a clarity-aware Stein variational informative planner that approximates a posterior distribution over informative trajectories, while accounting for spatially varying information decay and the evolution of clarity along the robot trajectory. **(ii)** We make the planner provably safe via a low-level `gatekeeper`-based filter that verifies nominal Stein trajectories and executes only certified-safe ones.

**Notation:** Let  $\mathbb{N} = \{0, 1, 2, \dots\}$ . Let  $\mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{R}_{> 0}$  be the set of reals, non-negative reals, and positive reals respectively. Let  $\mathcal{S}_{++}^n$  denote set of symmetric positive-definite in  $\mathbb{R}^{n \times n}$ . Let  $\mathcal{N}(\mu, \Sigma)$  denote a normal distribution with mean  $\mu$  and covariance  $\Sigma \in \mathcal{S}_{++}^n$ . The  $Q \in \mathcal{S}_{++}^n$  norm of a vector  $x \in \mathbb{R}^n$  is denoted  $\|x\|_Q = \sqrt{x^T Q x}$ . Let  $\mathcal{C}^i$  denote the space of functions that are  $i$  times continuously differentiable with respect to their arguments.

## 2. Preliminaries

### 2.1. Dynamics Model

Consider the continuous-time robot dynamics

$$\dot{x} = f(t, x, u), \tag{1}$$

with the state  $x \in \mathcal{X} \subset \mathbb{R}^n$ , the input  $u \in \mathcal{U} \subset \mathbb{R}^m$ , and vector field  $f : \mathbb{R} \times \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}^n$ . We assume  $f$  is locally Lipschitz in  $x$  and  $u$  and continuously differentiable in  $(x, u)$  on the domain

$\mathcal{X} \times \mathcal{U}$ . Given a feedback policy  $u = \pi(t, x)$  with  $\pi$  continuous in  $t$  and locally Lipschitz in  $x$ , the closed-loop system admits a unique solution over some interval.

**Definition 1 (Trajectory)** Let  $\mathcal{T} = [t_i, t_f] \subset \mathbb{R}$ . A trajectory is a pair of functions  $\xi = (\xi_x : \mathcal{T} \rightarrow \mathcal{X}, \xi_u : \mathcal{T} \rightarrow \mathcal{U})$ , satisfying  $\dot{\xi}_x(t) = f(t, \xi_x(t), \xi_u(t))$ ,  $\forall t \in (t_i, t_f)$ ,  $\xi_x(t_i) = x_i$ . The set of all trajectories defined over the interval  $\mathcal{T} = [t_i, t_f]$  and starting from  $(t, x) \in \mathbb{R} \times \mathcal{X}$  is denoted  $\Phi(t, x) = \{\xi = (\xi_x, \xi_u) : \xi_x(t) = x \text{ and } (\xi_x, \xi_u) \text{ is a trajectory}\}$ .

Let  $\mathcal{S} : \mathbb{R} \rightrightarrows \mathcal{X}$  denote the set of states satisfying the constraints. The system satisfies the constraints if  $x(t) \in \mathcal{S}(t), \forall t \geq t_0$ .

## 2.2. Clarity

We use clarity, introduced in [Agrawal and Panagou \(2023\)](#), to quantify uncertainty based on differential entropy.

**Definition 2** The differential entropy  $h[Z] \in (-\infty, \infty)$  of a continuous random variable  $Z$  with support  $S$  and density  $\rho : S \rightarrow \mathbb{R}$  is

$$h[Z] = - \int_S \rho(z) \log \rho(z) dz. \quad (2)$$

As the uncertainty in  $Z$  increases, the entropy approaches  $h[Z] \rightarrow \infty$ . Clarity is defined in terms of differential entropy.

**Definition 3** Let  $Z$  be an  $n$ -dimensional continuous random variable with differential entropy  $h[Z]$ . The clarity of  $Z$  is a normalized quantity  $q[Z] \in (0, 1)$  of  $Z$  is defined as

$$q[Z] = \left( 1 + \frac{\exp(2h[Z])}{(2\pi e)^n} \right)^{-1}. \quad (3)$$

In other words, the clarity  $q[Z]$  of a random variable  $Z$  lies in the interval  $(0, 1)$ , where  $q[Z] \rightarrow 0$  corresponds to the case where the uncertainty in  $Z$  is infinite, and  $q[Z] \rightarrow 1$  corresponds to the case where  $Z$  is perfectly known. See *Example 1* in [Agrawal and Panagou \(2023\)](#).

## 2.3. Stein Variational Trajectory Optimization (SVTO)

Model Predictive Control (MPC) computes control inputs by solving a finite-horizon optimization problem over a time interval  $[t_i, t_f]$  given an initial state  $x_i$ :

$$\min_{(\xi_x, \xi_u) \in \Phi(t_i, x_i)} C(\xi_x, \xi_u), \quad (4)$$

where  $C : \Phi(t_i, x_i) \rightarrow \mathbb{R}$  is the trajectory cost functional, and  $\Phi(t_i, x_i)$  is the set of admissible trajectories originating from the initial state  $x_i$  at time  $t_i$  and satisfying the system dynamics and constraints defined in *Def. 1*. Classical MPC returns a single deterministic optimum, which limits its ability to reason over multi-modal solutions that arise from nonconvex costs or constraints.

**Bayesian View of Trajectory Optimization.** SVTO reformulates (4) as a Bayesian inference problem over control trajectories  $\xi_u$ , whose state trajectories  $\xi_x$  satisfy  $(\xi_x, \xi_u) \in \Phi(t_i, x_i)$ . Let  $\mathcal{O}_\xi \in \{0, 1\}$  be a binary random variable that indicates whether a control trajectory  $\xi_u$  is optimal ( $\mathcal{O}_\xi = 1$ ) with respect to the cost  $C(\xi_x, \xi_u)$ . By Bayes' rule, the posterior distribution over feasible control trajectories, conditioned on the current state  $x_i$ , is given by

$$p(\xi_u \mid \mathcal{O}_\xi = 1, x_i) = \frac{p(\mathcal{O}_\xi = 1 \mid \xi_u, x_i) p(\xi_u)}{p(\mathcal{O}_\xi = 1 \mid x_i)}. \quad (5)$$

Here,  $p(\xi_u \mid \mathcal{O}_\xi = 1, x_i)$  is the posterior over trajectories given that they are optimal,  $p(\mathcal{O}_\xi = 1 \mid \xi_u, x_i)$  is the likelihood,  $p(\xi_u)$  is the prior over admissible control trajectories in  $\Phi(t_i, x_i)$ , and  $p(\mathcal{O}_\xi = 1 \mid x_i)$  is the marginal. The likelihood is modeled as an exponentiated cost function:

$$p(\mathcal{O}_\xi = 1 \mid \xi_u, x_i) \propto \exp(-\alpha C(\xi_x, \xi_u)), \quad \alpha > 0, \quad (6)$$

where  $\alpha$  controls the sharpness of the likelihood. This formulation assigns higher posterior probability to control trajectories with lower cost.

**Variational Approximation.** The posterior (5) can be approximated by minimizing the Kullback–Leibler (KL) divergence

$$q^* = \arg \min_{q \in \mathcal{Q}} D_{\text{KL}}(q(\xi_u) \parallel p(\xi_u \mid \mathcal{O}_\xi = 1, x_i)), \quad (7)$$

where  $\mathcal{Q}$  denotes the family of admissible trajectory distributions, and  $q(\xi_u)$  is an empirical distribution represented by a set of  $K$  control-trajectory particles  $\{\xi_u^{(k)}\}_{k=1}^K$ . Each particle  $\xi_u^{(k)}$  corresponds to a control sequence defined over the horizon  $[t_i, t_f]$ . The state trajectory  $\xi_x^{(k)}$  is obtained by integrating the system dynamics (1) from the current state  $x_i$  under the control sequence  $\xi_u^{(k)}$ .

**Stein Variational Gradient Update.** Stein Variational Gradient Descent (SVGD) (Liu and Wang, 2016) deterministically transports a set of particles to minimize the KL divergence between their empirical distribution and the target posterior. Each particle here represents a feasible control trajectory  $\xi_u^{(k)}$ , which is iteratively updated according to

$$\xi_u^{(k)} \leftarrow \xi_u^{(k)} + \varepsilon \phi^*(\xi_u^{(k)}), \quad (8)$$

where  $\varepsilon > 0$  is the step size. The optimal update field  $\phi^*(\cdot)$  defines the direction that maximally decreases the KL divergence and is given by

$$\phi^*(\xi_u^{(k)}) = \frac{1}{K} \sum_{j=1}^K \left[ k(\xi_u^{(j)}, \xi_u^{(k)}) \nabla_{\xi_u^{(j)}} \log p(\xi_u^{(j)} \mid x_i) + \nabla_{\xi_u^{(j)}} k(\xi_u^{(j)}, \xi_u^{(k)}) \right], \quad (9)$$

where  $k(\cdot, \cdot)$  is a positive-definite kernel defining the Reproducing Kernel Hilbert Space (RKHS) used to compute  $\phi^*$ . In the first term, the kernel weights each particle's gradient by its similarity to others, pulling them toward high-posterior regions; the second term introduces a repulsive force that preserves diversity in the RKHS. The log-posterior gradient is computed as

$$\nabla_{\xi_u^{(j)}} \log p(\xi_u^{(j)} \mid \mathcal{O}_\xi = 1, x_i) = -\alpha \nabla_{\xi_u^{(j)}} C(\xi_x^{(j)}, \xi_u^{(j)}) + \nabla_{\xi_u^{(j)}} \log p(\xi_u^{(j)}), \quad (10)$$

where the first term depends on the trajectory cost and the second encodes any prior belief over control trajectories (often taken uniform when no prior preference is imposed).

### 3. Problem Formulation

#### 3.1. Environment Specification

Consider the coverage space  $\mathcal{P}$ . We discretize the domain into a set of  $N_p$  cells each with size  $V$ .<sup>2</sup> Let  $m_p : [t_0, \infty) \rightarrow \mathbb{R}$  be the (time-varying) quantity of interest at each cell  $p \in \mathcal{P}_{\text{cells}} = \{1, \dots, N_p\}$ . We model the quantities of interest as independent stochastic processes:

$$\dot{m}_p = w_p(t), \quad w_p(t) \sim \mathcal{N}(0, Q_p), \quad (11a)$$

$$y_p = C_p(x)m_p + v_p(t), \quad v_p(t) \sim \mathcal{N}(0, R), \quad (11b)$$

where  $y_p \in \mathbb{R}$  is the output corresponding to cell  $p$ .  $R$  is the measurement noise variance, and  $Q_p \in \mathbb{R}_{>0}$  is the process noise variance at each cell  $p$ . Since the process noise  $Q_p$  varies across cells, the field values  $m_p$  evolve differently in time at each location, resulting in a *stochastic spatiotemporal environment*. To quantify the uncertainty in the environment, we define an independent clarity dynamics for each cell  $p \in \mathcal{P}_{\text{cells}}$ . The clarity dynamics derived in [Agrawal and Panagou \(2023\)](#) for each cell are given as follows:

$$\dot{q}_p = \frac{C_p(x)^2}{R}(1 - q_p)^2 - Q_p q_p^2 \quad (12)$$

where  $x$  is the state of the robot,  $C_p : \mathcal{X} \rightarrow \mathbb{R}$  is the mapping between robot state and sensor state at cell  $p \in \mathcal{P}$ , and  $R \in \mathbb{R}$  is the known variance of the measurement noise. Here,  $q_p \rightarrow 1$  represents the case when the state of the environment (e.g. smoke concentration) is perfectly known in the cell  $p$ , whereas lower values correspond to higher uncertainty.<sup>3</sup>

#### 3.2. Problem Statement

Consider a robotic system evolving under dynamics (1) within a stochastic spatiotemporal environment (11). Each cell  $p \in \mathcal{P}$  evolves with process noise  $Q_p$ , and its clarity state follows (12). A desired information level is specified as a *target clarity*  $\bar{q}_p < q_{\infty,p}$ , where  $q_{\infty,p}$  is the maximum attainable clarity, ensuring that the target is achievable within finite time. Clarity is used as the information metric because it naturally reflects both environmental stochasticity and sensor characteristics: (i) the decay term  $-Q_p q_p^2$  links the information decay rate directly to the process noise variance  $Q_p$ , and (ii) during sensing,  $q_p$  increases monotonically and saturates at  $q_{\infty,p} < 1$  for  $Q_p, R(x) > 0$ , guaranteeing an upper bound on attainable information.

The robot generates the informative trajectory in a receding-horizon manner: at each time  $t_i$ , it optimizes a feasible trajectory  $\xi = (\xi_x, \xi_u) \in \Phi(t_i, x_i)$  over a finite horizon  $[t_i, t_i + T_N]$ , executes the first control input, and replans at the next step. The optimization problem is formulated as:

$$\min_{\xi \in \Phi(t_i, x_i)} \frac{1}{N_p} \sum_{p=1}^{N_p} \int_{t_i}^{t_i+T_N} \max(0, \bar{q}_p - q_p(\tau)) d\tau \quad (13a)$$

$$\text{s.t. } \dot{\xi}_x = f(t, \xi_x(t), \xi_u(t)), \quad \forall t \in [t_i, t_i + T_N], \quad (13b)$$

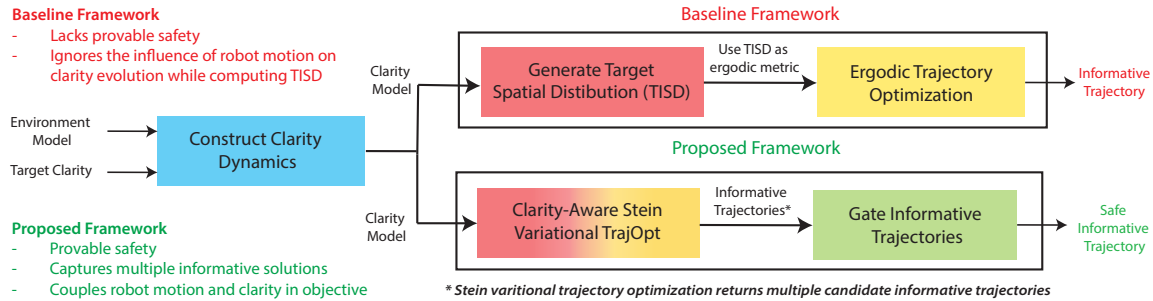
$$\dot{q}_p = g(\xi_x, q_p), \quad \forall p \in \mathcal{P}_{\text{cells}}, \quad (13c)$$

$$\xi_x(t) \in \mathcal{S}(t), \quad \forall t \in [t_i, t_i + T_N], \quad (13d)$$

$$\xi_x(t_i) = x_i \quad (13e)$$

2. Size is length in 1D, area in 2D, and volume in 3D.

3. In practice,  $Q_p$  captures temporal variability and can be estimated from historical data or physical models.



**Figure 1:** The baseline derives a TISD from the clarity model and uses it as an ergodic metric, whereas the proposed framework directly optimizes the clarity model via a Stein variational approach, producing multiple informative and provably safe trajectories.

where  $g(\cdot, \cdot)$  denotes the clarity dynamics (12), and  $\mathcal{S}(t) \subset \mathcal{X}$  defines the time-varying safe set. Problem (13) seeks a feasible trajectory  $\xi = (\xi_x, \xi_u)$  that respects system dynamics, clarity evolution, and safety constraints while minimizing the mean clarity deficit across the environment.

## 4. Methodology

### 4.1. Overview

The proposed framework, illustrated in Fig. 1, builds upon clarity-aware ergodic search by removing the multi-stage approximation introduced by the Target Information Spatial Distribution (TISD). In the baseline approach, the TISD for each cell is computed as the time required for its clarity to reach the target value assuming the robot is co-located with that cell. This neglects that the robot only influences clarity along its trajectory, breaking the coupling between motion and clarity evolution and yielding a static spatial objective that fails to reflect how information evolves during motion.

To address this, the proposed method directly couples the robot trajectory with the clarity dynamics through a Stein variational formulation. Multiple clarity-aware trajectories are evolved in parallel by minimizing a differentiable clarity-based cost that captures the spatiotemporal evolution of information. Safety is enforced by gating the nominal Stein trajectories, where each is paired with a backup trajectory, forward propagated, and evaluated for safety over the entire horizon. The safe candidate with the lowest cost is committed for execution.

### 4.2. Clarity-Aware Stein Variational Trajectory Optimization

The proposed framework performs trajectory optimization using a differentiable clarity-based cost. Since the Stein variational update requires gradients of the objective with respect to the trajectory, both the cost functional and the measurement model must be continuously differentiable.

**Differentiable Clarity-Aware Cost Functional.** The original mean clarity deficit objective (13a) contains a non-differentiable hinge term  $\max(0, \bar{q}_p - q_p)$ , which prevents analytic gradient computation through the clarity model. We replace this hinge with a smooth *softplus* surrogate:

$$\text{softplus}_\beta(z) = \frac{1}{\beta} \log(1 + e^{\beta z}), \quad \beta > 0, \quad (14)$$

where larger  $\beta$  values yield a closer approximation to  $\max(0, z)$  while maintaining differentiability. The resulting differentiable clarity-aware cost for a trajectory  $\xi = (\xi_x, \xi_u) \in \Phi(t, x)$  over the

horizon  $[t_i, t_i + T_N]$  is

$$J_\beta(\xi) = \frac{1}{N_p T_N} \sum_{p=1}^{N_p} \int_{t_i}^{t_i+T_N} \text{softplus}_\beta(\bar{q}_p - q_p(\tau; \xi_x)) d\tau, \quad (15)$$

where  $q_p(\tau; \xi_x)$  denotes the clarity at cell  $p$  obtained by integrating the clarity dynamics (12) along the trajectory  $\xi_x$ . The cost  $J_\beta(\xi)$  is continuously differentiable in  $\xi$  for any finite  $\beta > 0$ , allowing smooth gradient propagation through the clarity model and trajectory.

**Differentiable Measurement Model.** The measurement mapping  $C(x)$  determines how the robot's state influences clarity gain in each cell. To preserve differentiability of  $J_\beta(\xi)$ ,  $C(x)$  must be at least  $\mathcal{C}^1$  and locally Lipschitz in  $x$ . We model each cell-specific sensing footprint as a Gaussian field:

$$C_p(x) = \kappa_c \exp\left(-\frac{1}{2}\|x - \mu_p\|_{\Sigma_c^{-1}}^2\right), \quad (16)$$

where  $\mu_p \in \mathbb{R}^n$  is the center of cell  $p$ ,  $\Sigma_c \in \mathcal{S}_{++}^n$  defines the sensing footprint, and  $\kappa_c > 0$  scales the sensor strength. This ensures that  $C_p(x)$  and its gradient  $\nabla_x C_p(x)$  are continuous and bounded, enabling differentiable backpropagation through the clarity dynamics (12). Alternative smooth sensing models (e.g., radial basis or polynomial decay) are also admissible as long as  $C_p(x)$  remains continuously differentiable.

### 4.3. Safety filtering through gatekeeper framework

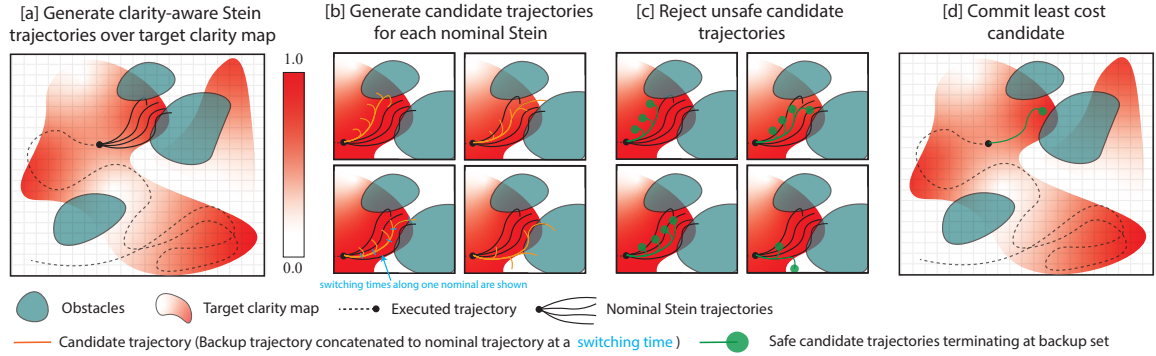
To ensure safety, we propose a method based on the gatekeeper framework (Agrawal et al., 2024) to filter and commit safe trajectories among the candidates generated by the Stein variational optimizer. At each decision time  $t_i$ , the optimizer produces  $K$  nominal trajectories, each evolved over a nominal horizon  $[t_i, t_i + T_N]$ . Along each nominal trajectory, multiple switching times are defined; at each switching time, a candidate trajectory is constructed by concatenating a backup trajectory to the nominal segment. All candidates are evaluated in parallel for safety and cost, ensuring that the executed trajectory remains within  $\mathcal{S}(t)$  for all  $t \geq t_0$ . Figure 2 illustrates the procedure for  $K = 4$ .

**Definition 4 (Nominal Trajectory)** *At time  $t_i$ , the Stein variational optimizer generates a set of nominal trajectories  $\Xi = \{\xi^{(k)}\}_{k=1}^K$ , where each  $\xi^{(k)} = (\xi_x^{(k)}, \xi_u^{(k)}) \in \Phi(t_i, x_i)$  is a trajectory evolved under the system dynamics (1) over the nominal horizon  $[t_i, t_i + T_N]$ .*

**Definition 5 (Backup Set)** *The backup set, denoted  $\mathcal{B}(t) \subseteq \mathcal{S}(t)$ , is a time-varying subset of the safe set such that there exists a backup controller  $\pi^b : \mathbb{R}_{\geq 0} \times \mathcal{X} \rightarrow \mathcal{U}$  for which the closed-loop dynamics  $\dot{x} = f(t, x, \pi^b(t, x))$  satisfy the forward-invariance condition*

$$x(t_0) \in \mathcal{B}(t_0) \Rightarrow x(t) \in \mathcal{B}(t), \quad \forall t \geq t_0. \quad (17)$$

Hence,  $\mathcal{B}(t)$  is the region from which safety is guaranteed under  $\pi^b$ .



**Figure 2:** Safety filtering through the `gatekeeper` framework. Candidate trajectories (shown in yellow) are generated in parallel for each nominal Stein trajectory. Unsafe candidate trajectories are rejected, and the least-cost safe candidate is executed.

**Definition 6 (Valid Backup Trajectory)** Given a nominal trajectory  $\xi^{(k)}$  and a switching time  $\tau_s \in [t_i, t_i + T_N]$ , a backup trajectory  $\xi^b = (\xi_x^b, \xi_u^b)$  is a feasible trajectory initialized from the nominal state at the switching time,  $\xi_x^b(\tau_s) = \xi_x^{(k)}(\tau_s)$ , and evolved over a backup horizon  $T_B$ . The trajectory  $\xi^b$  is valid if it remains within the safe set and terminates in the backup set (Def 5):

$$\xi_x^b(t) \in \mathcal{S}(t), \quad \forall t \in [\tau_s, \tau_s + T_B], \quad \xi_x^b(\tau_s + T_B) \in \mathcal{B}(\tau_s + T_B). \quad (18)$$

**Definition 7 (Candidate Trajectory)** Given a nominal trajectory  $\xi^{(k)} \in \Phi(t_i, x_i)$  and a switching time  $\tau_s \in [t_i, t_i + T_N]$ , a candidate trajectory  $\xi^{c(k,s)} \in \Phi(t_i, x_i)$  is defined over the horizon  $[t_i, \tau_s + T_B]$  as the concatenation of (1) the nominal segment, initialized at  $x(t_i) = x_i$  and propagated under  $\xi^{(k)}$  up to  $\tau_s$ , and (2) a backup trajectory  $\xi^b$ , initialized at  $\xi_x^b(\tau_s) = \xi_x^{(k)}(\tau_s)$  and generated for a backup horizon  $T_B$ :

$$\xi^{c(k,s)}(t) = \begin{cases} \xi^{(k)}(t), & t \in [t_i, \tau_s], \\ \xi^b(t), & t \in [\tau_s, \tau_s + T_B]. \end{cases} \quad (19)$$

**Definition 8 (Safe Candidate Trajectory)** A candidate trajectory  $\xi^{c(k,s)}$  is safe if all its states remain within the safe set and its concatenated backup is valid by Def 6:

$$\xi_x^{c(k,s)}(t) \in \mathcal{S}(t), \quad \forall t \in [t_i, \tau_s + T_B], \quad \text{and} \quad \xi^b \text{ is valid by Def 6} \quad (20)$$

Let  $\Phi_{\text{safe}}(t_i, x_i)$  denote the set of all safe candidates.

**Definition 9 (Committed Trajectory)** The least-cost safe candidate is committed for execution:

$$\xi^* = \arg \min_{\xi^{c(k,s)} \in \Phi_{\text{safe}}(t_i, x_i)} J_{\beta}(\xi^{c(k,s)}). \quad (21)$$

If no safe candidate exists, the robot continues to follow the previously committed trajectory.

**Theorem 10** Assuming that at time  $t_0$  a committed trajectory is available, and that at subsequent planning times  $\{t_0, t_1, \dots, t_i, \dots\}$  the committed trajectories are constructed according to Def. 9, the closed-loop state remains safe for all future time, i.e.,

$$x(t) \in \mathcal{S}(t), \quad \forall t \geq t_0. \quad (22)$$

**Proof** The proof is provided in Appendix B. ■

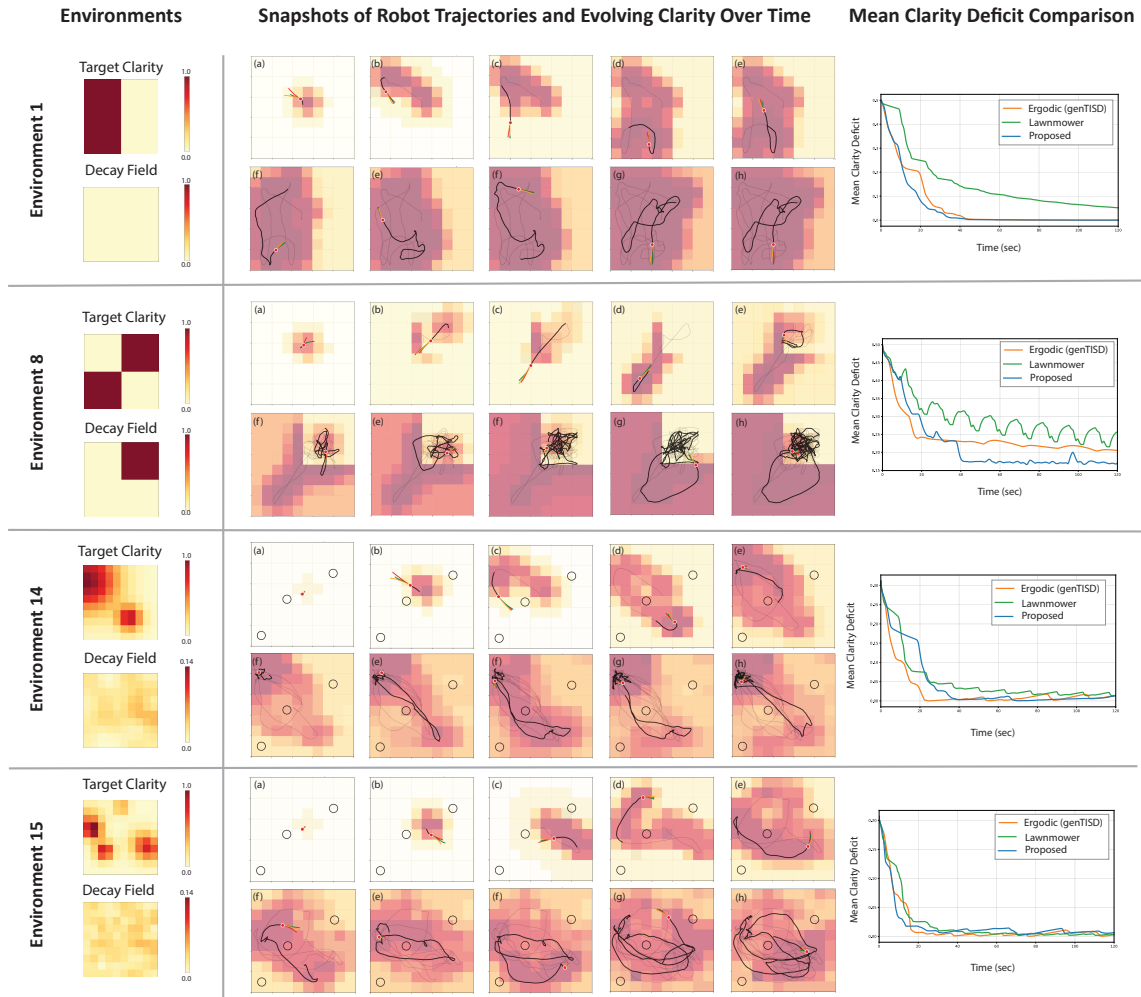


Figure 3: Planner behavior across representative environments. The planner maintains low clarity deficits and safe motion.

## 5. Results & Discussion

To evaluate the framework, we consider sixteen simulation environments with different target clarity fields, decay patterns, and obstacle layouts. These test the planner’s ability to explore informative regions, maintain clarity, and remain safe. We present four representative cases, with all sixteen shown in Appendix C, and include hardware experiments in Appendix A. In our implementation, the backup controller uses bounded deceleration to stop safely, and  $\mathcal{B}(t)$  contains states from which this maneuver is safe. Among feasible candidates, the gatekeeper commits the lowest-cost trajectory.

### 5.1. Planner Behavior Across Environments

**Environment 1:** This environment has a static field with no information decay since the process noise is zero. As shown in Fig. 3, the left half of the grid has higher target clarity while the right half has zero. The robot first moves to the high target clarity region and increases clarity until the deficit approaches zero and the objective provides no incentive to remain there, allowing the robot to move freely across the grid. Since there is no decay, previously explored areas do not require revisiting.

**Table 1:** Comparison of Methods Across Experiments

Method	No. of Experiments	Mean Safety Violations (%)
Proposed (without gating)	100 across 4 envs	2.94
Proposed (with gating)	100 across 4 envs	0.0

**Environment 8:** This environment contains variable decaying regions Fig. 3. The top-right and bottom-left quadrants have a target clarity of 1, while the others are 0. Decay is only in the top-right part. The robot first moves to the bottom-left, reaching the target clarity quickly since there is no decay. It then moves to the top-right, where decay causes clarity to decrease over time. The robot remains there to minimize the clarity deficit, showing how the planner adapts to temporal decay.

**Environments 14 and 15:** These environments contain obstacles near high-clarity regions (Fig 3) and test the planner’s ability to explore informative areas safely. The robot adapts its motion to the clarity deficit, reaching high-clarity regions while avoiding collisions. The safety mechanism keeps trajectories within the safe set as the planner minimizes clarity deficit. More results in Appendix C.

## 5.2. Comparison of Mean Clarity Deficit and Safety Verification

Figure 3 compares the mean clarity deficit across methods, while table 1 reports safety performance. The lawnmower strategy performs worst, following a fixed sweeping pattern that cannot adapt to high clarity-deficit regions. Both the ergodic and Stein variational planners achieve lower deficits by redistributing coverage as the field evolves. Their information-gathering performance is comparable; however, the Stein approach is far more efficient. Generating Stein trajectories over a 6 s horizon takes only 70 ms with particle-based parallelization in JAX, while the `gatekeeper` safety check adds about 40 ms. In contrast, computing an ergodic trajectory of the same horizon via nonlinear optimization takes about 0.7 s.

For the safety evaluation in table 1, we used four environments with obstacles and ran 100 randomized trials varying the robot’s initial position and obstacle size and location. Without the `gatekeeper`, soft-penalty enforcement led to an average of 2.94 % safety violations, denoting the fraction of time in collision. With the `gatekeeper` enabled, all runs were collision-free (0.0 %), confirming constraint satisfaction without loss of real-time performance.

## 6. Conclusion

We proposed a Stein variational clarity-aware informative planner that integrates Bayesian learning of informative trajectories with safety verification. Clarity, derived from differential entropy, provides a normalized representation of information quality that varies across space and time. Its dynamics capture how uncertainty decays as the robot moves through the environment. The Stein variational formulation learns diverse trajectory distributions by coupling motion with clarity dynamics, while the `gatekeeper` ensures safety through real-time filtering. Simulations demonstrate safe operation, computational efficiency, and reduced information deficits. Future work will extend the framework to multi-robot settings and online adaptation under unknown process noise.

## Acknowledgments

This work has been partially supported by the National Science Foundation (NSF) under Award Numbers 1942907 and 2223845.

## References

- Devansh R Agrawal and Dimitra Panagou. Sensor-based planning and control for robotic systems: Introducing clarity and perceivability. *IEEE Control Systems Letters*, 7:2623–2628, 2023.
- Devansh Ramgopal Agrawal, Ruichang Chen, and Dimitra Panagou. gatekeeper: Online safety verification and control for nonlinear systems in dynamic environments. *IEEE Transactions on Robotics*, 2024.
- Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8): 3861–3876, 2016.
- Lorenzo Bottarelli, Manuele Bicego, Jason Blum, and Alessandro Farinelli. Orienteering-based informative path planning for environmental monitoring. *Engineering Applications of Artificial Intelligence*, 77:46–58, 2019.
- Weizhe Chen, Roni Khardon, and Lantao Liu. AK: Attentive Kernel for Information Gathering. In *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022. doi: 10.15607/RSS.2022.XVIII.047.
- Dayi Ethan Dong, Henry Berger, and Ian Abraham. Time-optimal ergodic search: Multiscale coverage in minimum time. *The International Journal of Robotics Research*, 44(10-11):1664–1683, 2025.
- Louis Dressel and Mykel J Kochenderfer. Tutorial on the generation of ergodic trajectories with projection-based gradient descent. *IET Cyber-Physical Systems: Theory & Applications*, 4(2): 89–100, 2019.
- Gregory Hitz, Enric Galceran, Marie-Ève Garneau, François Pomerleau, and Roland Siegwart. Adaptive continuous-space informative path planning for online environmental monitoring. *Journal of Field Robotics*, 34(8):1427–1449, 2017.
- Yves Kompis, Luca Bartolomei, Ruben Mascaro, Lucas Teixeira, and Margarita Chli. Informed sampling exploration path planner for 3d reconstruction of large scenes. *IEEE Robotics and Automation Letters*, 6(4):7893–7900, 2021.
- Alexander Lambert, Adam Fishman, Dieter Fox, Byron Boots, and Fabio Ramos. Stein variational model predictive control. *arXiv preprint arXiv:2011.07641*, 2020.
- Darrick Lee, Cameron Lerch, Fabio Ramos, and Ian Abraham. Stein variational ergodic search. *arXiv preprint arXiv:2406.11767*, 2024.

- Yue Li, Yan Gao, Sijie Yang, and Quan Quan. Swarm robotics search and rescue: A bee-inspired swarm cooperation approach without information exchange. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1127–1133. IEEE, 2023.
- Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.
- George Mathew and Igor Mezić. Metrics for ergodicity and design of ergodic dynamics for multi-agent systems. *Physica D: Nonlinear Phenomena*, 240(4-5):432–442, 2011.
- Alexandra Meliou, Andreas Krause, Carlos Guestrin, and Joseph M Hellerstein. Nonmyopic informative path planning in spatio-temporal models. In *AAAI*, volume 10, pages 16–7, 2007.
- Brady Moon, Nayana Suvarna, Andrew Jong, Satrajit Chatterjee, Junbin Yuan, Muqing Cao, and Sebastian Scherer. Ia-tigris: An incremental and adaptive sampling-based planner for online informative path planning. *arXiv preprint arXiv:2502.15961*, 2025.
- Kaleb Ben Naveed, Devansh Agrawal, Christopher Vermillion, and Dimitra Panagou. Eclares: Energy-aware clarity-driven ergodic search. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14326–14332. IEEE, 2024.
- Kaleb Ben Naveed, Devansh R Agrawal, Daniel M Cherenon, Haejoon Lee, Alia Gilbert, Hardik Parwana, Vishnu S Chipade, William Bentz, and Dimitra Panagou. Enabling safety for aerial robots: Planning and control architectures. *arXiv preprint arXiv:2504.08601*, 2025a.
- Kaleb Ben Naveed, Haejoon Lee, and Dimitra Panagou. Multi-robot allocation for information gathering in non-uniform spatiotemporal environments. *arXiv preprint arXiv:2509.22883*, 2025b.
- Adam Seewald, Cameron J Lerch, Marvin Chancán, Aaron M Dollar, and Ian Abraham. Energy-aware ergodic search: Continuous exploration for multi-agent systems with battery constraints. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7048–7054. IEEE, 2024.

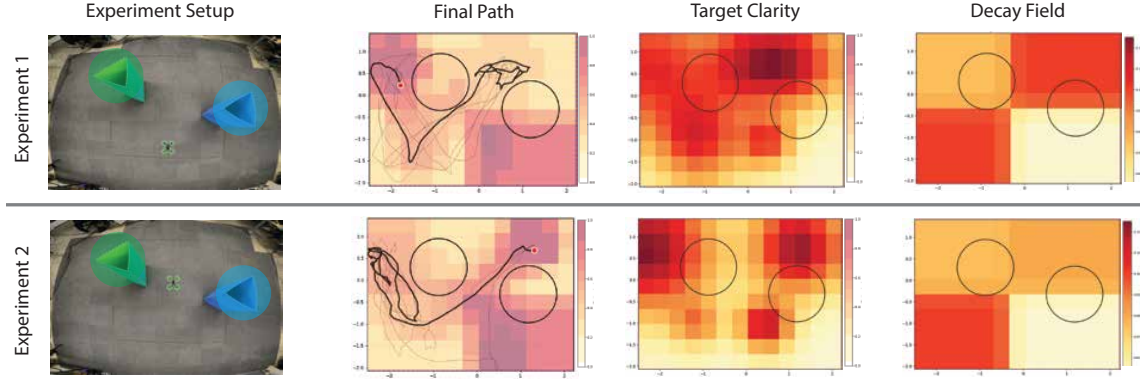


Figure 4: Hardware Experiments

## Appendix A. Hardware Experiments

Two environments with different configurations of the target clarity and decay field were demonstrated. Experiments were conducted in an indoor arena with 17 Vicon cameras for state estimation. During planning, the physical obstacles were modeled as circular objects with appropriate padding. Stein trajectories with a 6 s horizon were generated in a receding-horizon manner for the double-integrator model and tracked by the quadrotor using differential flatness. The average computation time per replanning step, including Stein trajectory generation and safety verification, was 100 ms. The experiment videos can be found here: [\[Experiment Videos\]](#)<sup>4</sup>.

## Appendix B. Proof of Theorem 10

**Proof** We prove the result by induction over the sequence of planning times  $t_0, t_1, \dots, t_k, \dots$ .

*Base case ( $t_0$ ):* By assumption, a committed trajectory  $\xi_0^{\text{com}}$  is available at time  $t_0$ . Since  $\xi_0^{\text{com}} \in \Phi_{\text{safe}}(t_0, x(t_0))$  by construction, its numerically integrated state satisfies  $x(t) \in S(t)$  for all  $t \in [t_0, t_1]$ . Hence the system is safe during the first execution interval.

*Inductive step:* Assume that for some  $k \geq 0$ , the system remains safe over  $[t_k, t_{k+1}]$  and that a committed trajectory  $\xi_k^{\text{com}}$  is available at  $t_k$ . At time  $t_{k+1}$ , the next committed trajectory  $\xi_{k+1}^{\text{com}}$  is constructed according to Def. 9.

If  $\Phi_{\text{safe}}(t_{k+1}, x(t_{k+1})) \neq \emptyset$ , then  $\xi_{k+1}^{\text{com}}$  is a validated safe trajectory whose state remains in  $S(t)$  for all  $t \in [t_{k+1}, t_{k+2}]$ . Otherwise, by the recursive property of Def. 9, the system continues executing the previously committed trajectory  $\xi_k^{\text{com}}$ , which has already been validated to remain safe over its remaining horizon. In both cases,  $x(t) \in S(t)$  for all  $t \in [t_{k+1}, t_{k+2}]$ .

By induction,  $x(t) \in S(t)$  for all  $t \geq t_0$ , establishing the global safety of the closed-loop system.  $\blacksquare$

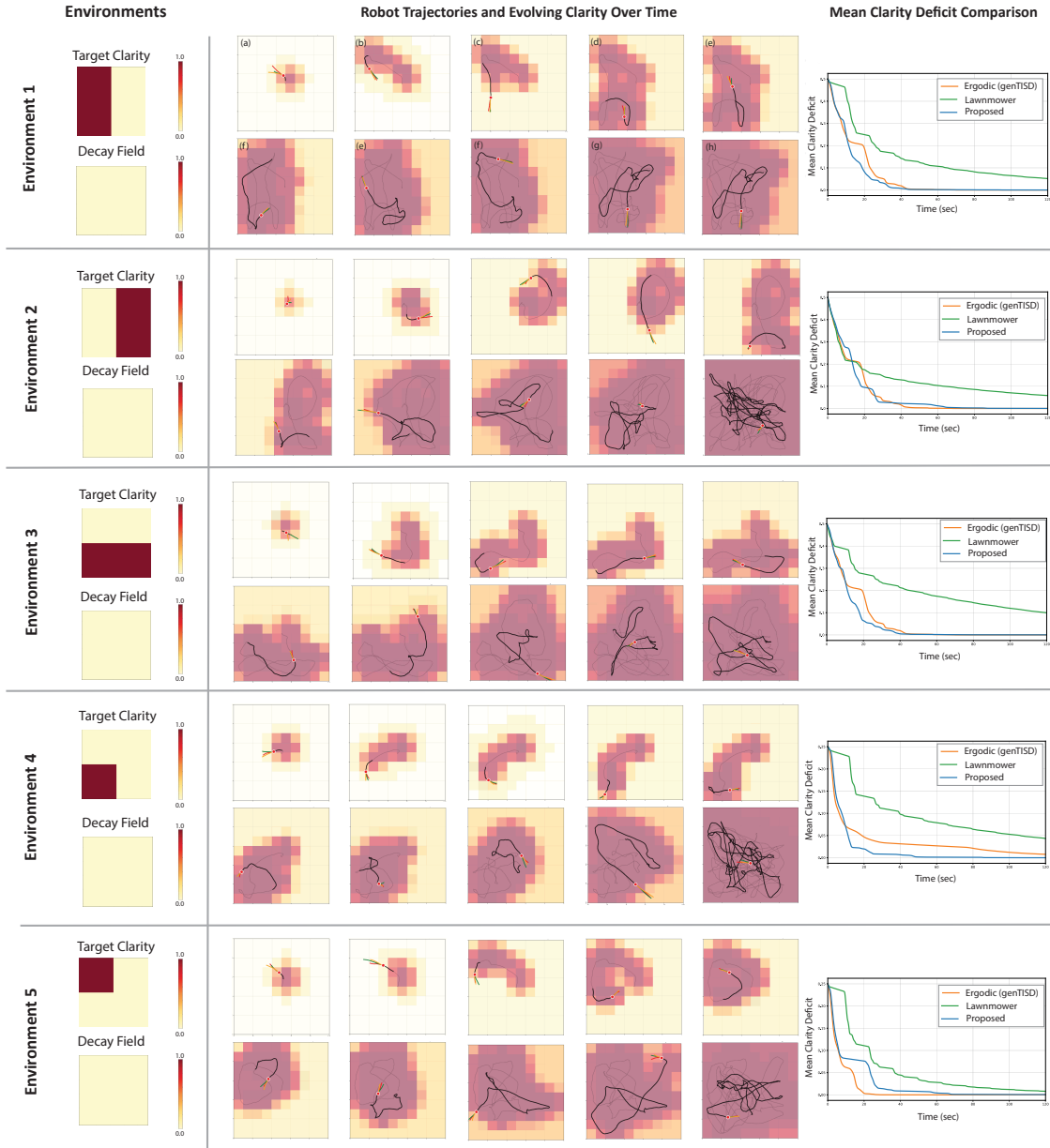


Figure 5

## Appendix C. More Simulation Results

### C.1. Environments

**Environment 1:** This environment has a static field with no information decay, as the process noise is zero. As shown in Fig. 5, the left half of the grid has higher target clarity, while in the right half its zero. The robot first moves to the high target clarity region and increases clarity until the target is reached. Once the clarity deficit in that region approaches zero, the objective provides no

4. Experiment Videos: [https://usahail8.github.io/stein\\_clarity/](https://usahail8.github.io/stein_clarity/)

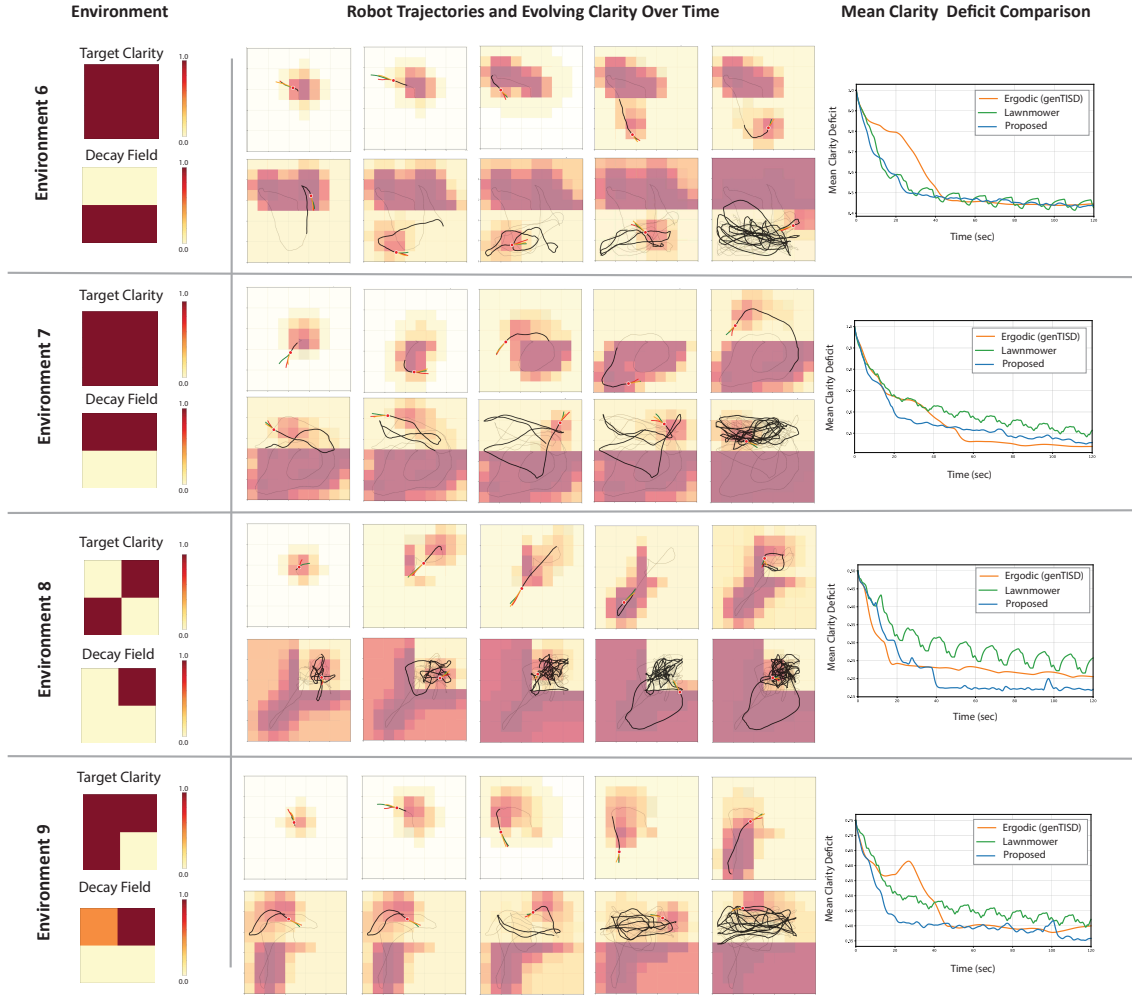


Figure 6

further incentive to remain there, allowing the robot to move freely across the grid. Since there is no decay, it does not need to revisit previously explored areas.

**Environment 2:** This environment has a static field with no information decay, as the process noise is zero. As shown in Fig. 5, the right half of the grid has higher target clarity, while in the left half its zero. The robot first moves to the high target clarity region and increases clarity until the target is reached. Once the clarity deficit in that region approaches zero, the objective provides no further incentive to remain there, allowing the robot to move freely across the grid. Since there is no decay, it does not need to revisit previously explored areas.

**Environment 3:** This environment has a static field with no information decay, as the process noise is zero. As shown in Fig. 5, the bottom half of the grid has higher target clarity, while in the top half its zero. The robot first moves to the bottom half and increases clarity until the clarity deficit in the region approaches zero; the objective provides no further incentive to remain there; the robot

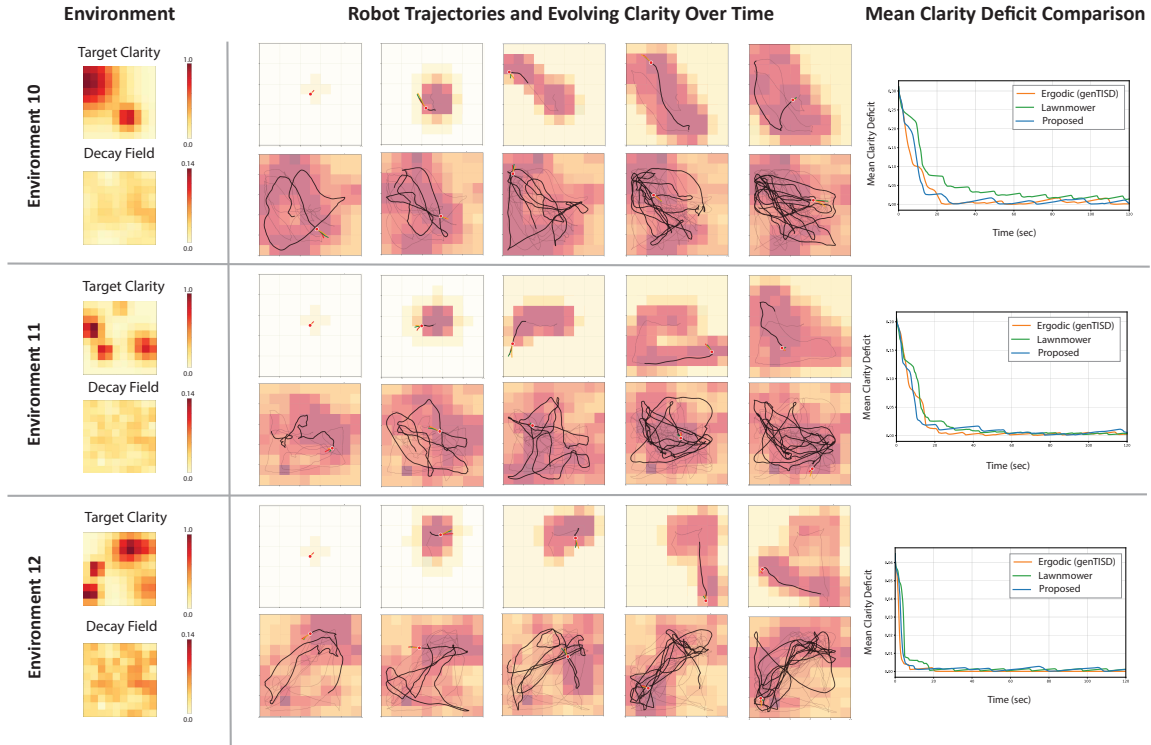


Figure 7

then moves freely across the grid so it explores the top half as well . Since there is no decay, it does not need to revisit previously explored areas.

**Environment 3:** This environment has a static field with no information decay, as the process noise is zero. As shown in Fig. 5, the bottom half of the grid has higher target clarity, while in the top half its zero. The robot first moves to the bottom half and increases clarity until the clarity deficit in the region approaches zero; the objective provides no further incentive to remain there; the robot then moves freely across the grid so it explores the top half as well . Since there is no decay, it does not need to revisit previously explored areas.

**Environment 4:** This environment has a static field with no information decay, as the process noise is zero. As shown in Fig. 5, the bottom left quadrant of the grid has a higher target clarity, while its zero elsewhere. The robot first moves to the bottom left and increases clarity until the clarity deficit approaches zero; the objective provides no further incentive to remain there; the robot then moves freely across the grid so it explores other parts as well. Since there is no decay, it does not need to revisit previously explored areas.

**Environment 5:** This environment has a static field with no information decay, as the process noise is zero. As shown in Fig. 5, the top left quadrant of the grid has a higher target clarity, while its zero elsewhere. The robot first moves to the top left and increases clarity until the clarity deficit approaches zero; the objective provides no further incentive to remain there; the robot then moves freely across the grid so it explores other parts as well. Since there is no decay, it does not need to revisit previously explored areas.

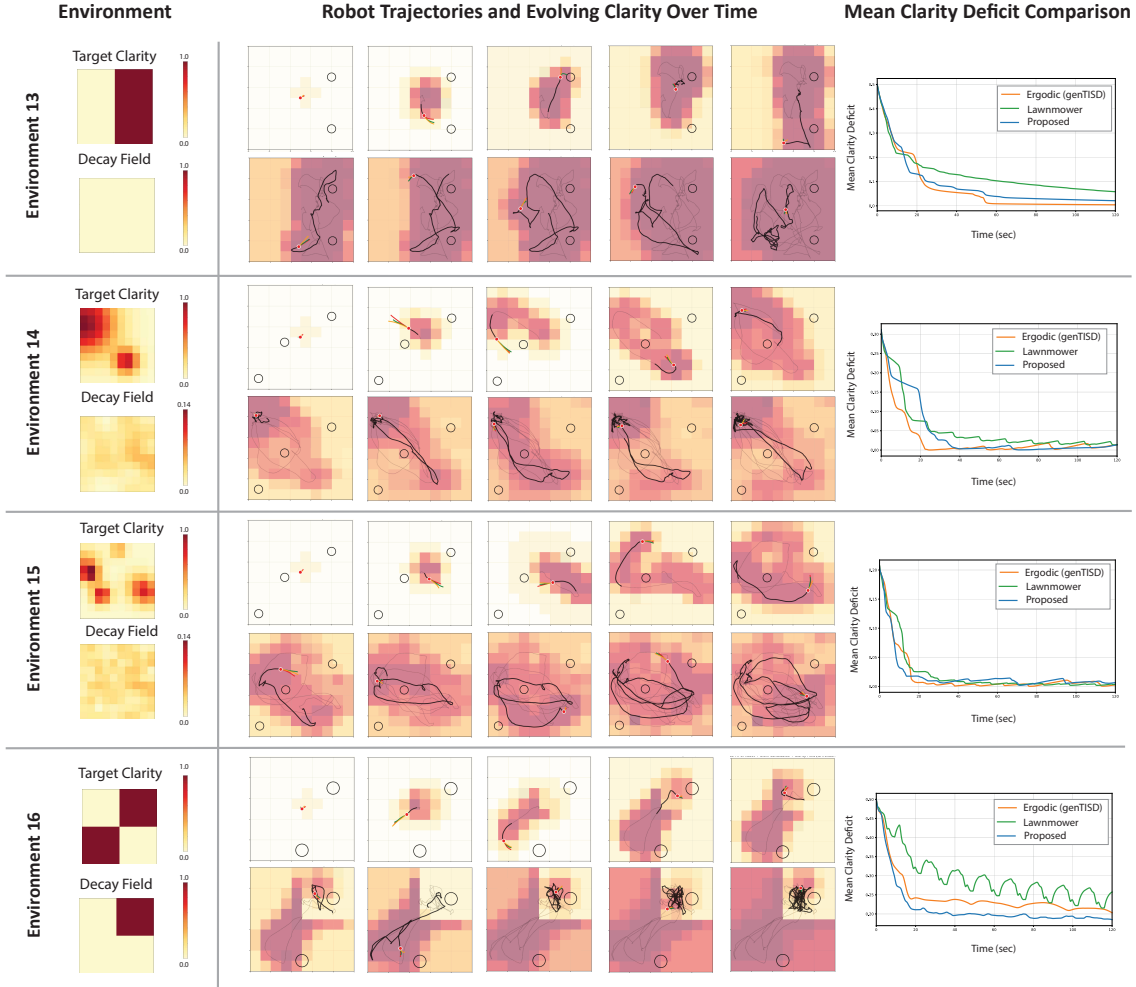


Figure 8

**Environment 6:** This environment as shown in Fig. 6 has a decay field that has a high decay in the lower half and no decay in the top half. and has uniform high target clarity across the grid. The robot first moves around in the upper half and the clarity deficit approaches zero; the objective provides no further incentive to remain there, and additionally there is no decay here so it will stay that way; the robot then moves and stays in the bottom half, which has high decay.

**Environment 7:** This environment as shown in Fig. 6 has a decay field that has a high decay in the upper half and no decay in the lower half. and has uniform high target clarity across the grid. The robot first moves around in the lower half and the clarity deficit approaches zero; the objective provides no further incentive to remain there, and additionally there is no decay here so it will stay that way; the robot then moves and stays in the upper half, which has high decay.

**Environment 8:** This environment contains decaying and non decaying regions, as shown in Fig. 6. The top right and bottom left quadrants have a target clarity of 1, while elsewhere it is 0. Decay is present only in the top right region. The robot first moves to the bottom-left quadrant, minimizing the target clarity deficit. It then moves to the top-right, where decay causes clarity to

decrease over time. The robot stays in this region to minimize the clarity deficit, illustrating how the planner adapts to temporal decay. Also the robot does not spend much time in other two quadrants.

**Environment 9:** This environment contains variable decaying and non decaying regions, as shown in Fig. 6. The top half and bottom left quadrant have a target clarity of 1, while elsewhere it is 0. Decay is present in higher is top right and then lower in top left and then 0 elsewhere. The robot first moves to the bottom-left quadrant, minimizing the target clarity deficit. It then moves to the top-half, where decay causes clarity to decrease over time. The robot moves between quadrants in this region to minimize the clarity deficit. Also the robot does not spend much time in bottom right quadrant.

**Environment 10:** This environment has variable decay across the environment, as shown in Fig. 7. The target clarity is also complex and varies through the environment and is higher in some parts. The robot first moves between the regions of high target clarity, minimizing the target clarity deficit. It then moves to the areas, where decay causes clarity to decrease over time. Also the robot does not spend much time in low target clarity regions.

**Environment 11:** This environment has variable decay across the environment, as shown in Fig. 7. The target clarity is also complex patchy pattern that varies through the environment and is higher in some parts. The robot first moves among regions of high target clarity, minimizing the target clarity deficit. It then moves to the areas, where decay causes clarity to decrease over time. The robot does not spend much time in low target clarity regions.

**Environment 12:** This environment has variable decay across the environment, as shown in Fig. 7. The target clarity is also complex patchy pattern that varies through the environment and is higher in some parts. The robot first moves among regions of high target clarity, minimizing the target clarity deficit. It then moves to the areas, where decay causes clarity to decrease over time. The robot does not spend much time in low target clarity regions.

**Environments 13, 14, 15, 16:** These environments include obstacles near regions of high target clarity in some of the previous environments, as shown in Fig. 8. They evaluate the planner's ability to safely explore informative areas separated by obstacles. The robot successfully navigates around obstacles without collisions or deadlocks, adjusting its motion based on the clarity deficit and visiting high-clarity regions as needed. The integrated gating mechanism ensures that all trajectories remain within the safe set, maintaining safety while the planner adapts its motion to minimize the clarity deficit in the presence of obstacles.