# Depth-Constrained ASV Navigation With Deep RL and Limited Sensing

Amirhossein Zhalehmehrabi, Daniele Meli, Francesco Dal Santo, Francesco Trotti, *Graduate Student Member, IEEE*, and Alessandro Farinelli

***Abstract*—Autonomous Surface Vehicles (ASVs) play a crucial role in maritime operations, yet their navigation in shallow-water environments remains challenging due to dynamic disturbances and depth constraints. Traditional navigation strategies struggle with limited sensor information, making safe and efficient navigation difficult. In this letter, we propose a reinforcement learning (RL) framework for ASV navigation under depth constraints, where the vehicle must reach a target while avoiding unsafe areas with only a single depth measurement per timestep from a downward-facing Single Beam Echosounder (SBES). To enhance environmental awareness, we integrate Gaussian Process (GP) regression into the RL framework, enabling the agent to progressively estimate a bathymetric depth map from sparse sonar readings. This approach improves decision-making by providing a richer representation of the environment. Furthermore, we demonstrate effective sim-to-real transfer, ensuring that policies generalize well to real-world aquatic conditions. Experimental results validate our method's capability to improve ASV navigation performance while maintaining safety in challenging shallow-water environments.**

***Index Terms*—Autonomous vehicle navigation, learning and adaptive systems, reinforcement learning, Gaussian process regression, sim-to-real transfer.**

## I. Introduction

AUTONOMOUS Surface Vehicles (ASVs) are unmanned vessels increasingly used in maritime operations such as environmental monitoring, search-and-rescue, and surveillance. However, they face major challenges in dynamic aquatic environments, where currents and waves complicate precise navigation and the development of robust control strategies [1], [2]. These challenges are amplified in shallow waters, where strict depth constraints apply. Recent advances in bathymetric surveying highlight the need for millimeter-level positioning accuracy, especially in coastal regions under 2.0 m deep, where rock formations and artificial structures create highly complex navigation conditions [3].

Received 12 June 2025; accepted 5 October 2025. Date of publication 24 October 2025; date of current version 7 November 2025. This article was recommended for publication by Associate Editor Alberto Maria Metelli and Editor Jens Kober upon evaluation of the reviewers' comments. This work was supported by PNRR iNEST Project under Grant ECS00000043. *(Corresponding author: Amirhossein Zhalehmehrabi.)*

The authors are with the Department of Computer Science, University of Verona, 37134 Verona, Italy (e-mail: amirhossein.zhalehmehrabi@univr.it).

The code is available at https://github.com/Isla-lab/depth-constrained-aquatic-navigation.

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3625520, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3625520

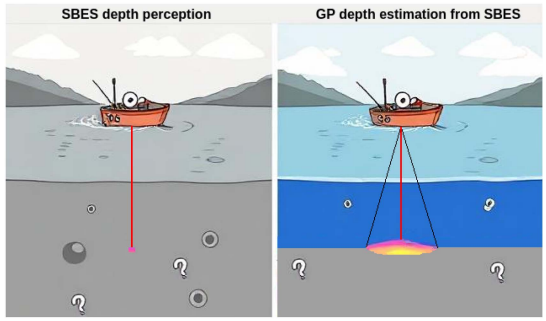


Fig. 1. Illustration of the improved observation space achieved through GP.

Deep RL has recently shown promise in robotics for tackling such complex problems [4], [5]. Model-free policy search develops parametric policies through direct interactions with the environment, making it suitable for tasks that are difficult to model. This approach allows optimizing high-capacity neural networks capable of handling diverse state representations, providing flexibility in algorithm design.

We address the problem of ASV navigation under depth constraints using RL, where the vessel must reach a target while avoiding both shallow and deep *unsafe areas*. We focus on the practical case of limited sensing: a single downward-facing Single Beam Echosounder (SBES) providing only one depth measurement at the current position. This reflects the growing trend toward low-cost, lightweight, and easily deployable ASVs for environmental monitoring, where minimal sensor setups reduce complexity and cost [6].

This requires the agent to infer unsafe areas without explicit knowledge of their positions, making the environment partially observable. In RL, partial observability is typically addressed through belief-state methods [7], history-based methods [8], model-based approaches [9], or direct observation [10]. We frame the problem as belief-state estimation and employ Gaussian Process (GP) regression to incrementally build a spatial belief of the bathymetric map from sparse SBES readings (Fig. 1). While GPs provide principled uncertainty estimates, predictive variance often saturates in regions with little data due to kernel and approximation limitations [11], leading to overconfident predictions and unreliable uncertainty quantification.

To address this, we extend local and sparse GP approaches [12], [13] with a confidence mechanism that mitigates variance shrinkage in unobserved regions. The resulting belief map, updated in real time, is embedded in the agent's observation space, improving policy learning and transfer.

We validate our method through sim-to-real deployment on a custom-built ASV equipped only with an SBES, requiring no additional fine-tuning in real-world tests (Fig. 7).

The main contributions of this paper are:

- A spatial belief model based on localized GP regression for uncertainty-aware depth estimation from sparse SBES readings.
- A novel confidence-based mechanism to mitigate variance saturation in GP models.
- The first integration of real-time belief updates into an RL policy for depth-constrained ASV navigation.
- Effective zero-shot sim-to-real transfer, with a trained policy deployed on a real ASV without fine-tuning.

## II. Related Works

Current research in the domain of aquatic navigation, particularly surface navigation using ASVs, generally falls into three primary categories: Model-based methods, RL-based methods and depth-constrained navigation approaches.

### A. Model-Based Aquatic Navigation

Model-based navigation approaches, such as Model Predictive Control (MPC), leverage a system dynamics model to predict future states and optimize control actions accordingly. These methods have been widely used in ASV navigation due to their ability to handle constraints and provide stable, interpretable control; however, they often struggle with modeling inaccuracies and computational complexity in highly dynamic environments [14], [15]. Unlike model-based approaches, RL does not rely on an explicit dynamics model, allowing it to handle unmodeled effects and adapt to complex environments. Following recent trends in complex domains such as UAV control [16], we focus on RL, which has been shown to outperform optimal control not by optimizing the same objective more effectively, but by optimizing a more suitable objective that accounts for real-world complexities, leading to more robust control strategies.

### B. Reinforcement Learning for Aquatic Navigation

A fundamental challenge in navigation is obstacle avoidance. In our case, the objective is to avoid unsafe areas, such as shallow regions, which act as obstacles. However, unlike traditional settings, we do not have access to direct distance measurements to these hazards—for example, no side-scan or multi-beam sonar—making the task more challenging.

Several studies have applied RL to collision avoidance and path planning in aquatic environments. Zhou et al. [17] developed RL algorithms for both single and multi-ASV path planning with a focus on obstacle avoidance, though their approach assumed perfect knowledge of obstacle locations within a fully observable simulated environment. Meyer et al. [18] used rangefinder-based perception systems to train RL agents to avoid static obstacles. Other studies have also addressed obstacle avoidance [19], [20], [21], [22], but all incorporate explicit obstacle information through sensors such as rangefinders or positional data.

The key distinction of our work is that it relies solely on a single SBES, provides depth readings only at the ASV's current location. We construct a probabilistic model to infer the surrounding areas, whereas existing RL methods for aquatic navigation use sensing mechanisms that directly detect obstacles.

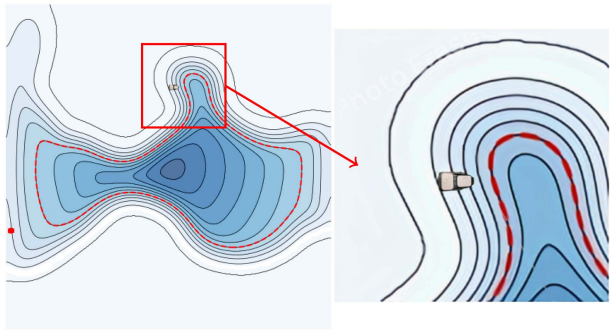

Fig. 2. Generated depth maps showing target point (red dot), unsafe area boundary (red dashed line at $L_d$ level), and shoreline (outermost contour). Inset shows ASV navigating near shallow waters.

### C. Depth-Constrained Navigation

Research focusing specifically on depth-constrained navigation has primarily utilized waypoint-based strategies [3] and adaptive control mechanisms [23] to ensure safe operation in varying underwater topographies. These methods rely on predefined paths or reactive control adjustments based on bathymetric data. Our research bridges these domains by employing RL techniques to solve the depth-constrained navigation problem using insufficient sensory input. By training an RL agent to effectively interpret limited depth information, we enable autonomous navigation in shallow-water environments without requiring comprehensive bathymetric maps or multiple sensing modalities.

## III. Method

We now introduce the depth-constrained ASV navigation problem and outline our RL- and GP-based solution.

### A. Task Definition

The depth-constrained ASV navigation task (Fig. 2) is formulated as an optimization problem, the objective is to minimize the distance to a designated target while avoiding unsafe areas, represented by the shoreline and regions where the water depth exceeds the maximum allowable depth $L_d$. In practice, $L_d$ is application- and hardware-specific, determined by the maximum safe depth at which reliable measurements can be obtained.

The ASV relies only on a downward-facing SBES, which is used to detect and avoid unsafe areas based on depth measurements.

Our goal is to develop a generalizable strategy for solving this task in unknown water basins, starting from a RL policy learned in arbitrary simulation environments. To this aim, we use online GP regression to estimate the depth map of the basin, as the ASV navigates and collects readings from the SBES. We then define the navigation problem as a POMDP, with an additional observation given by the GP estimation.

### B. ASV Dynamics

Following [24], the ASV is modeled as a six-degree-of-freedom (6 DOF) rigid body system with mass $m$. The nonlinear

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

dynamics are governed by the following equations:

$$\begin{aligned} m(\dot{u} + qw - rv) &= X_{\text{tot}}, \quad & I_x\dot{p} + (I_z - I_y)qr &= K_{\text{tot}} \\ m(\dot{v} + ru - pw) &= Y_{\text{tot}}, \quad & I_y\dot{q} + (I_x - I_z)rp &= M_{\text{tot}} \\ m(\dot{w} + pv - qu) &= Z_{\text{tot}}, \quad & I_z\dot{r} + (I_y - I_x)pq &= N_{\text{tot}} \end{aligned} \tag{1}$$

where $X_{\text{tot}}$, $Y_{\text{tot}}$, and $Z_{\text{tot}}$ represent the resultant forces in surge ($x$), sway ($y$), and heave ($z$) directions, while $K_{\text{tot}}$, $M_{\text{tot}}$, and $N_{\text{tot}}$ denote the moments about the roll ($x$), pitch ($y$), and yaw ($z$) axes. $I_x$, $I_y$, and $I_z$ are the principal moments of inertia, with $u$, $v$, $w$ and $p$, $q$, $r$ representing the linear and angular velocities in the body-fixed frame.

### C. GP Depth Estimation

Incorporating a single depth measurement per timestep provides insufficient information about the underlying depth map, making navigation challenging, as evidenced by the poor performance of our baseline model w/o GP in empirical evaluation. To address this limitation, we employ localized GP regression to estimate the depth of nearby locations, leveraging past SBES readings to construct a probabilistic representation of underwater terrain where the posterior update at location $\mathbf{x}_t$ considers only points in the local neighborhood $\mathcal{M}(\mathbf{x}_t)$. This localization reduces computational complexity and emphasizes nearby depth information, which is most relevant for real-time ASV navigation.

This approach enables the vehicle to infer depth variations beyond its measurement, improving decision-making in sparse sensing conditions.

Let $d : \mathbb{R}^2 \to \mathbb{R}$ denote the continuous depth function of the underwater terrain. At time $t$, the ASV records a sonar measurement $z_t$ corresponding to the depth $d(\mathbf{x}_t)$ at its current location $\mathbf{x}_t \in \mathbb{R}^2$, given by

$$z_t = d(\mathbf{x}_t) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2_{\text{SBES}}). \tag{2}$$

where $\epsilon$ is the measurement noise and $\sigma^2_{\text{SBES}}$ is the sensor's variance. To maintain computational efficiency and integrate with the RL pipeline, we represent the depth belief on a 2D grid where each cell covers $\xi \times \xi\,\text{cm}^2$. This discretisation applies only to the depth map representation[1] That is, the continuous domain is discretized into a regular grid, and GP estimates are maintained at the center of each grid cell. Let $\mathcal{M}(\mathbf{x_t}) = \{\mathbf{x}' \mid k(\mathbf{x_t}, \mathbf{x}') > \delta\}$[2] denote a local neighborhood of $\mathbf{x_t}$, where $k$ is the covariance function. Within this region, we assume a GP prior:

$$d(\mathbf{x_t}) \sim \mathcal{GP}(\mu_t(\mathbf{x_t}), k(\mathbf{x_t}, \mathbf{x}')), \tag{3}$$

where $\mu_t(\mathbf{x})$ is the prior mean function. For each location $\mathbf{x}^i_t \in \mathcal{M}(\mathbf{x}_t)$, where the subscript $t$ denotes the current timestep and the superscript $i$ is the index of the $i$-th neighbor location within the local set $\mathcal{M}(\mathbf{x}_t)$, the posterior parameters of the Gaussian process are updated by incorporating the new data in a Bayesian manner.

$$\sigma^2_{t+1}(\mathbf{x}^i_t) = \frac{\sigma^2_t(\mathbf{x}^i_t) \cdot \sigma^2_{\text{w}}}{\sigma^2_t(\mathbf{x}^i_t) + \sigma^2_{\text{w}}}$$

$$\mu_{t+1}(\mathbf{x}^i_t) = \frac{\sigma^2_{\text{w}} \cdot \mu_t(\mathbf{x}^i_t) + \sigma^2_t(\mathbf{x}^i_t) \cdot z_t}{\sigma^2_t(\mathbf{x}^i_t) + \sigma^2_{\text{w}}} \tag{4}$$

where

$$\sigma^2_{\text{w}} = \frac{\sigma^2_{\text{SBES}}}{k(\mathbf{x}^i_t, \mathbf{x}_t)^2}$$

represents the sensor variance weighted by the covariance function $k$. The update mechanism derives from the Gaussian conjugate prior framework, ensuring that new data incorporation leads to tractable sequential updates [25]. The resulting posterior distribution reflects both prior knowledge and the impact of the new observation on the regions near the current location.

To mitigate variance saturation in regions with sparse data, we define a confidence proxy $C$ that leverages the GP covariance structure to better distinguish between observed and inferred locations. This proxy provides a more reliable uncertainty signal for downstream decision-making.

For each estimated location $\mathbf{x}^i_t$, the confidence proxy is defined as:

$$C(\mathbf{x}^i_t) = \frac{\sum_{\mathbf{x}^j \in \mathcal{O}} k(\mathbf{x}^i_t, \mathbf{x}^j)}{\sum_{\mathbf{x}^k \in \mathbb{R}^2} k(\mathbf{x}^i_t, \mathbf{x}^k)} = \frac{\sum_{\mathbf{x}^j \in \mathcal{O}} k(\mathbf{x}^i_t, \mathbf{x}^j)}{\sum_{\mathbf{x}^k \in \mathcal{M}} k(\mathbf{x}^i_t, \mathbf{x}^k)} \tag{5}$$

where $\mathcal{O}$ represents the set of previously observed locations. The denominator ensures normalization by considering the total covariance sum over all spatial locations, restricting $C(\mathbf{x}^i_t)$ to the range $[0, 1]$. Furthermore, for every observed point $\mathbf{x}_t$, the confidence proxy is set to $C(\mathbf{x}_t) = 1$, ensuring maximum confidence.

After applying the update, the measurement $(\mathbf{x}_t, z_t)$ is discarded and does not affect future updates. This localized GP update enables the ASV to refine depth in real-time while ensuring computational efficiency.

### D. Gradient-Based Extrapolation

To enhance coverage along the boat's trajectory $\{\mathbf{x}_t\}_{t=0}^T$, we introduce a lightweight extrapolation strategy. Assuming local smoothness of the bathymetric function $d$, we estimate the depth at a forward position $\tilde{\mathbf{x}}_t = \mathbf{x}_t + \Delta s\,\mathbf{u}_t$, where $\Delta s$ denotes a fixed step size and $\mathbf{u}_t$ is the unit vector along the current heading direction of the ASV, using a finite-difference approximation of the directional derivative:

$$\tilde{z}_t = z_t + \Delta s\,\frac{z_{t+1} - z_t}{\|\mathbf{x}_{t+1} - \mathbf{x}_t\|}. \tag{6}$$

We treat $(\tilde{\mathbf{x}}_t, \tilde{z}_t)$ as a pseudo-observation and incorporate it into the GP model with reduced confidence, assigning $C(\tilde{\mathbf{x}}_t) = \alpha$ for a tunable $\alpha < 1$..[3] This improves predictive density in unexplored regions, as validated by ablation studies.

### E. POMDP Formulation

A Partially Observable Markov Decision Process (POMDP) [26] is a tuple $(S, A, O, P, Z, R, \gamma)$, where $S$ is a set of partially observable *states*; $A$ is a set of *actions*; $Z$ is a set of *observations*; $P$: $S \times A \to \Pi(S)$ is the *state-transition model*, mapping to a probability distribution $\Pi(\cdot)$ over states;

[1] Observations and actions remain continuous.

[2] In our experiments, we set $\delta = 0.01$ based on empirical observations.

[3] Empirically, we set $\alpha = 0.9$.

$O$: $S \times A \to \Pi(Z)$ is the *observation model*; $R$ is the *reward function* and $\gamma \in [0, 1)$ is a *discount factor*. The probability distribution over states $\mathcal{B} = \Pi(S)$, called *belief*, is used to model uncertainty about the true state. The goal of solving a POMDP is to compute a policy $\pi : \mathcal{B} \to A$, to maximize the expected return:

$$\mathbb{E}_{a_t \sim \pi} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right]. \tag{7}$$

We now specialize the POMDP formulation to the depth-constrained navigation problem as follows:

*1) Transition Model:* For practical control applications, a reduced-order 3 DOF model of the ASV dynamics (explained in III-B) is typically adopted by constraining pitch ($q$), roll ($p$), and heave ($w$) motions through the assumptions $p = q = w = 0$. This yields the simplified horizontal-plane dynamics:

$$m(\dot{u} - rv) = X_{\text{tot}}, \quad m(\dot{v} + ru) = Y_{\text{tot}}, \quad I_z \dot{r} = N_{\text{tot}} \tag{8}$$

This simplified formulation retains only surge, sway, and yaw dynamics while eliminating cross-coupling effects from vertical and rotational modes, maintaining sufficient fidelity for surface navigation control objectives [24].

*2) Action Space:* At each time step $t$, the policy outputs a two-dimensional action vector $a_t = [u_t, \omega_t]$ where $u_t$ and $\omega_t$ are the linear and angular velocities in the body frame of the ASV, respectively. The action space is continuous and bounded within the kinematic limits of the ASV.

*3) Observation Space:* At each time step $t$, the agent receives a temporal sequence of observations $\{o_i\}_{t-T}^{t}$, including key environmental and agent-related variables. Specifically, each observation $o_i$ includes the relative target's position in polar coordinates as $(r_i, \theta_i)$, where $r_i$ denotes the radial distance and $\theta_i$ denotes the angular distance in boat frame; the linear acceleration $\dot{u}_i$ and angular acceleration $\dot{\omega}_i$; the previous action $a_{i-1}$; the current depth reading $z_i$; the estimated depth gradients and the confidence proxy in four primary directions: forward, backward, right and left. Let $\mathbf{x}_i^m$ be the adjacent position in the main directions to the $\mathbf{x}_i$, where $m \in \{f, b, l, r\}$. For each direction, the gradient is computed as:

$$g_i^m = \frac{\mu_{t+1}(\mathbf{x}_i^m) - z_i}{\mathbf{x}_i^m - \mathbf{x}_i} \tag{9}$$

where $z_i$ is the current depth reading and $\mu_{t+1}(\mathbf{x}_i^m)$ is the predictive mean function evaluated at the corresponding position. Additionally, the confidence proxy values $C(\mathbf{x}_i^m)$ where $m \in \{f, b, l, r\}$ represent the uncertainty associated with each gradient estimate. This structured representation enables the policy to interpret depth variations and environmental changes based on partial observations, aiding decision-making.

*4) Reward Function:* The Bathymetry-Aware ASV navigation task aims to minimize the distance to the goal while ensuring that the ASV remains clear of unsafe areas. To achieve this objective, we define the following reward function:

$$R(s_t, a_t, s_{t+1}) = \begin{cases} r_{suc} & \text{if } success; \\ r_{fail} & \text{if } fail; \\ -\alpha_1 \Delta_{\text{d}} - \alpha_2 r_{\text{back}} - \alpha_3 r_{\text{depth}} & \text{otherwise}, \end{cases} \tag{10}$$

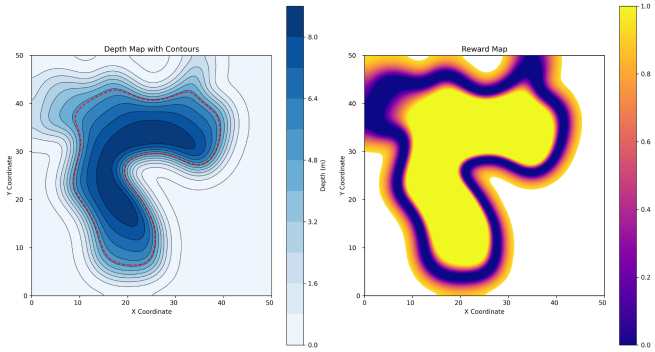


Fig. 3. Illustration of $r_{\text{depth}}$.

[4]where $\Delta_d$ denotes the change in the geodesic distance to the goal from the previous state, $r_{back} = |\min(u_t, 0)|$ quantifies the extent of backward movement. We penalize backward motion because, in reality, the ASV's reverse movement is limited by significantly reduced control and efficiency. Additionally, $r_{depth}$ at time $t$ is defined by

$$r_{\text{depth}} = \begin{cases} 0 & \text{if } \frac{1}{3} L_{\text{d}} \le z_t < \frac{2}{3} L_{\text{d}} \\ \frac{3}{L_d} \left( z_t - \frac{2}{3} L_{\text{d}} \right) & \text{if } \frac{2}{3} L_{\text{d}} \le z_t < L_{\text{d}} \\ \frac{3}{L_d} \left( \frac{1}{3} L_{\text{d}} - z_t \right) & \text{if } 0 \le z_t < \frac{1}{3} L_{\text{d}} \end{cases} \tag{11}$$

where $z_t$ is the depth at time $t$, and $L_d$ defines unsafe regions. No penalty is applied within the safe range $[\frac{1}{3} L_d, \frac{2}{3} L_d)$; outside this, the penalty increases linearly. This encourages the agent to maintain a safe depth, avoiding shallow and deep extremes (see Fig. 3). Additionally, discrete rewards are provided for specific events. A large negative reward $r_{fail} = -100$ is given upon entering to unsafe areas, or timeout event, terminating the episode. A large positive reward $r_{suc} = 200$ is given when the target is reached.

## IV. Experiments

We design our experiments to answer the following research questions:

RQ1: How does the success rate of our method compare to that of a privileged model with access to true depth gradients?
RQ2: What is the impact of GP regression with respect to the single SBES information from the depth sensor?
RQ3: How does our model compare to a safe RL approach [27], where the goal is to find an optimal policy under the constraint to keep $r_{\text{depth}} = 0$ (i.e., the ASV shall remain in the safe zone)?
RQ4: How does our online GP-based estimation of the local depth map compare to a pre-trained LSTM network?
RQ5: How well does our methodology generalize out of the training dataset, both in simulation and in a real depth-constrained navigation task?

We begin by describing the implementation details and the training setup. Next, we assess the performance of our approach in simulation, followed by an evaluation of the learned policy in real-world experiments.

[4]For this work, we set $\alpha_1 = 5$, $\alpha_2 = 1$, and $\alpha_3 = 0.05$.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

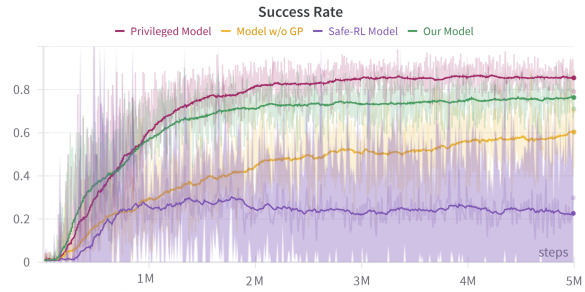


Fig. 4. Mean success rate over 5 million training steps, averaged across 5 random seeds for each approach.

### A. Implementation and Training Details

We generate 120 depth maps in simulation for training (Fig. 2), randomizing the shoreline and depth values with radial gradients, noise and filtering techniques. The maps are sized $50 \times 50$ m with a depth resolution of 10 cm$^2$. The dynamic model of the ASV is used to implement a vectorized OpenAI gym environment, which allows for simulating multiple ASV's in parallel, achieving approximately 2000 FPS on a single GeForce 4070 GPU. The RL environment is implemented using the Stable-Baselines3 [28], and the safe environment is constructed with the OmniSafe [29]. For our proposed method, we employ a two-layer multilayer perceptron (MLP) with 256 units per layer. We train our agent using Soft Actor-Critic (SAC) [30], which is a popular off-policy reinforcement learning algorithm that maximizes reward while optimizing for entropy, encouraging exploration and robust policies. For safe-RL, we used the Lagrangian version of SAC (SACLag) [31]. For the kernel function, we used the well-known Radial Basis Function (RBF) kernel with a length-scale set to 3.0 empirically. We trained all the models for 5 million steps and for 5 random seeds.

The method does not impose strict requirements on measurement frequency. In simulation and real-world experiments, the control loop ran at $\Delta t = 0.01$ s (100 Hz), while the SBES provided one depth reading at 10 Hz. Model inference averaged 0.002128 s with std 0.005840 s step, well below the control period, ensuring real-time operation. We evaluated approaches using five metrics: Success Rate (SR) measuring task completion percentage, Efficiency Score (ES) calculated as success rate divided by episodic length, Depth Break (DB) indicating episodes terminated by depth constraint violations, Velocity Smoothness (VS) quantifying velocity change smoothness, and Heading Smoothness (HS) measuring heading direction change rates [32].

### B. Simulation Results

We first evaluate our proposed approach in simulation, against several baselines to comprehensively assess the performance:

1) **Privileged model:** This model has privileged access to true depth gradients, bypassing estimation and serving as a performance upper bound.
2) **Model w/o GP:** In this variant, the GP information is omitted from the state space.
3) **Safe RL:** In this model, the cost is computed from $r_{\text{depth}}$ in (11), omitting it from the reward function in (10).

Fig. 4 shows the success rate (i.e., percentage of episodes where the policy reaches the target while keeping the depth constraint) along the learning curve of the different algorithms on the training maps. Our model (green) achieves 76.33% SR on average, with limited variance (hence being more robust). This is slightly worse than the performance achieved by the privileged model (red), which attains 85.45% success rate on average, showing that our GP-enhanced RL architecture well approximates full environmental awareness, requiring less sensory information (RQ1). In contrast, the model without GP information (yellow) achieves 60.34% success rate with larger variance, highlighting the key contribution of GP regression (RQ2). The safe RL approach (purple) performs worst (22.6% average), possibly because SACLag's cost function constraints restrict exploration and learning efficiency, leading to suboptimal policy updates. It also exhibits high variance, suggesting learning instability (RQ3).

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON THE NAVIGATION TASK ON 100 UNSEEN DEPTH MAPS FOR 1000 EPISODES

| Method | SR (%) | ES | MDB (%) | VS | HS |
|---|---|---|---|---|---|
| Privileged | 85.78 (1.54) | 0.781 (0.15) | 5.66 (1.16) | 69.71 (6.78) | 24.70 (15.06) |
| w/o GP | 54.52 (2.42) | 0.222 (0.14) | **10.98** (2.12) | **36.58** (7.20) | **8.73** (4.72) |
| Safe-RL | 23.98 (20.73) | 0.325 (1.15) | 58 (16.83) | 88.96 (8.95) | 68.98 (12.50) |
| LSTM-based | 56.08 (2.21) | 0.220 (0.04) | 16.62 (4.25) | 44.68 (4.12) | 25.44 (10.19) |
| Our Model | **76.26** (2.44) | **0.745** (0.28) | 12.16 (1.64) | 56.64 (11.02) | 22.21 (11.52) |

To evaluate RQ4, we replaced the GP module with an LSTM network trained on 65 k samples from pre-trained models, keeping all other components unchanged. An ablation study showed that doubling the training data improved validation loss by $< 0.5\%$, confirming 65k samples were sufficient. During RL fine-tuning, LSTM weights remained frozen. As shown in Fig. 6, this neural surrogate suffered from distribution shift, with success rates dropping from 76.33% (GP model) to 52.78% (LSTM), a 23.55 percentage point decline that demonstrates the robustness advantages of our GP-based approach.

To evaluate generalization (RQ5), we tested the best policies on 100 unseen depth maps, running 1,000 episodes per model (see Table I). Results align with findings from RQ1–4. Our model performs closest to the privileged one (76.26% vs. 86.78% success), significantly outperforming the w/o GP baseline (54.52%) and LSTM (56.08%), highlighting the importance of online GP regression over static neural models. We also assessed the *efficiency score*, where our method again ranks highest (0.745), close to the privileged upper bound (0.781). For *mean depth break* (percentage of episodes violating depth constraints), our approach ranks second (12.16%), slightly behind the no-GP model (10.98%) but with significantly higher success and efficiency.

In our analysis, an interesting pattern emerged when comparing model performance. Despite our model achieving a success rate approximately 10% lower than the privileged model, the efficiency scores between the two were similar. This efficiency score can be attributed to our model's shorter average episode length. The privileged model exhibited a distinctive movement pattern characterized by rapid acceleration followed by abrupt deceleration when detecting depth gradients indicative of unsafe areas. This resulted in a sinusoidal trajectory along safe regions. Conversely, our model operated with inherent uncertainty regarding depth readings and estimations, leading to more conservative movement strategies. It is important to note that this smoother trajectory is not an inherent advantage of our

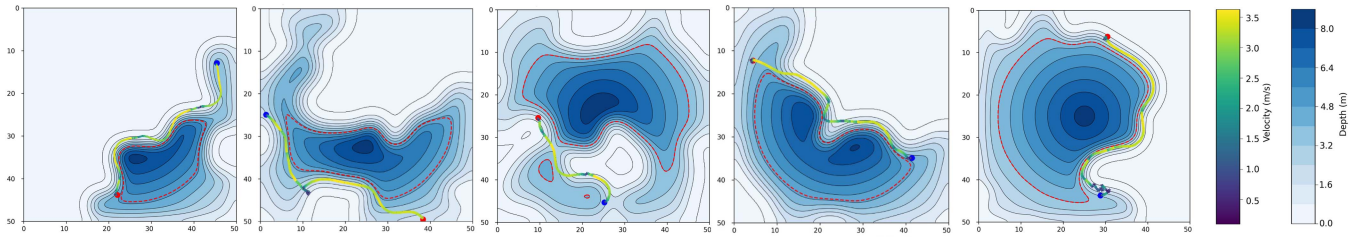


Fig. 5. ASV navigation trajectories showing multiple path strategies from starting points (blue) to targets (red), with color gradients representing velocity variations across different scenarios.

TABLE II
ABLATION STUDY RESULTS SHOWING THE IMPACT OF DIFFERENT COMPONENTS. THE LAST LINE REPRESENTS OUR FULL MODEL. ROW 2 USES TECHNIQUES PROPOSED BY [12], WHILE ROW 4 COMBINES TECHNIQUES FROM BOTH [12] AND [13]

| | GP | Gradient-Based Extrapolation | Variance | Proxy | SR (%) | ES | MDB (%) | VS | HS |
|---|---|---|---|---|---|---|---|---|---|
| 1 | ✗ | ✗ | ✗ | ✗ | 54.52 (2.42) | 0.222 (0.14) | **10.98** (2.12) | **36.58** (7.20) | **8.73** (4.72) |
| 2 | ✓ | ✗ | ✗ | ✗ | 62.4 (1.97) | 0.289 (0.03) | 24.07 (6.18) | 51.66 (12.99) | 28.10 (12.70) |
| 3 | ✓ | ✓ | ✗ | ✗ | 72.76 (4.63) | 0.488 (0.10) | 18.78 (4.14) | 51.78 (15.41) | 21.51 (6.55) |
| 4 | ✓ | ✓ | ✓ | ✗ | 69.9 (2.52) | 0.693 (0.30) | 16.16 (1.33) | 60.99 (7.41) | 23.14 (22.49) |
| 5 | ✓ | ✓ | ✗ | ✓ | **76.26** (2.44) | **0.745** (0.28) | 12.16 (1.64) | 56.64 (11.02) | 22.21 (11.52) |

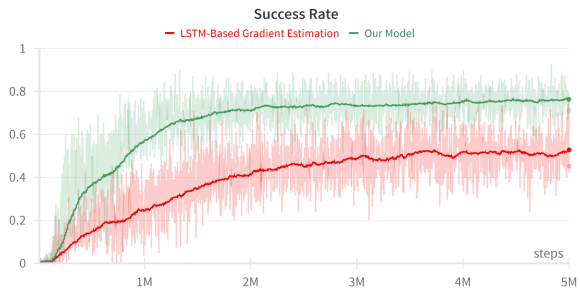


Fig. 6. Mean success rate over 5 million training steps, averaged across 5 random seeds for each approach.

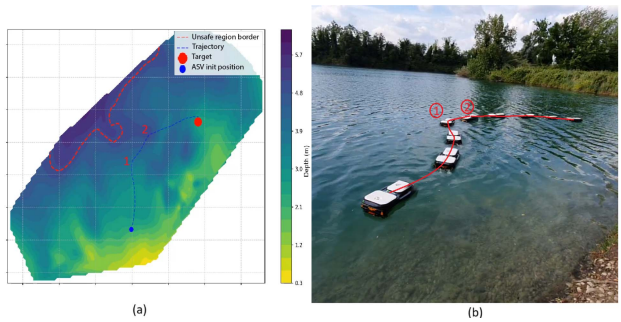


Fig. 7. (a) Depth map of the scenario with the ASV's trajectory. (b) Composite image from the real-world experiment showing the ASV navigating the environment.

approach, but rather a side effect of the uncertainty in our model's perception system. With appropriate reward functions designed to optimize trajectory smoothness, the privileged model could potentially achieve both higher success rates and smoother navigation. However, since neither model was explicitly optimized for smoothness in our experimental setup, our model's cautious approach manifested as slower but more consistent progression with fewer directional fluctuations compared to the privileged model. This observation is further validated by our study over VS and HS in Table I, where the privileged model used more linear and angular velocity changes.

### C. Ablation Study

We conducted an ablation study over 1,000 episodes in Table II, to highlight the independent contribution of 3 key items in our methodology: GP estimation, Gradient-Based Extrapolation (explained in III-D) and proxy ( (5)). Removing the GP module significantly reduced performance, confirming its role in depth-aware navigation. Enabling GP (row 2) [12] improved the baseline, while combining it with Gradient-Based Extrapolation [13] (row 3) further boosted the Success Rate and Efficiency Score. Adding the Variance component (row 4) led to a 2.52% drop in Success Rate and a 1.8% increase in depth violations, likely due to overconfident estimations. Replacing Variance with the Proxy module (row 5) achieved the highest Success Rate (74.14%) and Efficiency Score (0.698) with moderate depth constraints. These results affirm that GP, Gradient-Based Extrapolation, and Proxy are critical for optimal performance.

### D. Real-World ASV Navigation

To evaluate the transferability of our policy from simulation to the real world, we deployed it on a physical custom-built ASV (Fig. 7) with two underwater thrusters (Blue Robotics T200, providing up to 6.7 kgf of forward thrust at 20 V), one GPS, a centrally mounted downward-facing SBES sensor with the maximum range of 100 m and an accuracy of $\pm 0.1$ m, and a Raspberry Pi 3 that serves as the onboard computer for running the policy and managing sensor data. A schematic representation of the ASV platform, including thruster placement and sensor configuration, is provided in Fig. 8. The ASV was tasked with reaching randomly generated target points in a real aquatic environment. Fig. 7 illustrates the real-world scenario used in our experiments. A video of these real-world experiments is provided in the supplementary materials. We conducted five experiments, with the ASV successfully reaching the target in four out of five runs. The only failure was due to a depth constraint violation, caused by fluctuations in the depth sensor's sampling frequency. The efficiency score (ES) was 0.530. While

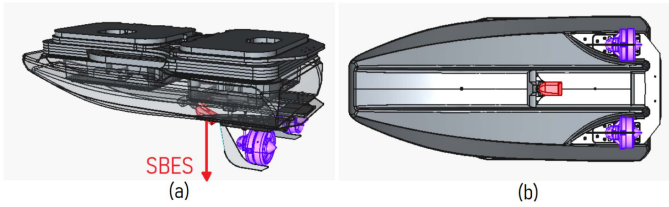


Fig. 8. Schematic views of the ASV highlighting the downward-facing SBES sensor (red) and twin thrusters (purple): (a) 3D perspective view and (b) bottom view.

the policy showed generally reliable performance, we observed fluctuations in control frequency stemming from the Raspberry Pi's limited processing power, occasionally leading to degraded control. Communication delays, which were not modeled during training, also contributed to the sim-to-real performance gap. Nonetheless, the method achieved a good success rate in real-world scenarios with limited sensing.

## V. Conclusion

We presented a reinforcement learning approach for ASV navigation under depth constraints, enabling safe and efficient navigation in shallow-water environments with partial observability and minimal sensing. Our method introduces a novel uncertainty proxy within a Gaussian Process framework to estimate depth uncertainty from sparse sonar data, enhancing environmental awareness for decision-making under uncertainty. This is the first application of a confidence-weighted GP model in bathymetric estimation for RL-based ASV navigation. We validated our approach through extensive simulations. It recovers 64% of the performance lost from removing privileged depth-gradient information, improving success rates over baselines (classical GP without proxy and RL using only current observations) from 60.34% to 76.33%. We also demonstrated transferability to a real-world ASV, achieving reliable performance despite limited onboard computation and unmodeled environments. A key limitation is reliance on smooth GP depth estimations, which may struggle with abrupt terrain changes or unmodeled obstacles. Future work will explore enhanced sensing strategies and domain adaptation to improve robustness and reduce the sim-to-real gap under hardware and communication constraints. Adapting the confidence-aware depth estimation framework to model-based control schemes such as MPC is another promising direction, enabling direct comparison between uncertainty-aware RL and optimization-based approaches.

## References

[1] D. K. M. Kufoalor, T. A. Johansen, E. F. Brekke, A. Hepsø, and K. Trnka, "Autonomous maritime collision avoidance: Field verification of autonomous surface vehicle behavior in challenging scenarios," *J. Field Robot.*, vol. 37, no. 3, pp. 387–403, 2020.

[2] C.-H. Chang, C. Kontovas, Q. Yu, and Z. Yang, "Risk assessment of the operations of maritime autonomous surface ships," *Rel. Eng. Syst. Saf.*, vol. 207, 2021, Art. no. 107324.

[3] J.-H. Hyun, D. H. Lee, and J. C. Lee, "Bathymetric surveying using autonomous surface vehicle for shallow-water area with exposed and partially exposed rocks," *Sensors Mater.*, vol. 35, 2023.

[4] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Sci. Robot.*, vol. 5, no. 47, 2020, Art. no. eabc5986.

[5] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2096–2103, Oct. 2017.

[6] D. Madeo, A. Pozzebon, C. Mocenni, and D. Bertoni, "A low-cost unmanned surface vehicle for pervasive water quality monitoring," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1433–1444, Apr. 2020.

[7] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[8] M. J. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," in *Proc. AAAI Fall Symposia*, 2015, p. 141.

[9] M. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 465–472.

[10] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 39, pp. 1–40, 2016.

[11] D. Sanz-Alonsoand and R. Yang, "Gaussian process regression under computational and epistemic misspecification," *SIAM J. Numer. Anal.*, vol. 63, no. 2, pp. 495–519, 2025.

[12] D. Nguyen-Tuong, J. Peters, and M. Seeger, "Local Gaussian process regression for real time online model learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, vol. 21.

[13] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using Pseudo-inputs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, vol. 18.

[14] Y. Xue, X. Wang, Y. Liu, and G. Xue, "Real-time nonlinear model predictive control of unmanned surface vehicles for trajectory tracking and collision avoidance," in *Proc. 7th Int. Conf. Mechatron. Robot. Eng.*, 2021, pp. 150–155.

[15] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Trajectory tracking of autonomous vessels using model predictive control," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 8812–8818, 2014.

[16] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza, "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning," *Sci. Robot.*, vol. 8, no. 82, 2023, Art. no. eadg1462.

[17] X. Zhou, P. Wu, H. Zhang, W. Guo, and Y. Liu, "Learn to navigate: Cooperative path planning for unmanned surface vehicles using deep reinforcement learning," *IEEE Access*, vol. 7, pp. 165262–165278, 2019.

[18] E. Meyer, H. Robinson, A. Rasheed, and O. San, "Taming an autonomous surface vehicle for path following and collision avoidance using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 41466–41481, 2020.

[19] E. Meyer, A. Heiberg, A. Rasheed, and O. San, "COLREG-compliant collision avoidance for unmanned surface vehicle using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 165344–165364, 2020.

[20] L. Zhao, M. I. Roh, and S. J. Lee, "Control method for path following and collision avoidance of autonomous ship based on deep reinforcement learning," *J. Mar. Sci. Technol.*, vol. 27, no. 4, p. 1, 2019.

[21] T. N. Larsen, H. Ø. Teigen, T. Laache, D. Varagnolo, and A. Rasheed, "Comparing deep reinforcement learning algorithms' ability to safely navigate challenging waters," *Front. Robot. AI*, vol. 8, 2021, Art. no. 738113.

[22] Q. Zhang, W. Pan, and V. Reppa, "Model-reference reinforcement learning for collision-free tracking control of autonomous surface vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8770–8781, Jul. 2022.

[23] T. Wilson and S. B. Williams, "Adaptive path planning for depth-constrained bathymetric mapping with an autonomous surface vessel," *J. Field Robot.*, vol. 35, no. 3, pp. 345–358, 2018.

[24] W. Gierusz, "Modelling the dynamics of ships with different propulsion systems for control purpose," *Polish Maritime Res.*, vol. 1, pp. 31–36, 2016.

[25] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Berlin, Germany: Springer, 2003, pp. 63–71.

[26] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.

[27] J. Achiam et al., "Constrained policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 22–31.

[28] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *J. Mach. Learn. Res.*, vol. 22, no. 268, pp. 1–8, 2021.

[29] J. Ji et al., "OmniSafe: An infrastructure for accelerating safe reinforcement learning research," *J. Mach. Learn. Res.*, vol. 25, no. 285, pp. 1–6, 2024.

[30] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[31] A. Ray, J. Achiam, and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," 2019, *arXiv:1910.01708*.

[32] J. Karwowski and W. Szynkiewicz, "Quantitative metrics for benchmarking human-aware robot navigation," *IEEE Access*, vol. 11 pp. 79941–79953, 2023.