One Ocean, Many Tasks: A Data-Grounded Simulation Suite and Benchmark for Marine Robotics

Project website: https://anonymous-2026.github.io/OneOcean-IROS
Online demo: https://anonymous-2026.github.io/OneOcean-demo-IROS

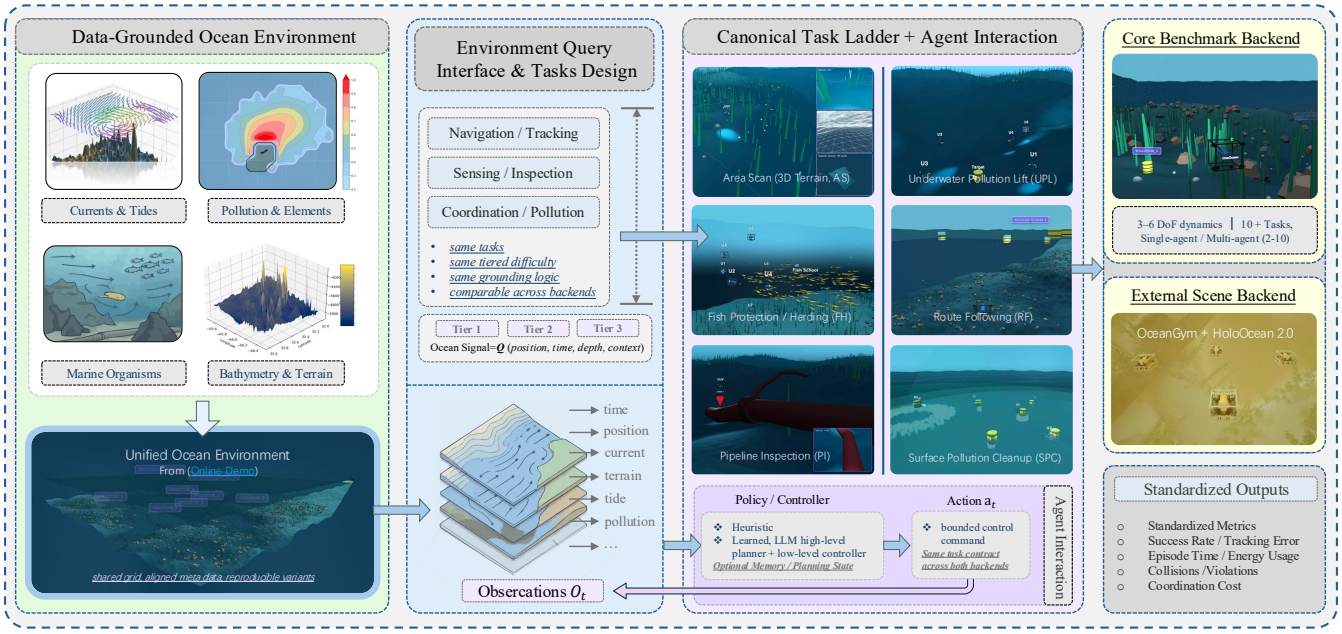


Fig. 1: OneOcean overview. A unified ocean-environment product grounds a canonical multi-task benchmark and two complementary execution backends, enabling reproducible evaluation across navigation, sensing, coordination, and pollution-related marine-robotics tasks.

***Abstract*— Simulation is essential for marine robotics, yet existing platforms often trade off between visually realistic worlds, high-throughput experimentation, and ocean-condition realism grounded in real data. We present OneOcean, a data-grounded simulation suite and benchmark built around a unified spatiotemporal ocean-environment product that harmonizes bathymetry with data-driven currents and tides, with optional pollution fields for cleanup-related evaluation. On top of this environment representation, OneOcean defines a task ladder spanning navigation and station-keeping under currents, waypoint and route following, depth-profile tracking, area scanning, pipeline inspection, and multi-agent coordination, including formation transit, fish protection/patrol (herding), and surface pollution localization, containment, and cleanup. The suite supports multi-agent settings with 2–10 vehicles. Our simulator supports 3–6 DoF vehicle dynamics and produces standardized metrics and reproducible run manifests, enabling consistent aggregation across tasks, difficulties, and scaling settings. Experiments across dataset variants and scenes systematically stress planning and control under realistic, data-driven disturbances and reveal trade-offs among heuristic baselines, behavior-cloning policies, and an LLM-based planner.**

I. Introduction

Marine robots must operate under complex, highly variable ocean conditions. In practice, **currents and tides** can dominate vehicle motion, while field trials remain expensive and time-limited and at-sea sensing/communication is constrained [1]. Simulation is therefore essential, but existing platforms often trade off between (i) visually realistic underwater worlds, (ii) high-throughput, reproducible experimentation, and (iii) **ocean-condition realism grounded in real data**.

We present **OneOcean**, a data-grounded simulation suite and benchmark for marine robotics. OneOcean is built around a unified spatiotemporal ocean-environment dataset (NetCDF) that harmonizes bathymetry, including GEBCO [2], with data-driven currents and tides from Copernicus Marine Service products (CMEMS/GOPAF [3], [4]). On top of this environment representation, OneOcean defines a **10-task ladder** spanning single-agent navigation and tracking, spatial sensing and coverage, and multi-agent coordination and pollution-related tasks, with support for teams of 2–10 vehicles. The same environment product is also packaged into multiple dataset variants so the benchmark can support fast local iteration, scene-oriented construction, and broader public-facing release without changing task semantics. Figure 2 summarizes this organization.

OneOcean provides two complementary execution back-

TABLE I: System-level comparison and task coverage.

Symbols: ● Provided/Available; ● Not Provided.

Task abbreviations: **G2G** (Go-to-Goal Under Currents), **SK** (Station Keeping), **RF** (Route Following), **DPT** (Depth-Profile Tracking), **AS** (Area Scan), **PI** (Pipeline Inspection). **FT** (Formation Transit), **SPC** (Surface Pollution Cleanup), **UPL** (Underwater Pollution Lift to Surface), **FH** (Fish Herding).

Platform / Work	Venue	Real-data In Loop	Multi-agent	Cross-backend	3D Currents	Pollution / Spill	Tide Modeling	Canonical Task	Metrics Protocol	Headless Batch	G2G	SK	RF	DPT	AS	PI	FT	SPC	UPL	FH
MIMIR-UW [5]	2023	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
HoloOcean [6]	ICRA 2022	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
HoloOcean 2.0 [7]	2025	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
OceanSim [8]	2025	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MarineGym [9]	IROS 2025	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
OceanGym [10]	2025	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
UNav-Sim [11]	ICRA 2023	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Stonefish [12]	ICRA 2025	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Volans [13]	2025	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
OneOcean (ours)	Target IROS 26	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

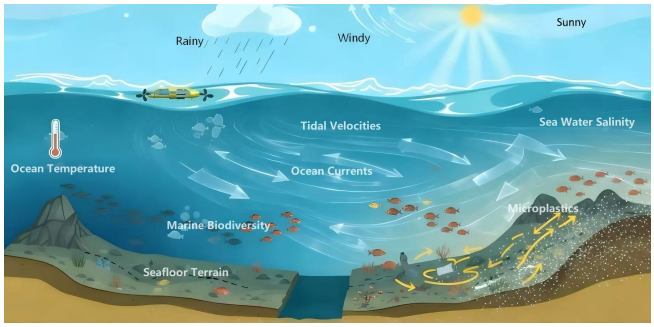


Fig. 2: OneOcean overview. A unified ocean-environment product powers a canonical task ladder and two complementary execution backends: a throughput-first core benchmark backend and an external underwater-scene backend that reuses packaged scenes while injecting the same data-grounded currents.

ends that share task definitions and evaluation semantics. The *core benchmark backend* is deterministic and throughput-first, producing standardized metrics and reproducible run manifests for paper-ready aggregation. The *external underwater-scene backend* reuses packaged underwater scenes from an existing simulator stack and injects OneOcean's data-grounded currents, enabling qualitative evidence in underwater-looking worlds while keeping the benchmark protocol consistent across backends. This split is deliberate: it lets us separate large-scale controlled benchmarking from scene-rich qualitative transfer while grounding both in the same ocean data product. In summary, our contributions are:

- A unified, reusable ocean-environment data product and tooling for exporting data-grounded disturbance signals to simulators, with variants that support fast iteration, scene construction, and broader public release.
- A canonical 10-task benchmark suite with difficulty levels, standardized metrics, auditable run manifests, and support for 3–6 DoF vehicle dynamics, with 6DoF used for the main quantitative evaluation.
- Experiments across the two backends that reveal task difficulty, scaling, and planner-compute trade-offs under realistic, data-driven disturbances.

II. Related Work

A. Underwater and marine robotics simulators

Classical simulators built on Gazebo [14], including UUV Simulator [15] and UWSim [16], provide dynamics and sensor stacks commonly used in robotics workflows, but they are not designed for data-grounded ocean-condition benchmarking at scale. More recent systems target higher-fidelity rendering and/or higher-throughput experimentation, including MARUS [17], HoloOcean [6], OceanSim on Isaac Sim [8], [18], DAVE [19], and MarineGym [9]. These platforms provide important simulator infrastructure, but they typically do not expose a reusable ocean-condition product, a fixed task ladder, and a reporting protocol that remain consistent across multiple execution backends. As a result, reproducing the same ocean-conditioned task across different simulators usually requires substantial one-off engineering. Table I summarizes this distinction at both the system and task levels.

B. Ocean models, trackers, and data products

Ocean-condition realism is commonly provided by numerical models and operational forecasting systems. Representative community models include Delft3D [20], NEMO [21], MITgcm [22], ROMS [23], and HYCOM [24]. Operational products such as the Copernicus Marine Service [3] (including GOPAF [4]) provide globally available, data-assimilative fields that can ground simulation disturbances in real observations, while toolkits such as Parcels [25] and OpenDrift [26] support Lagrangian drift/dispersion analysis. OneOcean is complementary to these efforts: we treat ocean conditions as a reusable spatiotemporal product for robotics and pair them with a benchmark protocol that can be executed on multiple simulator backends.

C. Benchmarks and evaluation protocols for marine robotics

Benchmarking in marine robotics remains difficult because realistic conditions (currents, tides, bathymetry, and environment signals) are hard to standardize, and multi-agent coordination introduces additional scaling axes. OneOcean addresses this gap by combining a canonical task suite,

standardized metrics, per-run manifests for auditability, and portability across both a throughput-first benchmark backend and an external underwater-scene backend. This positions OneOcean less as a new simulator in isolation and more as a benchmark contract that can be instantiated consistently on top of different simulation substrates.

III. OneOcean Suite

OneOcean is organized around a single principle: treat *ocean conditions* as a reusable, auditable product, and evaluate robotics methods on a canonical task ladder under these conditions. This section summarizes the environment product, how it is consumed by the benchmark backends, and the task/evaluation definitions.

A. Unified spatiotemporal ocean-environment product

The shared environment representation is a gridded spatiotemporal dataset (NetCDF), produced by our OceanEnv data pipeline. It harmonizes:

- Bathymetry/terrain, including GEBCO [2], on a queryable spatial grid.
- Data-driven ocean physics (currents and optional tide components) from Copernicus Marine Service products (CMEMS/GOPAF [3], [4]).

The product exposes standard coordinates (latitude, longitude, time, and depth when available) and current fields. It also provides bathymetry and a derived validity mask to support collision/terrain constraints.

Depth and time are treated as optional coordinates: some ocean products provide multiple discrete depth levels, while others effectively represent a near-surface or single-depth slice. In all cases, each benchmark run records the depth (and time window) it is grounded on, and the backends consume a consistent horizontal current vector field. The two backends share the same environment product but use different grounding strategies: the core benchmark backend typically selects a fixed (time, depth) slice for fast, deterministic rollouts, while the external underwater-scene backend advances a dataset clock to inject time-varying currents. This design enables repeatable 2D/2.5D disturbance injection for navigation, tracking, and coordination tasks, while leaving richer 3D current coupling as a clear extension point.

The OceanEnv pipeline crops terrain, fetches ocean-physics fields, and merges/interpolates them into a single NetCDF product with configuration recorded for reproducibility. In practice, the product is intended to be re-instantiated for a user-specified region, time window, and resolution, rather than treated as a single fixed benchmark file. A variant generator then produces multiple dataset variants, including a small "tiny" set for fast iteration, a "scene" set for scene construction, and a broader "public" set for release, that share the same schema but differ in spatial/temporal scope and resolution, and writes per-variant metadata capturing generation settings and realized time coverage.

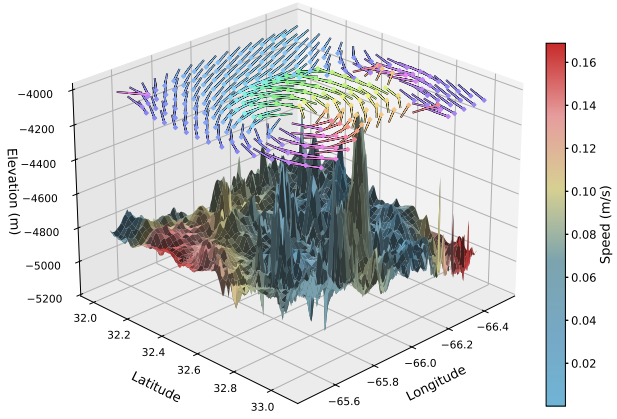


Fig. 3: Base current field visualization over bathymetry. The underlying horizontal current components are $u_{\text{data}}/v_{\text{data}}$; surface color indicates speed magnitude (m/s). View configuration: pitch $25°$, azimuth $45°$, depth index 0 m (sea surface). The terrain surface is colored by speed and overlaid with arrows indicating current direction.

B. Data-grounded disturbance signals

OneOcean exports data-grounded currents/tides into backend-friendly formats:

- **Grid export (core benchmark backend):** a compact export of gridded currents (and bathymetry/mask when needed) used for fast, deterministic rollouts.
- **Temporal export (external underwater-scene backend):** a compact time-varying current profile used to advance ocean conditions over a simulation clock.

Both backends apply currents as a data-grounded disturbance using a relative-velocity convention recorded per run. The core backend uses a world frame with x east, z north, and depth y positive down.

For reproducibility, each exported signal artifact stores its source dataset path and extraction indices, such as the time/depth slice, enabling downstream runs to be re-grounded on the same conditions. Figure 3 shows a representative visualization of the base current field grounded on our dataset over bathymetry. The underlying horizontal components are $u_{\text{data}}/v_{\text{data}}$; surface color encodes speed magnitude (m/s), and arrows indicate current direction. View configuration: pitch $25°$, azimuth $45°$, depth index 0 m (sea surface).

a) Environment queries and probes.: Tasks interact with the environment through a small set of queries that are shared across backends. At minimum, agents can query the local horizontal current vector at their position (and, when enabled, a tide-augmented current). For terrain-aware tasks, agents also query bathymetry and a validity mask to enforce clearance and prevent trajectories that cut through terrain.

For pollution-related tasks, OneOcean couples task definitions to spatial probes that read a pollution signal field, such as concentration, and define task-specific objectives and metrics such as source localization error, containment, and removal progress. In this work, these pollution fields are treated as regeneratable signals and are used primarily to

TABLE II: Canonical task ladder used in OneOcean (10 tasks). Each task is evaluated under Tier 1/2/3 difficulty settings and paired with a compact goal type and a primary task-specific metric.

Category	Task	Agents	Goal	Primary metric
Navigation	**G2G**	1	reach	final goal error
Navigation	**SK**	1	hold	station RMS
Navigation	**RF**	1	follow	cross-track error
Navigation	**DPT**	1	track	depth-tracking error
Sensing/ Coverage	**AS**	1–10	scan	coverage fraction
Sensing/ Inspection	**PI**	1–10	inspect	inspection score
Coordination	**FT**	2–10	transit	formation error
Pollution	**SPC**	2–10	clean	cleanup rate
Coordination	**UPL**	5	lift	lift success
Coordination	**FH**	8	herd	containment success

enable controlled evaluation and qualitative evidence.

C. Task ladder and difficulty levels

OneOcean defines a 10-task ladder with consistent evaluation semantics:

- **Go-to-Goal Under Currents (G2G):** reach a goal while compensating for data-grounded drift.
- **Station Keeping (SK):** hold position (and heading) under disturbance.
- **Route Following (RF):** follow a waypoint route under currents.
- **Depth-Profile Tracking (DPT):** track a target depth profile while executing a mission.
- **Area Scan (3D Terrain, AS):** scan a bounded region and reconstruct coverage over terrain.
- **Pipeline Inspection (Leak Detection, PI):** follow a pipeline corridor and localize leak cues.
- **Formation Transit (FT):** multi-vehicle transit while maintaining formation constraints ($N \in [2, 10]$).
- **Surface Pollution Cleanup (SPC):** multi-vehicle localization/containment/cleanup at the surface ($N \in [2, 10]$).
- **Underwater Pollution Lift to Surface (UPL):** cooperative lifting/salvage with a fixed team size ($N = 5$).
- **Fish Protection / Herding (FH):** herd/escort a fish school under currents with a fixed team size ($N = 8$).

Each task has increasing difficulty levels (reported as Tier 1/2/3 in the experiments), implemented via task-specific parameterizations such as current strength, required precision, coverage resolution, and coordination tightness. All tasks report standardized success/efficiency metrics plus task-specific primary metrics, including coverage fraction, formation error, and localization error.

Difficulty levels intentionally combine multiple stressors so that success requires both planning and control robustness. Typical knobs include stronger currents, tighter terminal tolerances, longer routes or larger coverage regions, stricter coordination constraints such as formation bounds, and more challenging sensing/inspection targets. Because the same environment product grounds all runs, these difficulty presets can be applied consistently across dataset variants and backends.

Across tasks, OneOcean uses a consistent template: a task defines (i) a target objective (goal, route, region, or event), (ii) what each agent observes, such as relative pose cues, local current probes, and task-specific signals, (iii) safety and feasibility constraints, including terrain clearance and team-size constraints, and (iv) success and primary metrics that are comparable across methods and backends.

We group tasks into three families. Navigation and tracking tasks (G2G, SK, RF, and DPT) test disturbance rejection and sustained tracking under data-grounded drift, with primary metrics that capture tracking quality such as cross-track error or depth deviation. Sensing and inspection tasks (AS and PI) stress coverage and event localization under currents. Coordination and pollution tasks (FT, SPC, UPL, and FH) emphasize constraint satisfaction, scaling with team size, and coupled objectives such as localization, containment, and cleanup.

D. Two execution backends

OneOcean provides two complementary execution backends that share the same task definitions and evaluation semantics. The **core benchmark backend** is headless and throughput-first: it is designed for reproducible, table-ready experimentation at scale, supports 3–6 DoF dynamics (with 6DoF used in our main experiments) [6], [15], and records per-episode metrics, configuration snapshots, and provenance for auditability and aggregation. The **external underwater-scene backend** is realism-first by scene reuse: it reuses packaged underwater scenes from an existing simulator stack (HoloOcean-style) [6] and injects OneOcean's data-grounded currents via the temporal current export, producing both quantitative outputs under the same protocol and curated media evidence for paper/web.

a) Benchmark protocol and metrics.: The core backend uses a lightweight control interface where actions specify a desired relative velocity in world coordinates, clipped to task- and vehicle-specific speed limits before integration. Let $\tilde{a}_t \in \mathbb{R}^3$ be the raw controller command, let $v_{\max}$ denote the speed limit, and define $a_t = \mathrm{clip}(\tilde{a}_t; v_{\max})$ as the executed

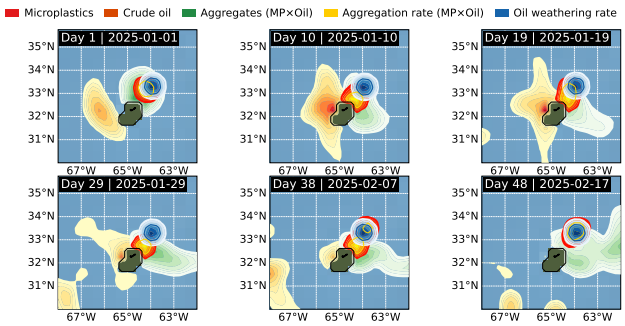


Fig. 4: Curated OCPNet multi-species diffusion/reaction pollution-field panel used as qualitative evidence. We treat these fields as regeneratable signals that can be coupled to tasks such as localization, containment, and cleanup via probes and task-specific metrics.

TABLE III: Headline core-benchmark results (6DoF): macro averages at Tier 2/3 plus Tier 3 key-task metrics. Macro Suite SR is an unweighted mean across the 10 canonical tasks; Coll is the mean collision-step ratio (%).

Method	Macro (Tier 2)			Macro (Tier 3)			Tier 3 key-task metrics					
	Suite SR(%)	E	Coll(%)	Suite SR(%)	E	Coll(%)	Go2Goal SR(%)	Route SR(%)	Station RMS	Scan cov.	Pipe score	Cleanup rate
Heuristic	**84.6**	1842.3	**5.68**	**29.0**	3352.4	8.19	**74.0**	**54.0**	**1.07**	**0.89**	**0.78**	**0.56**
BC	58.6	**1271.2**	9.72	6.0	**1635.1**	**6.41**	34.0	0.0	26.52	0.89	0.67	0.06

Episode budget: 50 episodes per task per tier. Best values per column are highlighted; collisions use radius 2.0 m.

TABLE IV: Core-benchmark planning-suite performance (Tier 2 with doubled currents): heuristic vs BC vs local LLM planners on three planning-sensitive tasks.

Method	SR avg(%)	Cleanup SR(%)	Cleanup rate	Scan SR(%)	Scan cov.	Pipe SR(%)	Pipe score	T(s)	E
Heuristic	**65.8**	60.0	**0.87**	**77.5**	0.80	**60.0**	**0.69**	**776.8**	9132.4
BC	45.0	10.0	0.32	67.5	0.79	57.5	0.61	921.6	**6908.2**
LLM-ChatGLM3-6B	55.0	60.0	**0.87**	77.5	**0.82**	27.5	0.54	860.3	9671.5
LLM-Qwen2.5-7B	47.5	57.5	0.81	57.5	0.78	27.5	0.57	1077.2	11703.8
LLM-Qwen2-7B	45.0	57.5	0.86	67.5	**0.81**	10.0	0.51	991.4	10422.1
LLM-Mistral-7B	42.5	55.0	0.82	55.0	0.78	17.5	0.52	1044.2	11288.9
LLM-Llama3-8B	44.2	**67.5**	0.84	47.5	0.77	17.5	0.59	1099.1	11614.2
LLM-Llama2-7B	30.8	55.0	**0.87**	27.5	0.75	10.0	0.46	1156.4	11654.3

Tier 2 only; episode budget: 40 episodes per task (120 per method). Best values per column are highlighted.

command. With current-scale gain g, the disturbance field is

$$c_t = g\,[u_{\text{data}}(x_t, z_t),\, 0,\, v_{\text{data}}(x_t, z_t)] + c_{\text{tide}}(t), \quad (1)$$

where u_{data} and v_{data} are the horizontal current components sampled from the environment product, and $c_{\text{tide}}(t) = A[\sin(\omega t + \varphi),\, 0,\, \cos(\omega t + \varphi)]$ is an optional synthetic tide term with amplitude A, frequency ω, and phase φ, used only in the robustness ablation. The position update in the benchmark abstraction is

$$p_{t+1} = p_t + \big(a_t + c_t\big)\Delta t, \qquad p_t = [x_t, y_t, z_t]^\top, \quad (2)$$

where Δt is the simulation step and $p_t = [x_t, y_t, z_t]^\top$ is the vehicle position. In this work we additionally integrate the 6DoF vehicle state under the same commanded translational action. We enforce hard feasibility constraints through a valid set

$$\mathcal{C} = \{(x, y, z) \mid m_{\text{land}}(x, z) < \tau_{\text{land}},\; y + \delta_{\text{clr}} \le -h(x, z)\}, \quad (3)$$

where m_{land} is the land mask, τ_{land} is the rejection threshold, δ_{clr} is the required seafloor clearance, and $h(x, z)$ is bathymetric elevation. If the proposed state leaves $\mathcal{C}$, we reject the move, keep $p_{t+1} = p_t$, and increment the constraint-violation counter. Across tasks, we report standardized metrics: success rate (SR), mean episode time T (time-to-success on successful episodes and timeout on failures, so it is always defined), and an energy proxy E defined over the clipped action sequence. For multi-agent settings we also report a near-collision ratio

$$\text{Coll} = \frac{1}{T_{\text{ep}}} \sum_{t=1}^{T_{\text{ep}}} \mathbf{1}\left[\min_{i<j} \|p_t^{(i)} - p_t^{(j)}\|_2 \le r_{\text{coll}}\right], \quad (4)$$

where T_{ep} is the episode length in steps, r_{coll} is the near-collision radius, and $\mathbf{1}[\cdot]$ denotes the indicator function. We also report task-family primary metrics, such as coverage fraction for area scan, formation error for formation transit, and localization/containment metrics for pollution tasks. Each run records task configuration, dataset grounding, controller settings, and manifest metadata so that results can be reproduced and compared across methods.

Our learned and language-based baselines fit this same interface. The behavior cloning (BC) policy maps a normalized feature vector ϕ_t—goal delta, depth, pollution probe, local current, and task one-hot—to an action prediction $\hat{a}_t = f_\theta(\phi_t)$, where θ denotes policy parameters, and is trained against heuristic actions with

$$\mathcal{L}_{\text{BC}}(\theta) = \frac{1}{N_{\text{BC}}} \sum_{t=1}^{N_{\text{BC}}} \left\| a_t^{\text{heur}} - \hat{a}_t \right\|_2^2, \quad (5)$$

where N_{BC} is the number of supervision samples and a_t^{heur} is the heuristic teacher action. The large language model (LLM) planner operates only at the high level. Every K steps it receives a compact task state summary s_k and returns a discrete plan token $g_k = \pi_{\text{LLM}}(s_k)$, such as a source assignment or waypoint assignment. A schema validator maps invalid outputs to a deterministic fallback, and the low-level controller then executes

$$a_t = \pi_{\text{low}}\big(x_t, g_k\big), \qquad t \in [kK, (k+1)K), \quad (6)$$

where x_t is the low-level state and K is the planning stride. This separation is intentional: it allows OneOcean to compare high-level reasoning methods without conflating them with low-level actuation.

b) Optional pollution fields (OCPNet).: For pollution-related evaluation, OneOcean supports optional pollution fields and probes that define task observations and metrics. In this work we use curated, regeneratable pollution-field examples for qualitative evidence and for pollution-specific task instrumentation; Figure 4 shows representative multi-species pollution-field patterns used as controlled task signals. These fields are treated as grounded task inputs rather than as immutable real-world labels.

We use an energy proxy consistent with the headless

backend implementation:

$$E := \sum_t \|a_t\|_2^2 \, \Delta t, \tag{7}$$

where a_t is the clipped action command at time step t.

IV. Experiments

We evaluate OneOcean along four axes: non-saturated method separation on the unified 10-task suite under 6DoF settings; meaningful difficulty and control/planning gaps across the tiered task ladder; success/efficiency/compute trade-offs among heuristic control, behavior cloning (BC), and local LLM planners; and transfer of the same task semantics to the external underwater-scene backend grounded by the same ocean data product.

A. Experimental setup

Backends and environment grounding. Our main quantitative results use the core benchmark backend in its 6DoF setting, with hard land-mask and bathymetry-clearance checks enabled and currents queried from a drift cache exported from the unified NetCDF environment product. We additionally report portability results on the external underwater-scene backend, which reuses packaged HoloOcean-style scenes while injecting an exported NPZ current time series driven by the same dataset clock and spatial grounding. In both cases, task definitions, success criteria, and disturbance signals are tied to the same combined ocean dataset rather than to synthetic hand-authored flow fields, so differences between the two backends are attributable to execution substrate and scene realization rather than to different environment sources.

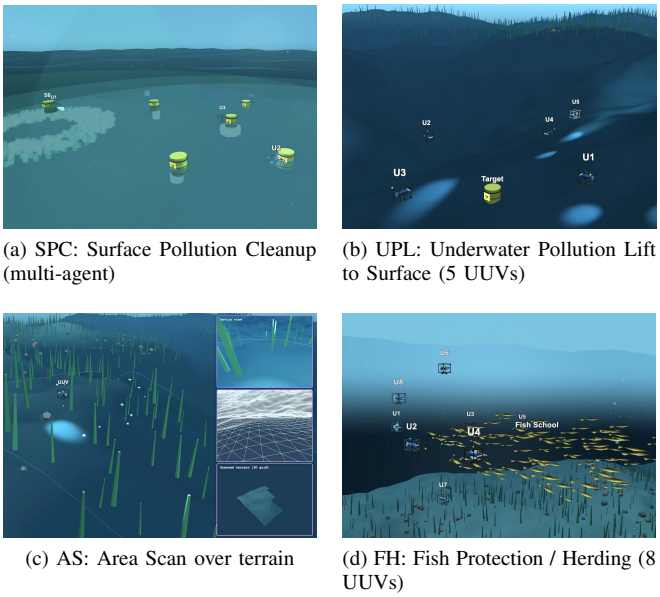
Metrics and notation. We report success rate (SR), mean episode time T, and energy proxy $E = \sum_t \|a_t\|_2^2 \Delta t$, where T uses success time on successful episodes and timeout on failures so that it remains defined for both successes and failures. For multi-agent tasks we also report Coll(%), the near-collision ratio computed from pairwise separations. Task-specific metrics include *Scan cov.* (coverage), *Pipe score* (inspection score), *Cleanup rate* (removed-mass fraction), *Station RMS* (station-keeping RMS error), *CTE* (cross-track error), *Waypoints* (reached waypoints), *Avg. SR* (planning-suite mean success), *Lat./call* (latency per uncached call), token counts, and the *current multiplier* used in the external underwater-scene backend. Tier 1/2/3 denote increasing difficulty through jointly tightened disturbance, tolerance, and coordination settings rather than through a single scalar knob.

Protocol and compute. The main core-benchmark suite uses 50 episodes per task per tier for heuristic and BC, giving a stable estimate of task-level success and efficiency over the full 10-task ladder. The planning suite uses 40 episodes per task for performance reporting and 20 episodes per task for the cost table, where the shorter protocol is sufficient because the purpose is to compare uncached local-inference overhead rather than final benchmark ranking. External-backend transfer results aggregate 4 scenes with 30 episodes per scene so that each scene contributes equally to the reported averages. Experiments were run on a workstation with **8 NVIDIA RTX 5880 Ada Generation GPUs (48GB each)**; local LLM planners use one GPU per model/run and are parallelized across devices.

TABLE V: Per-task success rates at Tier 2 and Tier 3 for the canonical 10-task suite.

Task	N	Heur T2	BC T2	Heur T3	BC T3
Navigation / Tracking					
G2G	1	**94.0**	44.0	**74.0**	34.0
SK	1	**100.0**	0.0	**14.0**	0.0
RF	1	**84.0**	24.0	**54.0**	0.0
DPT	1	**84.0**	64.0	**50.0**	0.0
Coordination					
FT	10	**100.0**	40.0	**20.0**	0.0
FH	8	94.0	**100.0**	4.0	**4.0**
Sensing / Pollution					
PI	1	**100.0**	90.0	**30.0**	14.0
AS	1	**90.0**	84.0	**14.0**	4.0
SPC	10	**84.0**	50.0	**10.0**	0.0
UPL	5	16.0	**90.0**	**20.0**	4.0

Episode budget: 50 episodes per task per tier. The highlighted rows are the tasks later revisited in the planning and external-backend transfer studies.



(a) SPC: Surface Pollution Cleanup (multi-agent)

(b) UPL: Underwater Pollution Lift to Surface (5 UUVs)

(c) AS: Area Scan over terrain

(d) FH: Fish Protection / Herding (8 UUVs)

Fig. 5: Representative core-benchmark tasks used in this work's experiments. These snapshots connect the quantitative tables to the task geometry and multi-agent interactions used in the benchmark.

B. Core benchmark backend

Figure 5 shows four representative tasks from the canonical suite. The quantitative evidence in Tables III and V establishes two main properties of the benchmark. First, Tier 2 is challenging but not saturated for all methods, making it useful for regression testing and planner comparison. Second, Tier 3 exposes large method gaps on both single-agent and multi-agent tasks, especially in station keeping, route following, surface cleanup, and leak inspection.

Figure 6 makes the ladder effect explicit on a representative task subset: the left panel shows the tier-wise success profile, while the right panel isolates where the largest drops occur as the benchmark moves from Tier 1 to Tier 3.

TABLE VI: Core-benchmark planning-suite cost metrics at Tier 2 with doubled currents: uncached local-inference calls, latency, and token usage, reported alongside average success.

Method	Avg. SR(%)	Calls	Lat. (s)	ms/ call	Prompt tok.	Output tok.	Total tok.
LLM-Qwen2.5-7B	**65.0**	6.48	14.21	2222.1	3932	4084	8016
LLM-Qwen2-7B	53.3	**4.71**	11.24	2440.3	2877	2996	5873
LLM-Mistral-7B	53.3	6.05	31.38	4942.5	3762	4128	7890
LLM-ChatGLM3-6B	53.3	5.38	**7.91**	**1393.6**	3414	**411**	**3825**
Heuristic	55.0	**0.00**	**0.00**	**0.0**	**0**	**0**	**0**
LLM-Llama3-8B	45.0	6.71	13.94	2117.5	3072	3218	6292
LLM-Llama2-7B	33.3	6.59	49.42	7401.8	4219	4851	9071
BC	21.7	**0.00**	**0.00**	**0.0**	**0**	**0**	**0**

Tier 2 only; 20 episodes/task. Costs count uncached local inference only.

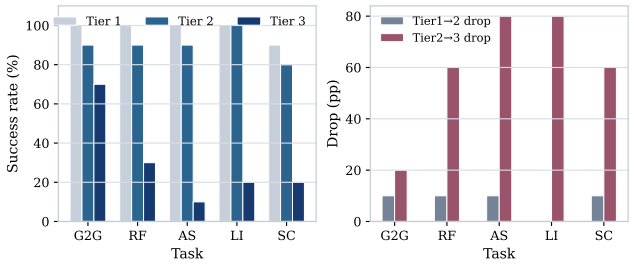


Fig. 6: Difficulty-ladder summary on a representative five-task subset. Left: success rate at Tier 1/2/3. Right: the corresponding success-rate drops from Tier 1→2 and Tier 2→3, highlighting that the hardest setting remains strongly discriminative for route following, area scan, leak inspection, and surface cleanup.

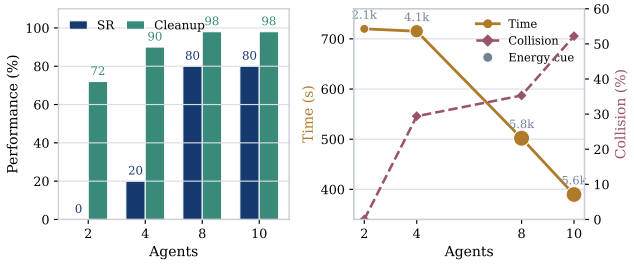


Fig. 7: Surface-cleanup scaling at Tier 2. Left: success rate and cleanup rate as the number of vehicles increases. Right: completion time and near-collision ratio, with per-point annotations showing the energy proxy in thousands. The figure highlights that scaling improves cleanup outcomes, but also shifts the coordination cost profile.

Two observations stand out. Heuristic control remains the strongest general-purpose baseline in the core benchmark backend, especially once Tier 3 activates stricter tolerances and stronger drift. BC is not uniformly weak—it remains competitive on *Underwater-Lift* and *Fish-Herding* at Tier 2—but it degrades sharply on tasks that require persistent disturbance rejection or coordinated geometry maintenance. Figure 7 further shows that the surface-cleanup task captures a meaningful team-size trade-off: larger teams improve success and removed mass, but collision risk increases and the energy budget does not scale monotonically.

a) Planning-sensitive tasks.: We next restrict attention to three planning-sensitive tasks—surface cleanup, area scan, and leak inspection—and compare deterministic heuristic control, BC, and several local LLM planners. The LLM never replaces low-level control; it only proposes high-level assignments or waypoint choices at a fixed stride, while the low-level controller and safety checks remain deterministic.

Tables IV and VI show a useful separation. The heuristic remains the most reliable overall planner, but several local LLMs are competitive on specific tasks: ChatGLM3-6B matches the heuristic on cleanup rate and scan coverage with lower latency than the other LLMs, while Qwen2.5-7B gives the best average success among the LLM family in the cost sweep.

C. External underwater-scene backend

(a) Surface pollution containment / cleanup in a packaged ocean scene.

(b) Multi-agent scene-backed deployment.

Fig. 8: Qualitative evidence from the external underwater-scene backend, which reuses packaged underwater worlds while injecting currents exported from the same combined dataset used by the core benchmark backend.

The external underwater-scene backend tests whether the same task semantics remain meaningful once we leave the throughput-first headless environment and reuse packaged underwater scenes. Figure 8 gives the visual context. Table VII shows that route-following degrades differently across scenes and difficulty levels. In a separate pollution-localization study not tabulated here, the OCPNet-3D observation model reduced localization error from 25.93 m to 20.81 m at nominal current strength and from 26.00 m to 22.64 m at the stronger current setting, while matching or improving success.

TABLE VII: Route-following across four underwater scenes after the success-criterion fix. Each scene uses 30 episodes per difficulty. "Waypoints" is the mean number of reached waypoints and CTE is cross-track error.

Scene	Tier 2			Tier 3		
	SR(%)	Waypoints	CTE [m]	SR(%)	Waypoints	CTE [m]
Dam	0.0	1.0	3.06	0.0	0.0	**0.87**
OpenWater	20.0	3.1	2.50	0.0	0.0	1.91
PierHarbor	56.7	3.5	**2.22**	0.0	2.0	1.41
SimpleUnderwater	**100.0**	**4.0**	3.04	**96.7**	**3.0**	2.16
Mean over scenes	44.2	2.92	2.70	24.2	1.25	1.59

The external underwater-scene results should be read as a portability study rather than as a replacement for core-benchmark throughput evaluation. This backend reuses realistic scenes and produces paper/web media, but it also introduces stricter scene geometry and a different disturbance hook. The value of OneOcean is that both backends still share the same task semantics, data provenance, and reporting contract.

V. Discussion and Limitations

OneOcean's results show that *data-grounded disturbances* create a useful and repeatable stress test across task families. Tier 2 supports rapid iteration and regression testing, while Tier 3 exposes meaningful failure modes in both single-agent tracking and multi-agent coordination. The method comparisons also show that success, efficiency, and compute do not collapse to a single ranking: heuristic control remains the strongest overall baseline in our current release, BC remains competitive on selected structured tasks, and local LLM planners can improve planning-sensitive behaviors at nontrivial latency and token cost.

At the same time, OneOcean remains a pragmatic abstraction rather than a full hydrodynamic simulator. Currents are applied through a recorded relative-velocity convention, the external underwater-scene backend uses an engineering disturbance hook rather than full fluid coupling, and the tide term used in our analysis is a controlled synthetic disturbance. Future work will strengthen depth-aware flow structure, tighten hydrodynamic coupling, broaden environment signals such as waves, and improve statistical reporting across seeds while preserving the same task ladder and evaluation contract.

VI. Conclusion

We introduced OneOcean, a data-grounded simulation suite and benchmark for marine robotics built around a unified spatiotemporal ocean-environment product, a canonical 10-task ladder, and standardized metrics/manifests that support paper-ready aggregation. Across a throughput-first headless benchmark backend and an external underwater-scene backend, OneOcean enables reproducible evaluation under realistic, data-driven disturbances and exposes meaningful trade-offs between heuristic control, behavior cloning, and local LLM planners. The updated experiments show that the benchmark is simultaneously useful for core numeric evaluation, robustness analysis, planner-cost analysis, and scene-backed portability studies. Future work will expand environment realism through richer tide coupling and 3D current structure, broaden task families and robot morphologies, and strengthen public release workflows for derived datasets and reproducible benchmark artifacts.

References

[1] E. Zereik, M. Bibuli, N. Mišković, P. Ridao, and A. Pascoal, "Challenges and Future Trends in Marine Robotics," *Annual Reviews in Control*, vol. 46, pp. 350–368, 2018.

[2] GEBCO Compilation Group, "The GEBCO 2025 grid – a global 15 arc-second bathymetry dataset," 2025.

[3] Copernicus Marine Service, "Copernicus Marine Service: Ocean Monitoring and Forecasting," 2024.

[4] Copernicus Marine Environment Monitoring Service (CMEMS), "Global Ocean Physics Analysis and Forecast," https://data.marine.copernicus.eu/, 2025, accessed: 2025-09-07.

[5] O. Álvarez-Tuñón, A. Amer, H. I. Ugurlu, *et al.*, "MIMIR-UW: A Multipurpose Synthetic Dataset for Underwater Navigation and Inspection," Zenodo, 2023.

[6] E. Potokar, S. Ashford, M. Kaess, and J. G. Mangelson, "HoloOcean: An Underwater Robotics Simulator," in *Proc. 2022 International Conference on Robotics and Automation (ICRA)*, 2022.

[7] B. Romrell, A. Austin, B. Meyers, *et al.*, "A Preview of HoloOcean 2.0," 2025.

[8] J. Song, H. Ma, O. Bagoren, A. V. Sethuraman, Y. Zhang, and K. A. Skinner, "OceanSim: A GPU-Accelerated Underwater Robot Perception Simulation Framework," Aug. 2025.

[9] S. Chu, Z. Huang, Y. Li, M. Lin, I. Carlucho, Y. R. Petillot, and C. Yang, "MarineGym: A High-Performance Reinforcement Learning Platform for Underwater Robotics," 2025.

[10] Y. Xue, M. Mao, X. Ru, *et al.*, "OceanGym: A Benchmark Environment for Underwater Embodied Agents," 2025.

[11] A. Amer, O. Álvarez-Tuñón, H. I. Ugurlu, *et al.*, "UNav-Sim: A Visually Realistic Underwater Robotics Simulator and Synthetic Data-generation Framework," 2023.

[12] M. Grimaldi, P. Cieslak, E. Ochoa, *et al.*, "Stonefish: Supporting Machine Learning Research in Marine Robotics," 2025.

[13] M. Moll, "Volans: A Physics Simulator for Autonomous Underwater Systems," AQ^{2}UASIM Workshop (ICRA 2025), 2025.

[14] N. Koenig and A. Howard, "Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[15] M. Manhães, S. Scherer, M. Voss, L. Douat, and F. Stein, "UUV Simulator: A Gazebo-based Package for Underwater Intervention and Multi-Robot Simulation," in *Proc. 2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, 2016.

[16] M. Prats, J. Perez, J. J. Fernandez, and P. J. Sanz, "An Open Source Tool for Simulation and Supervision of Underwater Intervention Missions," in *Proc. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[17] I. Lončar, J. Obradović, N. Kraševac, L. Mandić, I. Kvasić, F. Ferreira, V. Slošić, Đ. Nađ, and N. Mišković, "MARUS - A Marine Robotics Simulator," in *OCEANS 2022, Hampton Roads*, 2022.

[18] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," *CoRR*, vol. abs/2108.10470, 2021.

[19] M. M. Zhang, W.-S. Choi, J. Herman, D. Davis, C. Vogt, M. McCarrin, Y. Vijay, D. Dutia, W. Lew, S. Peters, and B. Bingham, "DAVE Aquatic Virtual Environment: Toward a General Underwater Robotics Simulator," 2022.

[20] G. R. Lesser, J. A. Roelvink, J. A. T. M. van Kester, and G. S. Stelling, "Development and validation of a three-dimensional morphological model," *Coastal Engineering*, vol. 51, no. 8-9, pp. 883–915, 2004.

[21] G. Madec and the NEMO Team, "NEMO Ocean Engine," *Note du Pôle de modélisation de l'Institut Pierre-Simon Laplace (IPSL)*, no. 27, 2019.

[22] E. Strobach, A. Molod, G. Forget, J.-M. Campin, C. Hill, D. Menemenlis, and P. Heimbach, "Consequences of Different Air-Sea Feedbacks on Ocean Using MITgcm and MERRA-2 Forcing: Implications for Coupled Data Assimilation Systems," *Ocean Modelling*, vol. 132, pp. 91–111, 2018.

[23] A. F. Shchepetkin and J. C. McWilliams, "The Regional Oceanic Modeling System (ROMS): A Split-Explicit, Free-Surface, Topography-Following-Coordinate Oceanic Model," *Ocean Modelling*, vol. 9, no. 4, pp. 347–404, 2005.

[24] HYCOM Consortium for Data-Assimilative Ocean Modeling, "HYCOM: Hybrid Coordinate Ocean Model," 2024.

[25] P. Delandmeter and E. van Sebille, "The Parcels v2.0 Lagrangian Framework: New Field Interpolation Schemes," *Geoscientific Model Development*, vol. 12, no. 8, pp. 3571–3584, 2019.

[26] K.-F. Dagestad, J. Röhrs, *et al.*, "OpenDrift v1.0: a generic framework for trajectory modelling," *Geoscientific Model Development*, vol. 11, no. 4, pp. 1405–1420, 2018.