AQUAWAM: A DYNAMICS-AWARE WORLD ACTION MODEL FOR UNDERWATER EMBODIED AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) are becoming increasingly important and useful for embodied intelligence, as they enable robots to anticipate the consequences of candidate actions before interacting with the physical environment. However, underwater robots are usually subject to passive dynamics, such as inertia, buoyancy, hydrodynamic drag, and persistent drift, which can continue to affect the vehicle even after an action is completed. Existing WAMs, which primarily predict action-conditioned visual observations, are not explicitly designed to capture such passive motion dynamics. In this paper, we present AquaWAM, the first World Action Model designed for underwater embodied agents. Instead of predicting future images, AquaWAM models both action-conditioned and passive physical dynamics, including the thruster dead band, the inertial glide that outlasts each command, and ambient currents. Specifically, it senses through the DVL, IMU, pressure sensor and joint encoders, while cameras supply only semantics for understanding goals and target pose. By modeling compact navigation states rather than high-dimensional visual observations, AquaWAM substantially reduces the model size and computational cost compared with conventional WAMs. Experimentally, AquaWAM achieves a 72.6% task success rate across 20 underwater tasks on the USIM benchmark, outperforming existing methods while making action decisions 2.7× faster than U0 on an NVIDIA Jetson AGX Orin. Our model also remains effective when some onboard sensor measurements are unavailable. For example, without DVL velocity measurements, our method still achieves a 61.6% success rate, compared with 39.4% for U0.

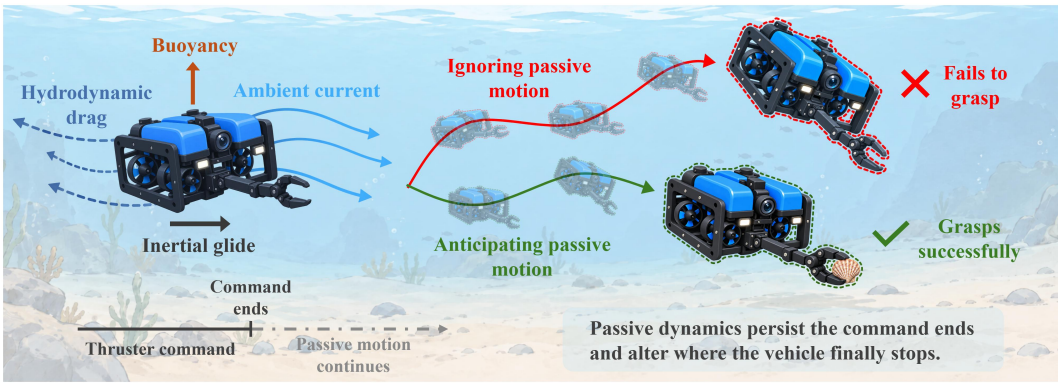


Figure 1: **Motivation for a dynamics-aware World Action Model.** Many existing World Action Models primarily predict future observations in video space, without explicitly modeling underwater vehicle dynamics. However, underwater motion is often strongly affected by passive physical factors such as buoyancy, drag, currents, and residual momentum, causing the vehicle state to continue evolving even after a control command is issued. Ignoring these dynamics can lead to inaccurate predictions of where and how the vehicle will move.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

World Action Models (WAMs) (Ye et al., 2026; Li et al., 2025; Cen et al., 2025) learn physical dynamics by jointly predicting the evolution of the world and the actions that drive it, enabling a robot to anticipate the consequences of a candidate action before executing it. Imitation-based policies lack such foresight, *i.e.,* they reproduce the actions demonstrated by experts, whereas the consequences that determine task success are often absent from the demonstrations. Recent WAMs have demonstrated the potential of action-conditioned prediction for embodied decision making, with future observations, particularly visual observations, serving as the primary representation of how the world changes in response to an action (Ye et al., 2026; Kim et al., 2026; Li et al., 2026).

Underwater embodied agents, however, operate under substantially different dynamics (Figure 1). The state of an underwater vehicle is not determined only by its current control command. Owing to the density of water and the inertia of the hull, the vehicle continues to glide once the thrusters stop, while buoyancy and hydrodynamic drag act on it irrespective of the command. Grasping, by contrast, requires the vehicle to rest within a centimeter of the object. The thrusters also exhibit a dead band, so the small corrective commands that grasping requires produce little or no thrust, and ambient currents perturb the vehicle continuously. Moreover, the Doppler velocity log (DVL) (Kinsey et al., 2006; Paull et al., 2014), which provides velocity and, by integration, position, may lose bottom lock at any time. Since these effects are not produced by the current command, they can only be anticipated.

Existing WAMs, including DreamZero (Ye et al., 2026), Cosmos Policy (Kim et al., 2026) and LingBot-VA (Li et al., 2026), however, predict in video space, which is ill-suited to this setting in both computational cost and prediction target. Furthermore, video-based WAMs typically rely on large generative backbones that are costly to deploy on embedded platforms. More importantly, visual prediction provides only an indirect representation of the vehicle's future motion state, particularly under degraded underwater visibility.

This observation motivates a different formulation of World Action Models for underwater embodied intelligence: rather than asking *what will the next image look like after this action?* an underwater agent should reason directly about *how its physical state will evolve under both commanded and passive dynamics*. Such a model should capture not only the immediate response to control inputs, but also motion that persists beyond individual commands and environmental effects that occur without explicit control. At the same time, its prediction should remain sufficiently lightweight for evaluating multiple candidate actions onboard resource-constrained robotic platforms.

In this paper, we present AquaWAM, the first World Action Model specifically designed for underwater embodied agents. AquaWAM retains the defining capability of a WAM, *i.e.,* predicting the outcome of a candidate action before execution, but replaces video prediction with a compact 35-dimensional underwater physical state (Section 3) measured by the DVL, inertial measurement unit (IMU) (Paull et al., 2014), pressure sensor (Kinsey et al., 2006) and joint encoders (Sivčev et al., 2018), together with the object pose relative to the gripper. Camera images are excluded from the dynamics model and are used only by perception modules to estimate task goals and object poses. We train AquaWAM on both the USIM demonstrations (Gu et al., 2025) and task-agnostic play data (Lynch et al., 2019); the former cover the commanded motions of the 20 tasks, while the latter expose a broader range of low-magnitude, transitional, and unstructured motions, allowing the model to learn effects such as thruster dead bands and residual glide directly from data.

Experimentally, AquaWAM achieves a success rate of 72.6% on the 20 USIM tasks, compared with 51.7% for U0 (Gu et al., 2025), the strongest model reported on the benchmark, with both fine-tuned on the same demonstrations under one protocol; all other vision-language-action models (VLAs) (Kim et al., 2024; Physical Intelligence et al., 2025; NVIDIA et al., 2025a; Zheng et al., 2026; Shukor et al., 2025) perform worse (Table 1). Given the object pose provided by the benchmark, the predicted thrust pulses grasp in 85.2% of trials, compared with 17.9% for a hand-tuned positioning law, indicating that the gain stems from prediction rather than tuning (Section 4.2). When the DVL fails, the same predictor serves as the estimator that replaces it, and AquaWAM still succeeds in 61.6% of trials, compared with 39.4% for U0 (Section 4.3).

Furthermore, as the model size is determined by the state dimension rather than the input resolution, AquaWAM is compact in both parameters and computation. It uses 2.4M parameters for the 19-

dimensional vehicle state during cruising and 2.6M for the full 35-dimensional state near an object, compared with 2.7B for U0, the strongest baseline. Evaluating 128 candidate action sequences over a two-second horizon therefore takes only 22 ms on a Jetson AGX Orin (Karumbunathan, 2022), so that every thruster command is selected from predicted outcomes and re-planned every 0.5 s; a complete decision, perception included, is 2.7× faster than that of U0 on the same hardware.

2 Related Work

2.1 Underwater State Estimation and Control

Underwater vehicles have long relied on model-based control, including hydrodynamic equations of motion (Fossen, 2011), thruster models that capture sluggish and dead-banded actuation (Yoerger et al., 1990; Bachmayer et al., 2000), model predictive control (Shen et al., 2018; Heshmati-Alamdari et al., 2020), and learned approaches that identify vehicle dynamics or replace the controller (Wehbe et al., 2017; Carlucho et al., 2018). When the DVL loses bottom lock, navigation typically falls back on inertial dead reckoning (Kinsey et al., 2006; Paull et al., 2014) or learned velocity estimation (Topini et al., 2020; Cohen et al., 2023). These approaches generally treat vehicle dynamics, task-level planning, and state estimation as separate components: the dynamics model supports control, while velocity estimation is handled by a dedicated module. In contrast, AquaWAM predicts the future physical state produced by each candidate action and selects commands directly from those predictions. The same predictor provides velocity when the DVL fails (Section 3.5), so that action prediction and navigation-state estimation share one model.

2.2 Vision-Language-Action Models

Vision-language-action (VLA) models adapt pretrained vision-language models to generate robot actions. RT-2 (Brohan et al., 2023) and OpenVLA (Kim et al., 2024) discretize action dimensions into tokens, whereas π_0 (Black et al., 2025), $\pi_{0.5}$ (Physical Intelligence et al., 2025), GR00T N1 (NVIDIA et al., 2025a), SmolVLA (Shukor et al., 2025), and the cross-embodiment X-VLA (Zheng et al., 2026) generate continuous action chunks by flow matching. USIM extends this paradigm to underwater embodied tasks, where U0 outperforms OpenVLA, $\pi_{0.5}$, and GR00T N1.5 trained on the same demonstrations. Despite their architectural differences, these models are trained primarily to reproduce expert actions rather than to predict their consequences. Four of the six VLAs we compare against also commit to 16-step chunks, corresponding to 1.6 s of motion without intermediate outcome evaluation (Appendix E). This is particularly limiting underwater, where inertia, buoyancy, drag and currents keep acting on the vehicle after a command has ended. These dynamics motivate selecting actions by their predicted outcomes rather than by imitation alone.

2.3 World Models

World models provide this predictive capability by modeling how the world evolves under actions. A learned world model (Ha & Schmidhuber, 2018) may train a policy inside imagined trajectories (Hafner et al., 2020; 2025) or plan at test time by rolling candidate action sequences forward (Chua et al., 2018; Hafner et al., 2019; Hansen et al., 2024). Video world models predict future frames from actions (Hu et al., 2023; Yang et al., 2024; NVIDIA et al., 2025b) or from latent actions inferred from video (Bruce et al., 2024), while VPP (Hu et al., 2025) and UVA (Li et al., 2025) build robot policies on video prediction models. More recent World Action Models, including DreamZero, couple future-world prediction more closely with action generation or selection, with video serving as a dense representation of the evolving world. DreamZero, for example, uses a 14B video-diffusion backbone and substantial model and system optimization to reach 7 Hz. We adopt the same predictive-action perspective, but treat video as one possible representation rather than a requirement. AquaWAM follows model-predictive action selection, while replacing video with the vehicle's physical state measured by its onboard sensors. This representation exposes the quantities that determine underwater control directly, including velocity, angular motion, depth, arm state, and object pose, while remaining compact enough to evaluate many candidate actions onboard.

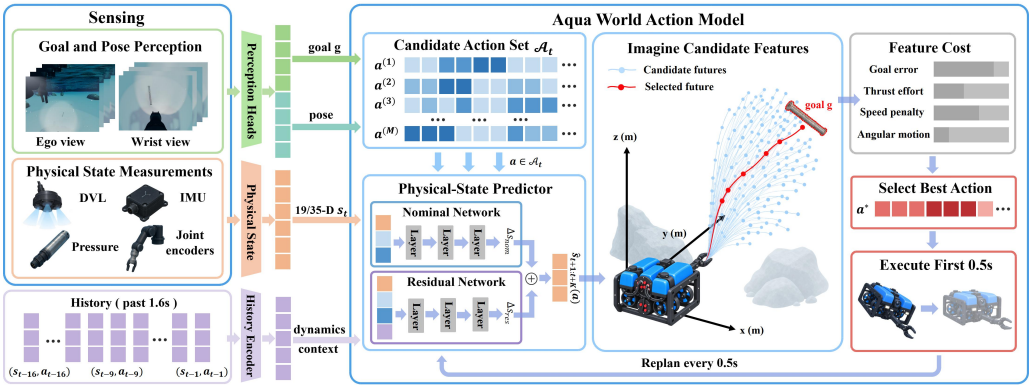


Figure 2: **Overview of AquaWAM.** Cameras supply the goal and the object pose; the predictor imagines each candidate two seconds ahead and executes the first 0.5 s of the best.

3 AquaWAM: A Dynamics-aware World Action Model

AquaWAM keeps the interface of the VLAs it is compared with (Gu et al., 2025; Physical Intelligence et al., 2025; NVIDIA et al., 2025a; Zheng et al., 2026; Shukor et al., 2025; Kim et al., 2024): two camera images, the DVL, IMU, pressure sensor and joint encoders, and the task instruction come in, and a chunk of thruster and joint commands goes out. What differs is how the commands are produced (Figure 2). Given the state s_t at control step t, the recent history $\mathcal{H}_t$ and a goal g,

$$\mathcal{W}_\theta : (s_t, \mathcal{H}_t, g) \longmapsto \big(\hat{s}^\star_{t+1:t+H},\ a^\star_{t:t+K}\big), \tag{1}$$

AquaWAM ($\mathcal{W}_\theta$), with parameters θ, returns the future it has chosen to bring about two seconds ahead ($H = 20$ steps of 0.1 s), and the first half second of the commands a that achieve it ($K = 5$ steps); the star marks the chosen candidate. A single predictor plays three roles: it predicts the physical state under any command (Section 3.1); it selects every command from these predictions, from cruising to the thrust pulses that cross the dead band (Sections 3.2 and 3.3); and it replaces the DVL when it fails (Section 3.5). Cameras supply only goals and target pose (Section 3.4).

3.1 Physical State and Passive Dynamics

The state consists of what the sensors read (Table 4 in Appendix A). Body velocity and altitude from the DVL, angular rate and acceleration from the IMU, depth from the pressure sensor and the last thruster command form the compact 19-dimensional vehicle block. Within reach of an object, the joint encoders and the wrist camera add the arm and the object's pose relative to the gripper (Section 3.4), giving 35 dimensions. Both scales share one architecture and one training recipe.

A recurrent encoder E_ϕ with parameters ϕ summarizes the last 1.6 s of states and commands, $\mathcal{H}_t$, into a disturbance token $d_t = E_\phi(\mathcal{H}_t)$ that captures what no sensor measures directly, namely the ambient current and the efficiency of each thruster. A nominal network f_{nom} and a residual network f_{res} then predict the next half second of the state as an offset from the present,

$$\hat{s}_{t+1:t+K} = s_t + f_{\text{nom}}(s_t, a_{t:t+K}) + f_{\text{res}}(s_t, a_{t:t+K}, d_t), \qquad K = 5 \text{ steps at } 10 \text{ Hz}. \tag{2}$$

Chaining four such predictions along a candidate command sequence $\mathbf{a} = a_{t:t+H}$ yields the predicted future $\hat{s}_{t+1:t+H}$, written $\text{imagine}(s_t, \mathcal{H}_t, \mathbf{a})$. We use two seconds because it is the shortest horizon over which the hull's own response, including its overshoot, becomes visible.

The predictor thus learns both the action-conditioned response and the passive dynamics that decide underwater tasks. First, the inertial glide outlasts each command: once thrust is cut, speed decays with a time constant of 2.1 s, and the offset in Eq. 2 carries this decay even under a zero command. Second, the thrusters exhibit a dead band: commands below about 0.12 change the speed by less than 0.4 cm/s per second, whereas commands between 0.3 and 0.6 change it by about 4 cm/s per second. Third, buoyancy and drag act between commands, and the ambient current is a persistent drift that only the history reveals, hence the disturbance token. None of these effects is hand-coded; they are learned from the demonstrations and from task-agnostic play (Appendix D). Play exposes the dead band and the glide, and removing it costs the most navigation success (Section 4.5).

3.2 PREDICTION BEFORE ACTION

The action head of a VLA usually produces the command an expert issued. Our AquaWAM, in contrast, has no such a action head, *i.e.,* its command selects from best of 128 predicted futures, one per candidate in a set $\mathcal{A}_t$ drawn in two rounds, and the decision is revised every half second. Algorithm 1 in Appendix A traces one decision from the sensor readings to the executed command. The candidates are structured: a third come from a thrust-allocation law over a grid of gains, others repeat the present command or move along one axis, and the rest are snippets of play. Each candidate is rolled out two seconds ahead and scored by a evaluation function J, specifically,

$$J(\mathbf{a}) = \lambda_g \sum_{k=1}^{H} \big\|\hat{v}_{t+k} - v_g\big\| + \lambda_u \sum_{k=0}^{H-1} \|a_{t+k}\|^2 + \lambda_s \Big[\max_k \|\hat{v}_{t+k}\| - v_{\max}\Big]_+ + \lambda_\omega \sum_{k=1}^{H} \|\hat{\omega}_{t+k}\|, \quad (3)$$

where k counts the predicted steps, $\hat{v}$ and $\hat{\omega}$ are the predicted body velocity and angular rate, v_g is the goal velocity, $v_{\max}$ the speed cap, $[x]_+ = \max(x, 0)$, and λ_g, λ_u, λ_s and λ_ω weight goal tracking, thrust, excess speed and spin. The second round is drawn around the sixteen best candidates of the first. Since the thrust law is only one candidate among many, it loses whenever the predictor foresees a glide or a dead band that the law ignores. Only the first half second of the winner is executed, and both rounds take 22 ms on a Jetson AGX Orin (Section 4.6). USIM's own task schema orders the sub-goals and supplies v_g, while every motion within it is chosen by prediction (Appendix C).

3.3 THRUST PREDICTION FOR FINE POSITIONING

For grasp operation, we usually require a find positioning of robot arm. In our design, a grasp require adjust in centimeters, as the judge counts it only when the gripper closes on the object inside a window of ± 0.7 cm on its tightest axis. This poses a dilemma: the corrections that reach the window are small, small commands fall inside the dead band, and any command beyond it makes the hull glide for two seconds. A continuous law must sacrifice one of the three.

AquaWAM instead predicts discrete pulses. Within reach, the candidates become thrust pulses of direction $\mathbf{u}$, magnitude m and duration τ, each followed by a coast, and each is scored on where it ends rather than where it passes:

$$\begin{gathered} \mathcal{A}_t^{\text{pulse}} = \{\text{hold}\} \cup \big\{\text{pulse}(\mathbf{u}, m, \tau) \oplus \text{coast}\big\}, \\ \mathbf{u} \in \big\{\boldsymbol{\delta}/\|\boldsymbol{\delta}\|,\ \text{sign}(\delta_i)\,\mathbf{b}_i\ (i = x, y, z),\ -v_t/\|v_t\|\big\}, \quad m \in \{0.12, \ldots, 0.40\}, \quad \tau \le 0.4\,\text{s}, \\ J^{\text{pulse}}(\mathbf{a}) = \|\hat{e}_{\text{end}}\|_Q^2 + \lambda_v \|\hat{v}_{\text{end}}\|^2, \qquad Q = \text{diag}\big(1/0.02^2,\ 1/0.007^2,\ 1/0.02^2\big). \end{gathered} \quad (4)$$

The magnitudes start where the recordings show the hull begins to respond, so every thrust pulse crosses the dead band by construction. Here hold lets the hull glide without translational thrust, $\oplus$ denotes concatenation in time, $\boldsymbol{\delta}$ is the body-frame displacement from the gripper to its target point above the object, $\mathbf{b}_i$ are the body axes, $-v_t/\|v_t\|$ brakes against the current velocity, $\hat{e}_{\text{end}}$ is the predicted gripper–object offset $e = (x, y, z)$ over the last third of the horizon, $\hat{v}_{\text{end}}$ the predicted velocity at its last step, $\|e\|_Q^2 = e^\top Q e$ with Q making the ± 0.7 cm tolerance on y dominate, and λ_v weights coming to rest (Appendix B). The pulse whose predicted glide ends inside the window at rest is selected, and because the last step of every candidate is a coast, the executed motion matches the predicted one even if the next decision is late. The hull acts as the fine manipulator: the arm's lateral reach is only ± 3 cm, so the arm holds a ready pose and the hull does the centimeter-level positioning (Appendix C). The gripper, the only discrete command, also closes by prediction, when an outcome model trained on 3,488 recorded closures predicts a firm hold with probability at least 0.7.

3.4 SEMANTIC DECODING FOR OBJECTIVES

Camera images are not directly used by the predictor. Instead, we have a perception heads, *i.e.,* a DINOv2-base backbones (Oquab et al., 2024) fine-tuned on USIM, convert both camera images and the instruction into the quantities the predictor works with, meters and meters per second. A navigation head predicts the displacement to the current waypoint three seconds ahead, which serves as the goal, and a wrist-camera head predicts the target pose, *i.e.,* the object's position and orientation in the gripper frame with a per-axis uncertainty, and locates the destination box for transport. Each of USIM's nine instructions names a task and a goal type. The heads are trained on the demonstrations

and on AquaWAM's own rollouts relabeled with the reference goals (Ross et al., 2011), so that they see the states AquaWAM visits rather than only those of the experts, while the predictor stays fixed.

3.5 AquaWAM as a Navigation-State Estimator

We further simulate a case when the DVL reports a loss of bottom lock, the state loses velocity and, through it, position. Vehicles usually carry a separate module for this, such as an inertial filter or a learned estimator. AquaWAM does not: prediction and estimation are two readings of the same network (Figure 3),

$$\begin{aligned} \text{imagine}: \ (s_t,\ \mathcal{H}_t,\ \mathbf{a}) &\mapsto \hat{s}_{t+1:t+H}, \\ \text{estimate}: \ \mathcal{H}_t^v &\mapsto \hat{v}_t. \end{aligned} \tag{5}$$

Estimation, the second reading in Eq. 5, masks the velocity channels of the history, $\mathcal{H}_t^v$, recomputes the disturbance token from the remaining channels, and asks three velocity heads trained on such masked histories what the IMU readings and the commands imply; their mean is the estimate and their spread is σ_t. The estimate takes the place of the DVL, the IMU attitude and the pressure sensor complete the dead-reckoned pose, and the target pose is re-anchored in that frame. Since a blind estimate should not be trusted equally at all times, a cumulative statistic S_t accumulates the deviation z_t of the estimate from its value v_0 at the moment of loss, in units of the spread σ_t plus a noise floor σ_0, with a forgetting factor γ and a slack κ,

$$z_t = \frac{\|\hat{v}_t - v_0\|}{\|\sigma_t + \sigma_0\|}, \qquad S_t = \max\big(0,\ \gamma S_{t-1} + z_t - \kappa\big), \qquad \alpha_t = \min(1,\ S_t/\eta). \tag{6}$$

The trust α_t reaches one when S_t reaches the level η, and it bounds how far the new decision may depart from the last one made with the DVL. Both κ and η are calibrated on held-out blind windows to a chosen false-alarm rate, and a known change of goal makes trust immediate. When cruising toward a fixed goal, AquaWAM first brakes into the speed range where the estimate is most accurate, while an attitude guard keeps it level. The estimator is retrained on the vehicle's own blind flights and agrees with the predictor by construction, since it reports the velocity the predictor would forecast for the executed command. AquaWAM therefore steers blind with the same physics it predicts with (Section 4.3).

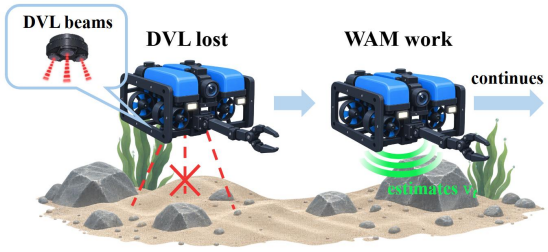


Figure 3: **AquaWAM as a navigation-state estimator.** After DVL loss, it can still estimate the velocity.

4 Experiments

We evaluate on USIM, the benchmark introduced together with U0. It simulates a BlueROV2-class vehicle with an arm in Stonefish (Cieślak, 2019) and comprises 20 tasks in four families (6 navigation, 12 grasping, 1 transporting, 1 tracking), with 700 trials per model allocated as the benchmark prescribes (160, 480, 40 and 20). Every model sees the same two cameras, issues the same 13 commands and is scored by the same judges. Under *DVL dropout*, the DVL loses bottom lock partway through every trial and never recovers (8 s into a goto, 20 s into tracking, 30 s into a grasp or transport, 40 s into an inspection or scan), with unchanged judges. Figure 4 shows examples.

All baselines, namely U0, $\pi_{0.5}$, GR00T N1.5, X-VLA, SmolVLA and OpenVLA, are fine-tuned on the USIM demonstrations (Appendix E). We report successes per family and the overall success rate (SR), together with the quality metrics as defined by USIM: success weighted by path length (SPL) (Anderson et al., 2018), average success duration (ASD) and stage success rate (SSR).

4.1 Main Results

Under full sensing, AquaWAM succeeds in 72.6% of the 700 trials, compared with 51.7% for U0 (Table 1). The margin holds in every family and is largest where the water matters most: grasping

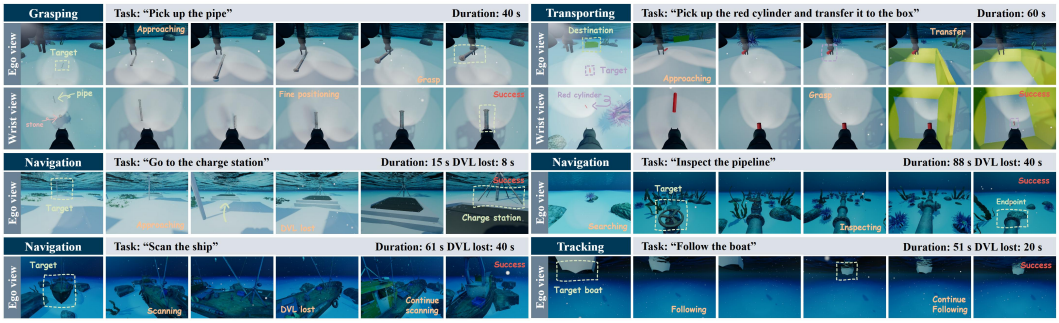


Figure 4: **Execution traces of AquaWAM.** Grasping and transporting run with full sensing; navigation and tracking lose the DVL where marked and finish on the estimate.

Table 1: **Results on the 20 USIM tasks.** Succ. denotes the number of successes trials; SPL and ASD follow the USIM definitions. The **best** and second-best are highlighted.

Model	Navigation		Grasping		Transporting		Tracking	Overall
	Succ.↑	SPL↑	Succ.↑	ASD (s)↓	Succ.↑	SSR↑	Succ.↑	SR↑ (succ./trials)
OpenVLA	50/160	0.71±0.19	0/480	–	0/40	0.0%	0/20	7.1% (50/700)
$\pi_{0.5}$	58/160	0.59±0.19	1/480	87.3	0/40	0.0%	0/20	8.4% (59/700)
GR00T N1.5	122/160	0.78±0.23	140/480	102.2	10/40	32.5%	4/20	39.4% (276/700)
X-VLA	85/160	0.64±0.24	4/480	63.5	0/40	0.0%	1/20	12.9% (90/700)
SmolVLA	74/160	0.64±0.25	7/480	94.7	0/40	5.0%	1/20	11.7% (82/700)
U0	140/160	0.75±0.24	195/480	99.8	13/40	52.5%	14/20	51.7% (362/700)
AquaWAM (Ours)	**151/160**	**0.84±0.21**	**314/480**	**62.9**	**23/40**	**65.0%**	**20/20**	**72.6% (508/700)**

improves from 40.6% to 65.4%, transporting from 32.5% to 57.5% and tracking from 70% to 100%, while navigation rises from 87.5% to 94.4%. GR00T N1.5, U0's backbone, reaches 39.4%.

The quality metrics follow the success rates: AquaWAM's successful grasps take 62.9 s compared with 99.8 s for U0, and its navigation paths are the most efficient (SPL 0.84).

4.2 Grasping Experiments

Figure 5 breaks the grasping margin down by task: U0 succeeds in 12 to 20 of 40 trials per object and AquaWAM in 20 to 34, outperforming U0 on every object. To identify its source, we fix everything except the decision rule, using the same task program, closure thresholds and the benchmark's own object pose, so that perception plays no part. In this setting, the hand-tuned positioning law of Section 3.3 grasps in 17.9% of the 480 trials and the predicted pulses in 85.2%, the inner and outer rings of Figure 5. The law already fails during the approach: only 24% of its trials bring the gripper inside the closure window, while the rest dither in the dead band.

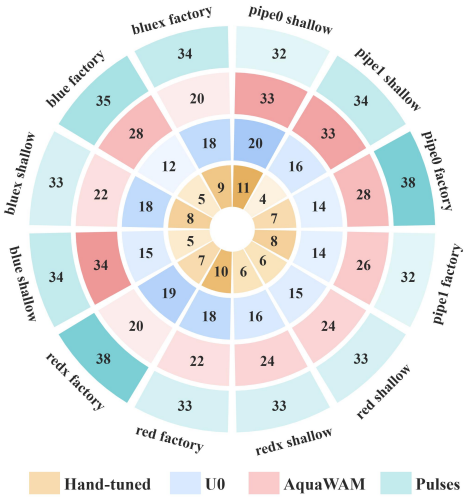


Figure 5: **Prediction, not tuning, drives the grasp margin.** Successes of 40 per task.

We further rule out three alternative explanations on the same tasks. For cadence, at AquaWAM's half-second rhythm the law reaches 12/40 and 11/40 on two tasks (11 and 6 at 1.6 s), against 32/40 and 33/40 for the pulses. For the trigger, the learned closure model turns 32/40 and 33/40 into 35/40 and 34/40, so the moment of closing does not explain the margin. For the grid, pulse grids of 2×2, 5×4 (deployed) and 9×4 give 34, 32 and 31 of 40. The margin thus requires a predictor that knows where a pulse ends. AquaWAM itself reads the object pose from the wrist camera and grasps in 65.4% (Table 1), so the remaining gap to 85.2% lies in perception.

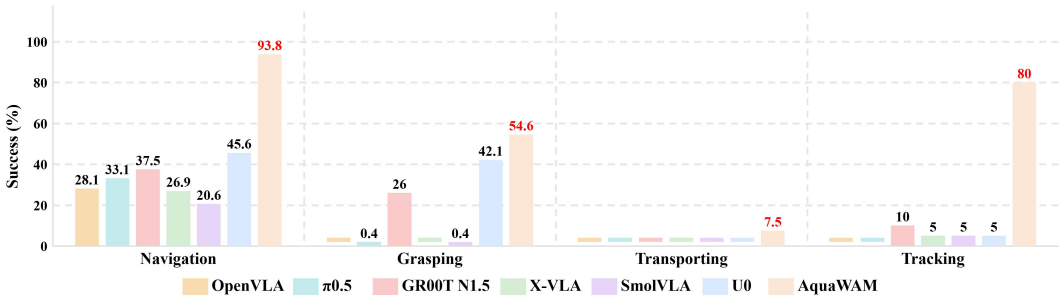


Figure 6: **Success under DVL dropout, per task family.** The DVL loses bottom lock partway through every trial and never recovers. Colored baseline marks denote zero success.

Table 2: **Results on extreme cases, *i.e.,* USIM-Hard.** Left: perturbed worlds, 20 trials per task. Right: stricter judges re-score the trials of Table 1, and in the DVL-loss rows those of Section 4.3.

Perturbed world	U0	AquaWAM	Stricter judge	U0	AquaWAM
Start heading rotated 90–180°	52/80	63/80	Goto, all waypoints within 1.0 m	53/80	61/80
Goto endpoint within 0.5 m, not 1 m	26/40	35/40	Goto, all waypoints within 0.5 m	44/80	44/80
Thruster 3 at 50 %, seen in play	3/40	30/40	Goto, 0.5 m waypoints, DVL loss	28/80	45/80
Thruster 1 at 50 %, unseen	4/40	21/40	Goto, all waypoints within 0.25 m	25/80	24/80
Thruster 1 at 0 %, unseen	1/40	4/40	Inspect, 1 m ball, 90 % coverage	36/40	39/40
Turbid water, Jerlov 0.50	1/20	9/20	Inspect, same judge, DVL loss	7/40	39/40

4.3 When the DVL Fails

When the DVL loses bottom lock, the models of Table 1 separate (Figure 6). U0 falls from 51.7% to 39.4%: its navigation halves (87.5% to 45.6%), its tracking nearly disappears (70% to 5%), and only grasping, which relies on sight, holds at 42.1%. GR00T N1.5 falls from 39.4% to 26.7%. AquaWAM retains 61.6%: navigation 93.8% and tracking 80% on dead reckoning, and grasping 54.6%, where the approach runs on the estimate until the wrist camera re-anchors the object. Transporting, which carries an object 2.9 m after the loss, remains low for both (7.5% against 0%).

The estimator also transfers to other policies. Used as a takeover layer that stands in for U0's DVL once it reports loss, it restores U0's navigation (96.9%) and most of its tracking (60%), raising U0 from 39.4% to 49.7%; the remaining twelve points to AquaWAM lie in grasping and tracking.

4.4 Experiments on Extreme Cases

Since the official judges saturate near the top of Table 1, we tighten them and perturb the world (Table 2). Under stricter judges, AquaWAM passes every waypoint within 1.0 m in 61 of 80 gotos against 53 for U0, the two are level at 0.5 m with 44 each, and U0 keeps 25 against 24 at 0.25 m. This is expected, since AquaWAM advances once it is within 0.9 m of a waypoint (Appendix C) and is never asked to pass closer; nevertheless, on the harder water-tower route it leads at every tolerance (Appendix F). With the DVL lost, AquaWAM still passes the 0.5 m waypoints in 45 of 80 gotos against 28 and completes 39 of 40 inspections against 7. Under perturbation, AquaWAM outperforms U0 on every protocol, by 27 of 40 with thruster 3 at half thrust and by 8 of 20 in turbid water, and with thruster 1 delivering no thrust the two gotos drop to 4 and 1 of 40 (Appendix F).

4.5 Ablations

Table 3 removes the key components of AquaWAM one at a time on the same 700 trials. Without imagination, *i.e.,* with a single forward pass of an amortized head in place of the predicted rollouts, grasping falls from 65.4% to 16.5% and transporting from 57.5% to 7.5%, while navigation remains at 79.4%, indicating that the command crossing the dead band must be predicted. Without the play data, the predictor never observes the glide or the dead band, and navigation drops to 50.6% with SPL 0.53. Without the disturbance token, the predictor cannot distinguish the ambient current from its own motion, and overall success falls to 305 of 700. With the IMU alone standing in for the DVL, success falls to 184 of 700 trials, against 431 with AquaWAM as DVL, and tracking falls from 16/20 to 4/20. Each component helps most where the dynamics it models matter.

Table 3: **Ablation studies.** Successes are given as succeeded/trials; the last two rows run when DVL is not available. SPL (mean ± s.d.) and ASD (s) follow definitions from USIM dataset. We simulate a case when DVL fails and alternatively use AquaWAM/IMU to predict/calculate velocity.

Model	Navigation		Grasping		Transporting		Tracking	Overall
	Succ.↑	SPL↑	Succ.↑	ASD (s)↓	Succ.↑	SSR↑	Succ.↑	SR↑ (succ./trials)
AquaWAM	**151/160**	**0.84±0.21**	**314/480**	**62.9**	**23/40**	**65.0%**	**20/20**	**72.6% (508/700)**
w/o imagination	127/160	0.69±0.22	79/480	93.7	3/40	17.8%	14/20	31.9% (223/700)
w/o play data	81/160	0.53±0.24	138/480	97.3	7/40	26.3%	13/20	34.1% (239/700)
w/o disturbance token	89/160	0.59±0.23	193/480	81.7	11/40	41.7%	12/20	43.6% (305/700)
AquaWAM as DVL	**150/160**	**0.82±0.21**	**262/480**	**65.8**	**3/40**	**67.5%**	**16/20**	**61.6% (431/700)**
IMU as DVL	86/160	0.49±0.26	93/480	103.6	1/40	13.2%	4/20	26.3% (184/700)

4.6 Computation Costs

AquaWAM is designed to run on the vehicle. On a Jetson AGX Orin, evaluating 128 candidates two seconds ahead in two rounds takes 22 ms and a pulse decision 19 ms. Including the perception heads, a decision costs 79.5 ms on a navigation step and 95.7 ms on a grasp step, measured end to end on replayed sensor streams, which is 2.7× faster than U0's 214 ms for each 1.6 s action chunk. Per inference call, GR00T N1.5 needs 236 ms, X-VLA 301, SmolVLA 752, OpenVLA 910 and $\pi_{0.5}$ 1167 (Figure 7). The 2.4M and 2.6M parameters belong to the predictor; the DINOv2-base backbones of the perception heads add 86M each, and their time is included in every AquaWAM figure above.

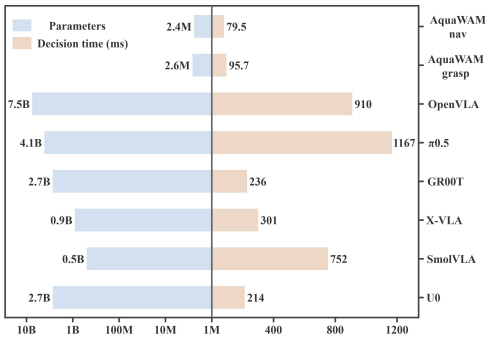


Figure 7: **Computation cost on Jetson AGX Orin 32 GB.** We show the number of parameters (left) and the running time (right).

5 Conclusion

In this paper, we presented AquaWAM, the first World Action Model for underwater embodied agents. AquaWAM retains the defining capability of a WAM, predicting what a candidate action will do before acting, but changes what is predicted: a 35-dimensional physical state read from the vehicle's own sensors, whose passive dynamics are learned from task-agnostic play. This design yields three benefits. First, the model runs on the vehicle's own computer, taking 79.5 ms per decision on a Jetson AGX Orin. Second, its predicted thrust pulses cross the dead band and grasp within a centimeter where a hand-tuned law fails. Third, the same predictor replaces a failed DVL, both for itself and as a takeover layer for a VLA.

Limitations. All results are obtained in simulation: USIM models the hydrodynamics that AquaWAM learns from, but a real deployment adds effects it only approximates, such as sensor noise, latency and intermittent DVL dropouts. Although AquaWAM outperforms the other models, it still struggles in three cases. Transport after DVL loss remains at 7.5%, since the object must still be carried 2.9 m on an estimated velocity; a thruster that delivers no thrust, a fault absent from the play, leaves only 4 of 40 gotos successful; and perception remains a bottleneck: the same pulses grasp in 85.2% of trials with the benchmark's object pose but 65.4% from the wrist camera.

Future work. Our next step is to deploy AquaWAM on a physical remotely operated vehicle (ROV) (Sivčev et al., 2018): the predictor reads only sensors that every ROV carries, so it can be adapted with unlabeled play data from the target hull, and the disturbance token, which already compensates for a weakened thruster (Table 2), offers a mechanism for online adaptation. We further plan to feed visual cues into the navigation-state estimator so that the cameras that supply the goals also bound the dead-reckoning drift that currently limits transport after DVL loss.

AI USE STATEMENT

Generative AI tools were used only to polish the writing of this paper, that is, grammar, wording and readability. They were not used to generate data, formulate claims, design or run experiments, implement methods, or analyze and interpret results. The authors reviewed every edit and take full responsibility for the content of the paper.

ETHICS STATEMENT

This work studies underwater robot control in simulation only, on the USIM benchmark and the open-source Stonefish simulator. It involves no physical vehicle, no human subjects and no personal data. Deploying a learned controller on a real vehicle would require safety validation beyond the scope of this paper.

REPRODUCIBILITY STATEMENT

The predictor, the score weights, the hand-tuned positioning law, the task list, play-data collection, baseline fine-tuning, the per-task results, and the failure cases are documented in Appendices A–G. Supplementary videos show representative episodes of every model (Appendix H). The code, the play data and the trained models will be released with the paper.

REFERENCES

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Ralf Bachmayer, Louis L. Whitcomb, and Mark A. Grosenbaugh. An accurate four-quadrant nonlinear dynamical model for marine thrusters: Theory and experimental validation. *IEEE Journal of Oceanic Engineering*, 25(1):146–159, 2000. doi: 10.1109/48.820747.

Kevin Black et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025. doi: 10.15607/RSS.2025.XXI.010.

Anthony Brohan et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023.

Jake Bruce et al. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 2024.

Ignacio Carlucho, Mariano De Paula, Sen Wang, Yvan Petillot, and Gerardo G. Acosta. Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning. *Robotics and Autonomous Systems*, 107:71–86, 2018. doi: 10.1016/j.robot.2018.05.016.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. WorldVLA: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014. doi: 10.3115/v1/D14-1179.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Patryk Cieślak. Stonefish: An advanced open-source simulation tool designed for marine robotics, with a ROS interface. In *OCEANS 2019 – Marseille*, pp. 1–6, 2019. doi: 10.1109/OCEANSE.2019.8867434.

Nadav Cohen, Zeev Yampolsky, and Itzik Klein. Set-Transformer BeamsNet for AUV velocity forecasting in complete DVL outage scenarios. In *IEEE Underwater Technology (UT)*, pp. 1–6, 2023. doi: 10.1109/UT49729.2023.10103453.

Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, 2011. doi: 10.1002/9781119994138.

Junwen Gu, Zhiheng Wu, Pengxuan Si, Shuang Qiu, Zhentao Zhang, Yukai Feng, Luoyang Sun, Laien Luo, Lianyi Yu, Jian Wang, and Zhengxing Wu. USIM and U0: A vision-language-action dataset and model for general underwater robots. *arXiv preprint arXiv:2510.07869*, 2025. Version 4, May 2026.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020. arXiv:1912.01603.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.16828.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Shahab Heshmati-Alamdari, George C. Karras, Panos Marantos, and Kostas J. Kyriakopoulos. A robust predictive control approach for underwater robotic vehicles. *IEEE Transactions on Control Systems Technology*, 28(6):2352–2363, 2020. doi: 10.1109/TCST.2019.2939248.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24328–24346. PMLR, 2025.

Nils G. Jerlov. *Marine Optics*, volume 14 of *Elsevier Oceanography Series*. Elsevier, Amsterdam, 2nd edition, 1976.

Leela S. Karumbunathan. NVIDIA Jetson AGX Orin Series: A giant leap forward for robotics and edge AI applications. Technical Brief TB_10749-001_v1.2, NVIDIA, July 2022. URL `https://www.nvidia.com/content/dam/en-zz/Solutions/gtcf21/jetson-orin/nvidia-jetson-agx-orin-technical-brief.pdf`.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2024.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2601.16163.

James C. Kinsey, Ryan M. Eustice, and Louis L. Whitcomb. A survey of underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC)*, 2006.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Robotics: Science and Systems (RSS)*, 2026. doi: 10.15607/RSS.2026.XXII.016.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Robotics: Science and Systems (RSS)*, 2025. doi: 10.15607/RSS.2025.XXI.074.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on Robot Learning (CoRL)*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1113–1132. PMLR, 2019.

NVIDIA et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025a. The N1.5 checkpoint used by U0 and in this paper is a later release of the same model line.

NVIDIA et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025b.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, 39(1):131–149, 2014. doi: 10.1109/JOE.2013.2278891.

Physical Intelligence et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011.

Chao Shen, Yang Shi, and Bradley Buckham. Trajectory tracking control of an autonomous underwater vehicle using Lyapunov-based model predictive control. *IEEE Transactions on Industrial Electronics*, 65(7):5796–5805, 2018. doi: 10.1109/TIE.2017.2779442.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, et al. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Satja Sivčev, Joseph Coleman, Edin Omerdić, Gerard Dooly, and Daniel Toal. Underwater manipulators: A review. *Ocean Engineering*, 163:431–450, 2018. doi: 10.1016/j.oceaneng.2018.06.018.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

Edoardo Topini, Alberto Topini, Matteo Franchi, Alessandro Bucci, Nicola Secciani, Alessandro Ridolfi, and Benedetto Allotta. LSTM-based dead reckoning navigation for autonomous underwater vehicles. In *Global Oceans 2020: Singapore – U.S. Gulf Coast*, pp. 1–7. IEEE, 2020. doi: 10.1109/IEEECONF38699.2020.9389379.

George E. Uhlenbeck and Leonard S. Ornstein. On the theory of the Brownian motion. *Physical Review*, 36(5):823–841, 1930. doi: 10.1103/PhysRev.36.823.

Bilal Wehbe, Marc Hildebrandt, and Frank Kirchner. Experimental evaluation of various machine learning regression methods for model identification of autonomous underwater vehicles. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4885–4890, 2017. doi: 10.1109/ICRA.2017.7989565.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.06114.

Seonghyeon Ye et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Dana R. Yoerger, John G. Cooke, and Jean-Jacques E. Slotine. The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design. *IEEE Journal of Oceanic Engineering*, 15(3):167–178, 1990. doi: 10.1109/48.107145.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.10274.

APPENDIX

A THE PREDICTOR

Table 4 lists the channels of the physical state, all read from the vehicle's own sensors. Specifically, the 19-dimensional vehicle block consists of body velocity and altitude from the DVL, angular rate and linear acceleration from the IMU, depth from the pressure sensor, and the last thruster command. Within reach of an object, the joint encoders and the wrist-camera head add 16 further dimensions, and the predictor predicts every channel. Each command consists of eight thruster values in $[-1, 1]$ and five joint targets, the first being the gripper (0 open, 0.015 closed). The history encoder E_ϕ embeds the last 16 steps (1.6 s) of states and commands with a linear layer and a Gaussian error linear unit (GELU) (Hendrycks & Gimpel, 2016), runs a single-layer gated recurrent unit (GRU) (Cho et al., 2014) with a hidden width of 384, and maps its last hidden state through layer normalization (Ba et al., 2016), dropout (Srivastava et al., 2014) and a linear layer to the 96-dimensional disturbance token; its parameters ϕ are part of θ and are trained jointly with f_{nom} and f_{res}. The nominal and residual networks are three-layer multilayer perceptrons (MLPs) of the same width, with GELU activations, layer normalization and a dropout of 0.1. Each call predicts $K = 5$ steps, and four chained calls cover the two-second horizon. The velocity estimate is the mean of three heads that read the history with the velocity channels masked, and their spread gives σ_t. Both scales of the state share this network, differing only in input and output widths. The perception heads, built on DINOv2-base, never enter the predictor, and the gripper closes when an outcome head on the predicted state reports a probability of at least 0.7.

Table 4: **Details of the 35-dimensional physical state used in AquaWAM.**

Channels	Dim.	Sensor
Vehicle (19)		
body velocity v	3	DVL
angular rate ω	3	IMU
linear acceleration	3	IMU
depth, altitude	2	pressure, DVL
last thruster command	8	own output
Arm and object (16)		
joint positions and velocities	10	encoders
object pose in the gripper frame	6	wrist camera

Algorithm 1 One decision of AquaWAM (every half second)

Input: state s_t from the DVL, IMU, pressure sensor and joint encoders; history $\mathcal{H}_t$; camera images; instruction
Output: the future $\hat{s}^\star_{t+1:t+H}$ that AquaWAM has chosen and the commands $a^\star_{t:t+K}$ that bring it about (Eq. 1)

goal g, target pose $\leftarrow$ perception heads(camera images, instruction) ▷ Section 3.4
if the DVL has lost bottom lock **then** ▷ Section 3.5
    $\hat{v}_t \leftarrow \text{estimate}(\mathcal{H}^v_t)$; write $\hat{v}_t$ into s_t; dead-reckon the pose; update the trust α_t (Eq. 6)
end if
scale $\leftarrow$ vehicle block while cruising, full state within 10 cm of the object ▷ Section 3.1
$\mathcal{A}_t \leftarrow$ thrust laws, single-axis moves and play snippets while cruising; thrust pulses within reach (Eq. 4)
for each candidate $\mathbf{a} \in \mathcal{A}_t$, in one batch **do**
    $\hat{s}(\mathbf{a}) \leftarrow \text{imagine}(s_t, \mathcal{H}_t, \mathbf{a})$ ▷ Eq. 2 chained four times, two seconds ahead
end for
while cruising: draw a second round around the 16 best and imagine it too
$\mathbf{a}^\star \leftarrow \arg\min_{\mathbf{a}} J(\hat{s}(\mathbf{a}))$ ▷ Eq. 3; J^{pulse} within reach
execute the first half second of $\mathbf{a}^\star$; close the gripper if $p(\text{grasp} \mid \hat{s}(\mathbf{a}^\star)) \geq 0.7$

B SCORES AND THE HAND-TUNED LAW

The weights of the score in Eq. 3 are $\lambda_g = 5$, $\lambda_u = 0.02$, $\lambda_s = 0.4$ and $\lambda_\omega = 0.5$. The speed cap $v_{\max}$ follows the current task: 0.25, 0.22 and 0.20 m/s for goto, scan and inspection, and 0.25 m/s when following the boat. Because the cap enters only as a soft penalty, the goal term outweighs it during tracking, which allows the vehicle to match the boat's 0.44 m/s. The velocity weight of the pulse score in Eq. 4 is $\lambda_v = 400$, and the noise floor of the trust statistic in Eq. 6 is $\sigma_0 = 0.05$, in units of the normalized velocity.

The pulse candidates of Eq. 4 combine the magnitudes $m \in \{0.12, 0.18, 0.25, 0.32, 0.40\}$, in units of the normalized body-thrust command that the mixer maps to the eight thruster values in $[-1, 1]$, with the durations $\tau \in \{0.1, 0.2, 0.3, 0.4\}$ s, i.e., at most $K - 1$ steps, because the last executed step is always a coast. This 5×4 grid (Section 4.2) yields 20 pulses per direction; the coarse grid there

is $\{0.18, 0.32\} \times \{0.2, 0.4\}$ s and the fine grid has nine magnitudes. An axis direction $\mathrm{sign}(\delta_i)\,\mathbf{b}_i$ enters only when $|\delta_i| > 4$ mm, and the braking direction $-v_t/\|v_t\|$, with v_t the body velocity measured by the DVL, only when $\|v_t\| > 1.5$ cm/s, so a decision weighs one to five directions, that is, between 21 and 101 candidates. There is no separate braking pulse: braking is a pulse along $-v_t$. The hold candidate issues no translational thrust over the whole horizon and keeps only the thrust that levels the attitude. In J^{pulse}, the offset e is taken in the world frame, with x and y horizontal as in the judge and z vertical, measured from a target point 1.2 cm above the object; $\|\hat{e}_{\mathrm{end}}\|_Q^2$ is the mean of $e^\top Q e$ over the last $\lfloor H/3 \rfloor = 6$ predicted steps (1.5 to 2.0 s), and $\hat{v}_{\mathrm{end}}$ is the predicted linear velocity at step $t + H$.

The hand-tuned law of Section 4.2 shares the task program and the closure window of AquaWAM, so that the two differ only in how the hull is positioned. It commands a body speed of 0.03 m/s, held for $\mathrm{distance}/(0.03 \times 0.1\ \mathrm{s})$ steps, and switches from approach to grasp at 8 cm. It closes the gripper once $|\Delta y| < 0.7$ cm and $|\Delta x| < 3$ cm, where Δx and Δy are the horizontal offsets from the gripper to the object rotated into the heading frame of the vehicle (x forward, y sideways), and once the distance is below 3.5 cm, the yaw error is below 0.2 rad and the hull is at rest.

C Tasks

Table 5 lists the twenty USIM tasks and the number of trials behind each entry of Table 1. AquaWAM follows USIM's task schema, which orders the sub-goals, and advances to the next waypoint once the hull is within 0.9 m of it for a goto, within 3.5 m and 0.8 rad for a scan node, and within 1.5 m and 0.4 rad for an inspection node; all of these thresholds lie inside the judge's own tolerances. The goal velocity points at the current waypoint and is capped as in Appendix B. When following the boat, the standoff is 3 m behind it along its heading and 0.5 m deeper. Within reach of an object, the arm holds a ready pose with joint targets $(0, 0.50, 0.50, 0.51, 0)$, the first of which keeps the gripper open, while the hull performs the positioning, and the pulse candidates take over within 10 cm. On the decision of the outcome head, the gripper is commanded to its closed position (joint target 0.015), the same value the experts use in the demonstrations, after which a transport task carries the object to the box head's goal.

Table 5: **The twenty USIM tasks** and trials per task.

Task	n	Task	n
Navigation			
goto charge station	40	scan ship modern	20
goto water tower	40	scan ship ancient	20
inspect pipeline pool	20	inspect pipeline sea	20
Grasping			
pick pipe0 shallow	40	pick pipe1 shallow	40
pick pipe0 factory	40	pick pipe1 factory	40
pick red shallow	40	pick redx shallow	40
pick red factory	40	pick redx factory	40
pick blue shallow	40	pick bluex shallow	40
pick blue factory	40	pick bluex factory	40
Transporting		*Tracking*	
transfer red shallow	40	follow boat	20

D Play Data

The demonstrations consist of repeated task executions, whereas the play carries no task or success label. The play consists of three recordings of the same vehicle, namely an Ornstein–Uhlenbeck exploration (Uhlenbeck & Ornstein, 1930), a pass through the scenes and a deployment log. Because its commands differ from those the experts issue, the play exposes the predictor to the dead band and to the glide beyond it, which the demonstrations alone do not reveal (Section 4.5).

E Baselines and DVL Dropout

Fine-tuning. Every model in Table 1 is fine-tuned on the same demonstrations. U0 trains its action head and commits to a 16-step chunk. OpenVLA uses sandwich fine-tuning, in which the vision encoders, the projector and the output layer are trained while the language model stays frozen, for 22k steps at batch size 64, with the actions discretized into 256 bins. GR00T N1.5 receives full action-head fine-tuning for 120k steps at batch size 16 and, like U0, commits to a 16-step chunk. $\pi_{0.5}$ trains the vision tower and the action expert with the language model frozen, on absolute actions, for 22k steps at batch size 64, and, like OpenVLA, acts at every step. X-VLA adapts its released

Table 6: **Per-task success of AquaWAM with full sensing and under DVL dropout.** Bold marks the higher of the two.

Task	Full	Dropout	Task	Full	Dropout
goto charge station	40/40	40/40	scan ship modern	19/20	**20/20**
goto water tower	**36/40**	33/40	scan ship ancient	16/20	**17/20**
inspect pipeline pool	20/20	20/20	inspect pipeline sea	20/20	20/20
follow boat	**20/20**	16/20	transfer red shallow	**23/40**	3/40
pick pipe0 shallow	**33/40**	21/40	pick pipe1 shallow	**33/40**	25/40
pick pipe0 factory	**28/40**	21/40	pick pipe1 factory	**26/40**	25/40
pick red shallow	**24/40**	21/40	pick redx shallow	24/40	**25/40**
pick red factory	**22/40**	17/40	pick redx factory	20/40	**21/40**
pick blue shallow	**34/40**	27/40	pick bluex shallow	**22/40**	18/40
pick blue factory	**28/40**	21/40	pick bluex factory	20/40	20/40

Table 7: **The estimator of AquaWAM as a takeover layer for U0 when DVL is not available.** Bold marks the best in each column.

Model	Navigation	Grasping	Transporting	Tracking	Overall SR
U0	73/160	202/480	0/40	1/20	39.4% (276/700)
U0 + takeover	**155/160**	181/480	0/40	12/20	49.7% (348/700)
AquaWAM	150/160	**262/480**	**3/40**	**16/20**	**61.6% (431/700)**

base model for 120k steps at batch size 16 and renews its chunk every 16 steps, as does SmolVLA, which is fine-tuned for 65k steps at batch size 32.

DVL dropout. Table 6 compares the per-task success of AquaWAM under full sensing with that under the dropout of Section 4.3. The goto, scan and inspection tasks are almost unaffected, and tracking drops from 20 to 16 of 20 trials. In contrast, grasping loses at most 12 of 40 successes per object, and transport falls from 23 to 3 of 40, since the object must still be carried after the loss.

Takeover layer. Table 7 breaks down by family the takeover experiment of Section 4.3, in which, once the DVL reports loss, U0 receives the velocity estimated by AquaWAM in place of the DVL measurement. Navigation rises from 73 to 155 of 160 and tracking from 1 to 12 of 20, whereas grasping falls from 202 to 181 of 480 and transport stays at 0 of 40.

F USIM-HARD

Tables 8 and 9 break the protocols of Table 2 down by task. Every perturbation is applied before the scene loads and is seeded by task and trial, so that every model faces the same perturbation in the same trial. The perturbations are defined as follows. *Heading*: the start yaw is rotated by 90 to 180 degrees, whereas the demonstrations always start facing the goal. *Tight endpoint*: the goto judge's ball shrinks from 1.0 to 0.5 m. *Thruster at 50%*: one horizontal thruster delivers half of its commanded thrust without announcement; the efficiency of thruster 3 falls inside the range covered by the play data, and that of thruster 1 outside it. *Thruster 1 at 0%*: that thruster delivers none of its commanded thrust. *Turbid water*: the Stonefish water-type parameter, which follows the Jerlov classification (Jerlov, 1976) from clear oceanic water (0) to turbid coastal water (1), is raised from 0.15 to 0.50. The stricter judges, in contrast, perturb nothing: they re-score the recorded trajectories of Table 1 and, in the two DVL-loss rows, those of the dropout runs of Section 4.3. For a goto, every intermediate waypoint must be passed within the stated radius; for an inspection, every node must lie within a 1 m ball and 90% of the nodes must be covered.

G FAILURE CASES

Three results in the main text mark where the margin of AquaWAM shrinks or where both models fail. We collect them here together with their causes.

Table 8: **Perturbed worlds of Table 2, per task.** 20 trials per cell.

Task	U0	AquaWAM
Heading rotated 90–180°		
goto charge station	10/20	7/20
goto water tower	16/20	16/20
inspect pipeline pool	12/20	20/20
scan ship modern	14/20	20/20
Goto endpoint within 0.5 m		
goto charge station	18/20	20/20
goto water tower	8/20	15/20
Thruster 3 at 50%		
goto charge station	3/20	18/20
goto water tower	0/20	12/20
Thruster 1 at 50%		
goto charge station	4/20	13/20
goto water tower	0/20	8/20
Thruster 1 at 0%		
goto charge station	1/20	4/20
goto water tower	0/20	0/20
Turbid water, Jerlov 0.50		
goto water tower	1/20	9/20

Table 9: **Stricter judges of Table 2, per task.** Gotos of 40 trials, inspections of 20.

Task	U0	AquaWAM
Goto, all waypoints within 1.0 m		
goto charge station	37/40	39/40
goto water tower	16/40	22/40
Goto, all waypoints within 0.5 m		
goto charge station	34/40	32/40
goto water tower	10/40	12/40
Goto, 0.5 m waypoints, DVL loss		
goto charge station	27/40	36/40
goto water tower	1/40	9/40
Goto, all waypoints within 0.25 m		
goto charge station	20/40	18/40
goto water tower	5/40	6/40
Inspect, 1 m ball, 90% coverage		
inspect pipeline pool	16/20	19/20
inspect pipeline sea	20/20	20/20
Inspect, same judge, DVL loss		
inspect pipeline pool	4/20	20/20
inspect pipeline sea	3/20	19/20

First, transport after DVL loss succeeds in only 3 of 40 trials, against 0 of 40 for U0. The object must still be carried 2.9 m after the loss, so the estimate, rather than a measurement, has to hold the velocity over the whole carry.

Second, with thruster 1 delivering no thrust, a fault that the play never contains, the two goto tasks succeed in only 4 of 40 trials, against 1 of 40 for U0. A weakened thruster is largely absorbed by the disturbance token (21 of 40 at half thrust), whereas a missing one is not.

Finally, turbid water with a Jerlov water-type parameter of 0.50 degrades the cameras that supply the goal, and AquaWAM reaches the water tower in only 9 of 20 trials, against 1 of 20 for U0.

H Qualitative Comparisons

Figures 8 and 9 show selected episodes. In Figure 8 and Figure 9a, every model starts from identical initial conditions. On pick blue shallow, AquaWAM closes the gripper on the object at 19 s, whereas U0 and GR00T N1.5 stay near it without a successful grasp until the 180 s timeout; on pick blue factory, AquaWAM succeeds at 20 s and U0 at 57 s, whereas GR00T N1.5 times out. On inspect pipeline pool, AquaWAM, $\pi_{0.5}$ and U0 complete the route in 53, 95 and 122 s, and on scan ship modern, GR00T N1.5, AquaWAM and U0 complete the scan in 61, 63 and 67 s and X-VLA in 173 s. Across these four episodes, $\pi_{0.5}$ and X-VLA succeed only on the inspection and the scan, respectively, and SmolVLA and OpenVLA on none. Figure 9b–d covers navigation to a landmark, tracking and transport, for which no shared episode was recorded; each row shows the episode of AquaWAM when it was also recorded for that model and otherwise the recorded episode of median duration, so these rows do not share an initial condition and their durations are not directly comparable. The supplementary video plays the shared episodes side by side, with the simulated time and the playback speed shown. All clips are selected examples, and the success rates are those of Tables 1 and 6.

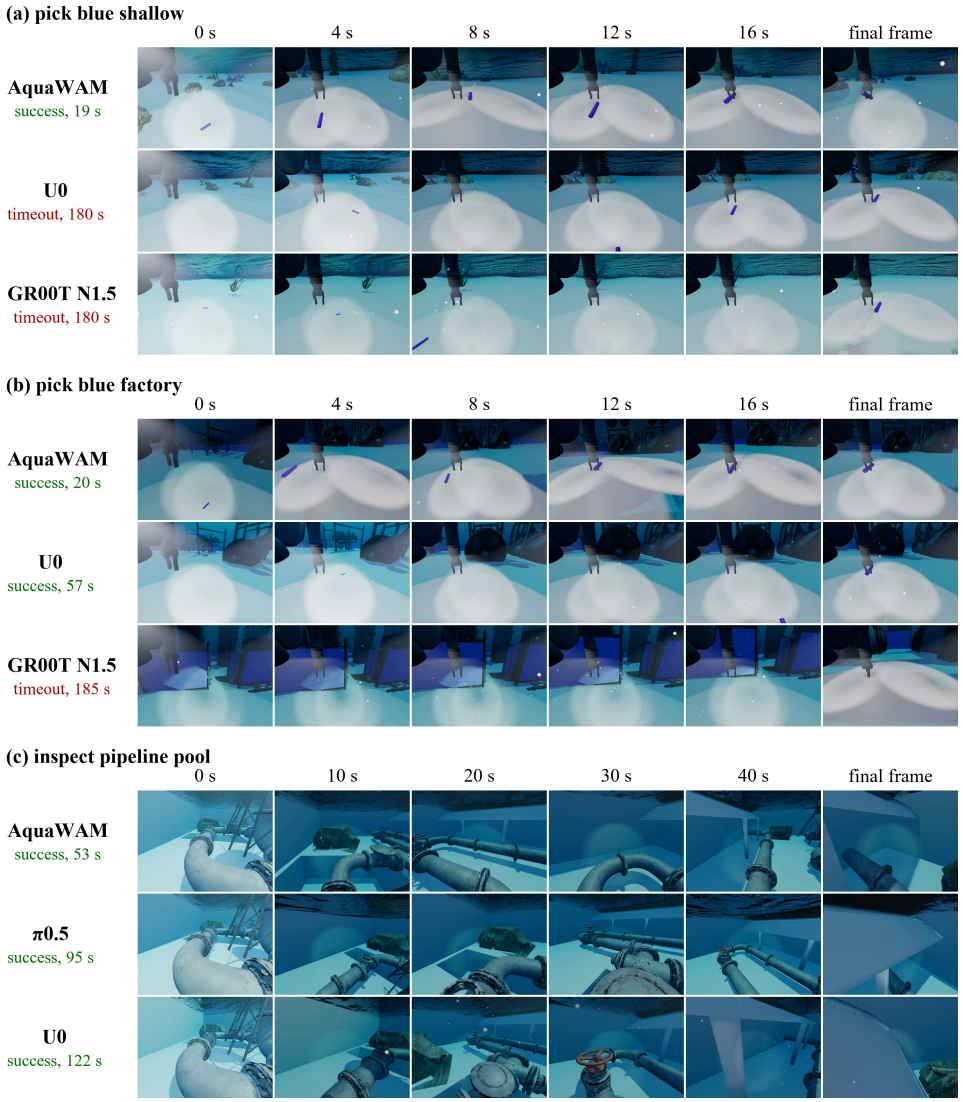


Figure 8: **Shared episodes: two grasps and an inspection.** Each panel follows one episode from identical initial conditions; the last column is the final frame.

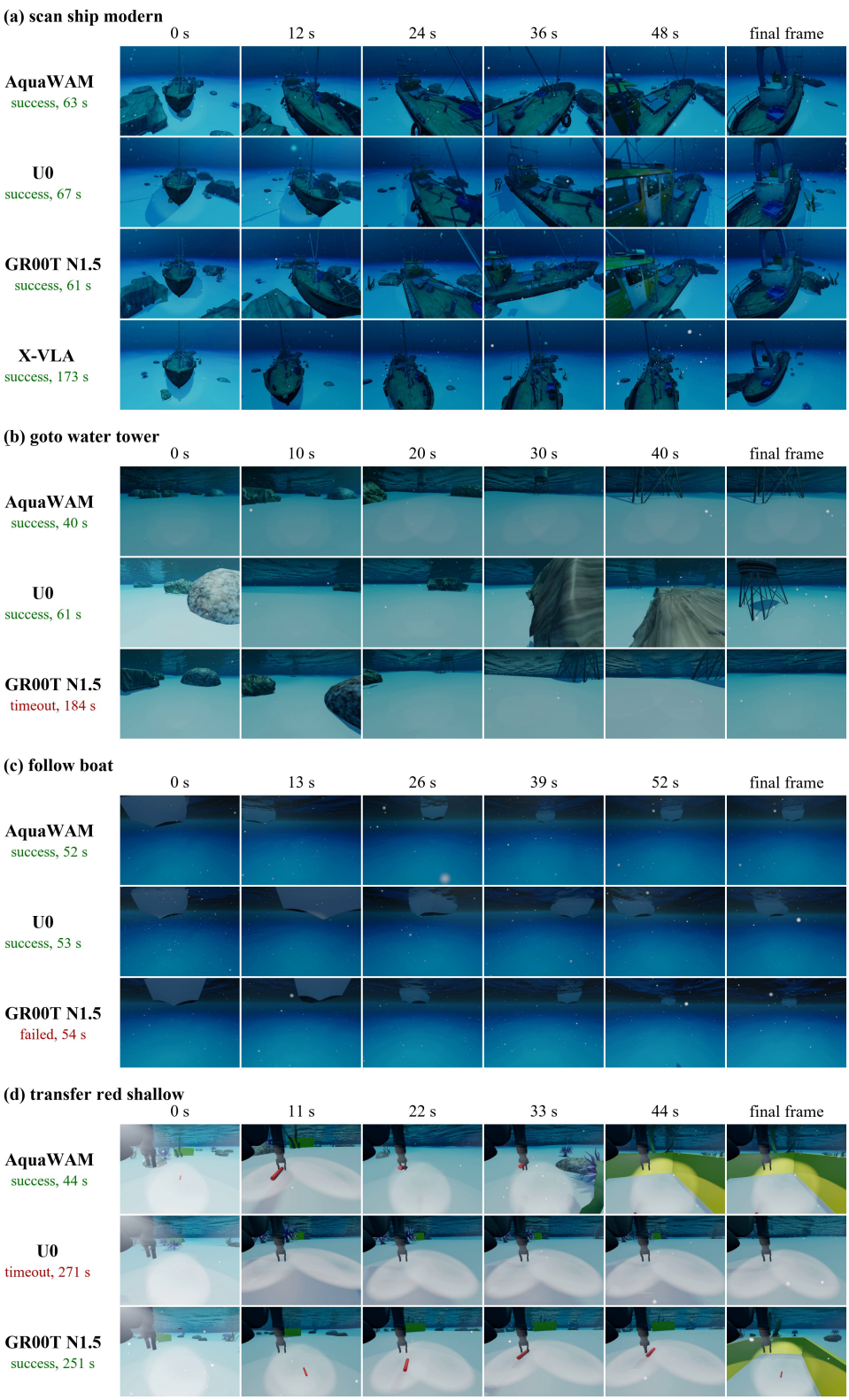


Figure 9: **Scanning, navigation, tracking and transport.** (a) Shared episode; in (b)–(d), columns mark quarters of the AquaWAM episode. The last column is the final frame.