BRINE: MATCHED-STATE COUNTERFACTUAL SUPERVISION FOR RISK-AWARE COGNITIVE ROUTING IN UNDERWATER ROBOTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon underwater control requires cognitive computation to be allocated by current risk and capability. Reactive control efficiently handles routine conditions, while degraded sensing and disturbances benefit from adaptive deliberation; learning this allocation requires evidence about the consequences of alternative cognitive branches. We introduce BRINE, a multimodal system that addresses this coupled decision, learning, and execution problem. From observed robot state, RGB images, point clouds, sonar, task commands, and episodic memory, BRINE determines when deliberation is needed and coordinates three role-specialized components: a state refiner, a safety verifier, and a capability planner. A safety-constrained action projection governs execution. To train the router, we construct matched-state counterfactual supervision by evaluating alternative expert combinations from the same simulator state and comparing their task, safety, and computation outcomes. We further formulate risk-shielded contextual-bandit refinement of the router and introduce role-aware mixed-precision execution that retains precision-sensitive semantic and safety computations in FP32. On 72 paired underwater tasks, the standard-precision system succeeds in 62 cases, compared with 48 for reactive control alone and 52 for always-on deliberation. Counterfactual supervision improves held-out expected return by 0.255 (95% confidence interval: 0.139–0.377) over matched risk and teacher supervision. In separate paired 24-task evaluations, contextual-bandit refinement reduces actual expert-forward duty from 0.609 to 0.482 (paired difference -0.127; 95% confidence interval: -0.180 to -0.074; Holm-adjusted $p < 0.001$), completes 24/24 tasks with zero collisions, and matches the reference route on all 287 high-risk decisions; mixed precision reduces peak temporary allocated memory by 7.13% and matches FP32 with 24/24 successes. These findings support counterfactual learning for selective, safety-conscious deliberation and mixed-precision execution with preserved behavioral semantics.

1 INTRODUCTION

Autonomous underwater robots must act continuously while observations and task demands change. Limited visibility, uncertain geometry, and environmental disturbances make perception and control difficult even in local navigation (Lin et al., 2024). Longer missions benefit from deliberative decisions that reinterpret ambiguous scenes, verify maneuver safety, and revise infeasible subgoals. Invoking every deliberative component at every decision, however, adds computation and can introduce unnecessary interventions. The central question is therefore which cognitive operations a robot needs *now*, given its current risk, available capabilities, and task progress. The resulting problem is to allocate heterogeneous cognitive roles online while capability and safety rules retain execution authority.

Conditional computation offers a way to allocate model capacity selectively (Shazeer et al., 2017). For embodied control, the selection problem has additional structure. Each component has a specialized decision role: state refinement updates the evidence used by later decisions, safety verification constrains an action, and capability planning revises what the robot should attempt next. Their combinations respect these roles, and an independent safety mechanism retains final execution au-

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.

thority. We introduce **BRINE**, a multimodal architecture that couples a reactive controller with role-specialized deliberation. It fuses robot state, RGB images, point clouds, sonar, task commands, and episodic memory; a risk- and capability-conditioned router then selects among admissible combinations of a state refiner, safety verifier, and capability planner. The state refiner provides an early diagnostic update, whereas safety verification and capability planning are invoked according to the selected route. A safety-constrained projection retains authority over the executed action, including when learned routing proposes a cheaper mode.

Learning this router requires simultaneous evidence about alternative cognitive choices. We obtain this evidence through *six-branch counterfactual cognitive supervision*: from the same restored state and under the same exogenous disturbance schedule, we evaluate six admissible cognitive branches and score their subsequent task, safety, and computation outcomes. The resulting branch values supervise the router's choice of whether to deliberate and which roles to engage. Inspired by world models' use of prospective outcomes, the six alternative futures expose the consequences of each cognitive route (Hafner et al., 2025). We additionally use the complete branch-value vector as full feedback for contextual-bandit refinement. The learned proposal is optimized on low-risk states, while a risk shield retains calibrated high-risk routing. This couples task, safety, and computation outcomes in the allocation objective while preserving explicit high-risk authority.

The active experts also contribute to memory cost. Mixed-precision neural computation can reduce memory use (Micikevicius et al., 2018), and BRINE assigns precision according to each operation's semantic role. Suitable expert neural computations use lower precision, while the router, memory readout, semantic thresholds, safety and capability decisions, and action projection remain in FP32. A precision assignment is accepted when it preserves the meaning of expert outputs and the resulting system decisions.

We evaluate how cognitive allocation is learned, refined, and executed. The standard-precision system succeeds on 62 of 72 tasks, compared with 48 for reactive control and 52 for always-on deliberation. Counterfactual supervision improves held-out expected return by 0.255 (95% confidence interval: 0.139–0.377). In downstream paired evaluations, contextual-bandit refinement lowers actual expert-forward duty by 20.9%, from 0.609 to 0.482, while completing every task with zero collisions and preserving all 287 high-risk routes; mixed precision reduces peak temporary allocated memory by 7.13%, with both configurations completing every task.

Our contributions address cognitive allocation at the decision, learning, and execution levels:

- A risk- and capability-conditioned cognitive router that selects legal combinations of role-specialized experts under explicit safety authority, together with a full-feedback contextual-bandit refinement that reduces expert computation while preserving shielded high-risk routing.
- Six-branch counterfactual supervision that compares task, safety, and computation consequences from the same state and provides a learning signal for selective deliberation.
- Role-aware mixed-precision execution that preserves FP32 semantic and safety authority, with measured temporary-memory reduction and paired closed-loop behavioral preservation.

2 Related Work

Multimodal perception and embodied control. Underwater navigation couples perception and action under visibility loss, ambiguous visual structure, and modality-dependent sensing failures. Existing studies address these challenges through learned visual navigation and complementary sensing: Lin et al. (2024) learn vision-based navigation for underwater surveys, while Coffelt et al. (2024) combine camera and sonar measurements for odometry and reconstruction. These efforts establish a multimodal foundation for underwater control, where different sensing modalities contribute complementary evidence about the environment. At the policy level, recent vision-language-action systems investigate compact deployment, long-horizon memory, and adaptive transitions between acting and explicit reasoning (Shukor et al., 2025; Shi et al., 2026; Lin et al., 2026). Together, these directions motivate controllers that can both integrate heterogeneous observations and adapt their computation to the current decision. Building on this foundation, BRINE studies when a

reactive controller should receive additional support from state refinement, safety verification, or capability planning, according to the evolving risk and capability of the embodied system.

Adaptive computation under safety and resource constraints. Selective computation can operate at the level of both module activation and numerical execution. Sparse mixture-of-experts models activate selected subnetworks for each input (Shazeer et al., 2017), establishing a general mechanism for allocating computation conditionally. In embodied systems, this allocation must also account for what the robot can execute and which decisions are admissible. Capability grounding connects language-guided plans to feasible skills (Ichter et al., 2023), while shielding restricts unsafe choices of learned policies (Alshiekh et al., 2018). Execution efficiency introduces a further constraint: mixed-precision computation retains higher precision where numerically necessary (Micikevicius et al., 2018), hardware-aware methods allocate precision according to resource and accuracy requirements (Wang et al., 2019), and recent VLA systems incorporate action sensitivity into quantization and dynamic token pruning (Xu et al., 2026; Pei et al., 2026). These studies motivate a joint view of computation selection, execution feasibility, and numerical cost. BRINE applies this view to experts with distinct decision roles: refining the state estimate, evaluating safety, and revising an attainable subgoal. Cognitive branch selection is constrained by admissible role combinations, while action projection constrains execution. Role-dependent precision further reduces computational cost while retaining FP32 for routing, semantic decisions, and safety-sensitive operations.

Counterfactual supervision for cognitive routing. Learning when to allocate additional computation requires evidence about the consequences of alternative choices. Counterfactual risk minimization addresses learning from logged bandit feedback through propensity-weighted risk estimation (Swaminathan & Joachims, 2015). Model-based reinforcement learning uses transition models to support policy improvement, from Dyna (Sutton, 1990) to Dreamer (Hafner et al., 2025). Recent work further evaluates world models through their utility in closed-loop embodied tasks and uses imagined trajectories for VLA policy optimization (Zhang et al., 2026; Zhu et al., 2026). These approaches connect decisions to prospective outcomes and motivate supervision based on their downstream consequences. In BRINE, the alternatives being evaluated are cognitive routes that can modify the information, safety assessment, or subgoal used by the controller. BRINE restores a shared state and evaluates alternative expert combinations under matched disturbances, obtaining paired measurements of task progress, safety, and computational cost. The resulting branch-wise outcomes provide full-feedback supervision for contextual routing, linking the allocation of cognitive operations to their consequences in closed-loop control.

3 METHOD

Figure 1 summarizes BRINE's online causal execution and offline counterfactual learning paths. The online path separates the reactive proposal, risk-shielded cognitive routing, role-specialized experts, and FP32 execution authority; the offline path restores a common state to compare all six legal cognitive modes.

3.1 MULTIMODAL STATE REPRESENTATION AND CONTROL INTERFACE

At decision time t, the robot receives an observed state x_t, RGB image I_t, point cloud P_t, sonar observation S_t, and task command c. The task command specifies the goal through a dedicated task-control interface. Each decision uses the causally available observations. A multimodal state constructor produces a representation h_t and modality reliability and conflict diagnostics. An observed local navigation context supplies goal-relative and geometric information to the reactive controller. The current capability k_t belongs to a task-specific capability graph whose edges define legal successor capabilities and recovery options.

The reactive controller proposes $a_t^{\text{fast}} = \pi_{\text{fast}}(h_t, g_t, \mathcal{H}_t)$, where g_t is the observed local goal context and $\mathcal{H}_t$ is causal state-action history. The router also receives risk diagnostics r_t, action uncertainty u_t, action inconsistency d_t, failure evidence f_t, a risk trend q_t, the capability encoding, and an episodic memory readout e_t. We denote its context by

$$z_t = \big[h_t, r_t, u_t, d_t, f_t, q_t, \text{emb}(k_t), e_t\big]. \tag{1}$$

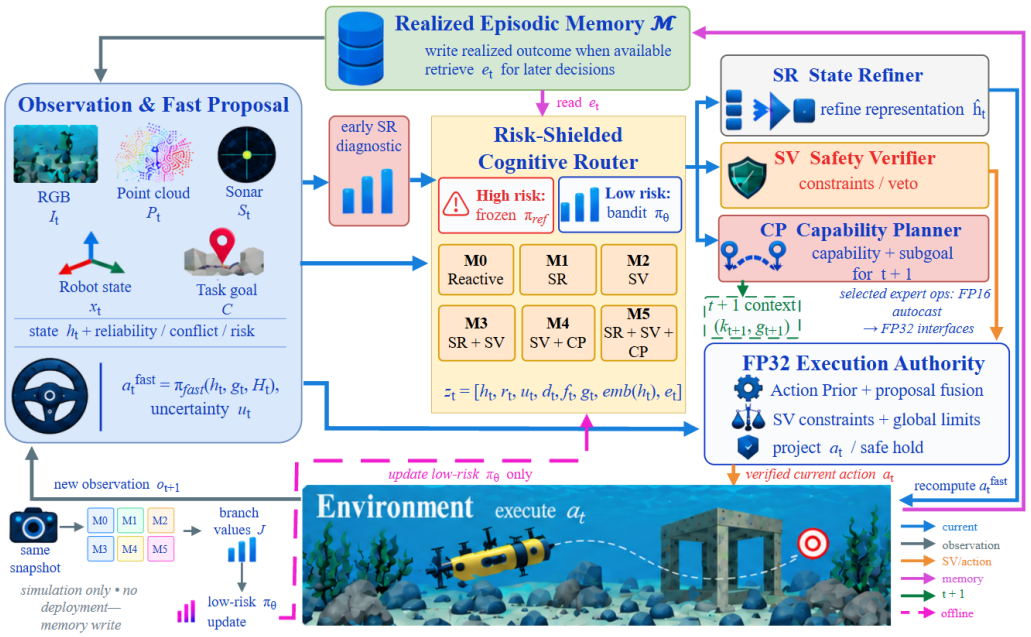

Figure 1: BRINE overview. The online flow routes the fast proposal through selected cognitive roles to FP32 execution authority; safety constraints govern the current action, while confirmed CP revisions enter the next-decision context. Observation and realized-memory loops close the interaction cycle. Dashed magenta paths show matched-state supervision and low-risk bandit updates, while shielded decisions retain calibrated high-risk routing.

Memory summarizes similar *realized* past decisions within the relevant capability and is updated after their outcomes become available. Deployment memory contains realized experience, while counterfactual branch outcomes provide router supervision. Explicit availability indicators distinguish observed measurements from missing evidence.

3.2 Risk- and Capability-Conditioned Role Routing

BRINE separates three interventions. The state refiner (SR) updates the multimodal representation and diagnoses unreliable sensing. The safety verifier (SV) evaluates an action proposal and supplies safety constraints or a veto. The capability planner (CP) proposes a legal capability or subgoal revision; a confirmed revision enters the control context at the next decision. SR's diagnostic forward pass occurs before routing, and an SR-containing mode determines whether its refined representation is used for the current action. Measured expert-forward duty includes this diagnostic computation.

Table 1: Legal routes. SR diagnostics are computed before routing; the table lists role outputs used in the selected branch.

Route	Role outputs used in the current branch
M_0	Reactive control only
M_1	SR
M_2	SV
M_3	SR, SV
M_4	SV, CP
M_5	SR, SV, CP

The router chooses from the six structurally legal combinations in Table 1. CP is paired with SV so that a revised goal remains subject to safety verification. Let $p_t^{\mathrm{up}} = \sigma(v(z_t))$ be the upgrade probability and $p_t^{\mathrm{role}} = \mathrm{softmax}(w(z_t)) \in \mathbb{R}^5$ be the conditional distribution over upgraded modes. The six-mode policy is

$$\pi_\theta(M_0 \mid z_t) = 1 - p_t^{\mathrm{up}}, \qquad \pi_\theta(M_j \mid z_t) = p_t^{\mathrm{up}} p_{t,j}^{\mathrm{role}}, \quad j = 1, \ldots, 5. \tag{2}$$

Risk, capability, progress, and memory evidence influence both heads. Emergency evidence and a local reactive-action guard can override a nominally cheap route. During bandit refinement, shield-active decisions retain the calibrated router's route. The learned policy thus proposes cognitive allocation, while safety and capability authority remain outside that proposal.

The reactive proposal is recomputed from the refined representation when SR is selected and is combined with the calibrated action-prior correction to form an action proposal $\tilde{a}_t$. This correction may condition on an already-visible CP context from an earlier decision; however, the current CP output cannot feed back into the same decision; a newly confirmed CP capability or subgoal revision enters the next decision's control context. The executed action is projected onto the global action limits and, when SV is active, its predicted feasible set:

$$a_t = \operatorname*{argmin}_{a \in \mathcal{U}_t} \|a - \tilde{a}_t\|_2^2, \qquad \mathcal{U}_t \subseteq \mathcal{U}_{\text{global}}. \tag{3}$$

Whenever the feasible set is nonempty, the projected action lies in $\mathcal{U}_{\text{global}}$; otherwise the robot executes a safe hold. An accepted CP revision follows an edge of the capability graph and enters the control context at the next decision. These execution rules apply to every selected route.

3.3 Matched-state counterfactual cognitive supervision

For each training snapshot i, we restore the same simulator state, sensor history, controller state, and exogenous disturbance schedule before executing each $m \in \{M_0, \ldots, M_5\}$. These six continuations provide a shared-state comparison of the consequences of alternative cognitive routes and serve as the router's offline supervision. Each continuation produces capability completion $C_i(m)$, clipped normalized progress $P_i(m) \in [-1, 1]$, collision indicator $U_i(m)$, and nominal role cost $c(m) = |\mathcal{R}(m)|/3$, where $\mathcal{R}(m)$ is the role set in Table 1. The router-local branch value is

$$J_i(m) = C_i(m) + \alpha P_i(m) - \beta U_i(m) - \gamma c(m). \tag{4}$$

We use a continuation of up to 20 decisions with $\alpha = 0.1$, $\beta = 10$, and $\gamma = 0.01$. An infeasible constraint set yields a dedicated safe-abort outcome. The continuation value evaluates local routing consequences, and the role cost provides a target-side complexity proxy. Actual expert-forward duty separately measures SR's pre-routing diagnostic and all executed expert calls.

To form a target, we first consider collision-free branches when any exist. Among branches within a value tolerance of the maximum, we prefer fewer roles. This yields a preferred route $m_i^\star$ and an upgrade target that requires either a safety rescue or sufficient improvement over M_0. Binary upgrade supervision, conditional route supervision, and benefit prediction train the router alongside risk and memory regularization. The six matched continuations provide counterfactual supervision for each common state.

3.4 Risk-shielded contextual-bandit refinement

The complete branch-value vector supports full-feedback contextual-bandit refinement. Let π_{ref} denote the risk-and-teacher-supervised router, let $\widetilde{J}_i(m)$ be the normalized value of branch m, and partition the fitting contexts into low-risk and high-risk sets $\mathcal{L}$ and $\mathcal{H}$ using the guard predicate. Optimization updates the proposal router:

$$\begin{aligned}\mathcal{L}_{\text{CB}}(\theta) = & -\mathbb{E}_{i \in \mathcal{L}} \sum_m \pi_\theta(m \mid z_i) \widetilde{J}_i(m) - \beta_H \mathbb{E}_{i \in \mathcal{L}} H(\pi_\theta(\cdot \mid z_i)) \\ & + \lambda_H \mathbb{E}_{i \in \mathcal{H}} D_{\text{KL}}(\pi_{\text{ref}}(\cdot \mid z_i) \| \pi_\theta(\cdot \mid z_i)) \,. \end{aligned} \tag{5}$$

The first term directly optimizes the expected six-branch return where refinement is permitted; entropy regularization maintains exploration during fitting, and the high-risk KL term anchors the proposal to the reference distribution. The state constructor, reactive controller, experts, memory readout, and action-fusion modules remain fixed. At execution, the effective policy is π_{ref} on guarded states and π_θ otherwise. Capability legality, emergency handling, and safety-constrained action projection govern every resulting action.

The refinement is specialized to BRINE's cognitive-allocation structure. Its action space is the six legal SR–SV–CP role combinations rather than low-level robot commands, and its context joins the router's multimodal state with risk, uncertainty, inconsistency, failure, capability, progress, trend, and realized-memory evidence. Its return is the system-specific branch value in Eq. 4, combining capability completion, closed-loop progress, collision cost, and cognitive-role cost. Matched restoration supplies the return of every legal route under the same state and disturbance schedule, enabling a full-feedback update over the complete cognitive action set. The high-risk predicate, capability graph, emergency handling, and action projection then integrate the learned proposal with

BRINE's existing execution authorities. This adaptation focuses reinforcement learning on the system decision that BRINE exposes, which cognitive roles to invoke, while retaining the calibrated experts and control stack.

3.5 Role-aware mixed-precision experts

Precision is assigned to neural operations according to expert roles. Selected SR image and sonar encoder operations, SV risk-prediction operations, and CP subgoal-prediction operations use lower-precision arithmetic. The router, memory readout, semantic thresholds, feasibility geometry, capability legality, revision lifecycle, and final action projection remain in FP32. Expert outputs return to FP32 before these decisions. A non-finite mixed-precision output triggers a full FP32 expert evaluation.

The precision configuration is selected on shared expert calibration inputs by requiring finite outputs, agreement of discrete routing and role decisions with FP32, bounded action deviation, and lower measured peak temporary allocated memory. It is then evaluated on paired closed-loop tasks with unchanged model weights and route semantics. Calibration and closed-loop evaluation measure resource use and behavioral preservation.

4 Experiments

4.1 Evaluation design

We evaluate closed-loop underwater control in three simulated scene families: an open tank, a pipe corridor, and an industrial frame. The primary task set contains 72 predefined parent tasks, with 24 from each scene family; 48 are comparatively simple and 24 require more complex recovery or coordination. Paired variants start from the same parent task and are scored by task completion, collision, and normalized goal progress. We additionally measure *actual expert-forward duty*, defined as the total number of SR, SV, and CP forwards divided by three times the number of decisions. This measure includes SR's pre-routing computation and can exceed one when a role is called multiple times within a decision.

Binary task outcomes use paired exact McNemar tests; continuous paired outcomes use bootstrap intervals or permutation tests, with Holm correction for prespecified comparison families. Routing is examined through system-level selection, targeted role challenges, six-branch counterfactual supervision, and paired risk-shielded bandit evaluation. The 72-task system and external-policy comparisons use FP32. Router refinement and mixed-precision execution are each confirmed on 24 paired simple tasks under their corresponding calibration protocols. Simulation time is paused during policy inference, and latency is measured separately in wall-clock time. The reported task outcomes therefore do not establish closed-loop performance under inference-induced delay.

Unless otherwise stated, the reactive controller executes M_0, always-on deliberation activates all three roles as M_5, and BRINE selects an effective route adaptively from M_0–M_5. Experiments that isolate supervision, router refinement, or numerical precision retain the route space and execution rules defined in Table 1.

Figure 2 provides a schematic visualization of a successful navigation episode. The six views illustrate the robot's progression along a curved trajectory around the scene obstacles toward the goal. The blue segment indicates the trajectory executed up to each view, while the pale curve shows the complete recorded path for retrospective visualization. This example illustrates the spatial progression of the episode and complements the quantitative closed-loop comparisons below.

4.2 Selective deliberation and external policies

Table 2 evaluates the complete standard-precision BRINE routing system against its reactive controller alone and unconditional activation of all deliberative roles. BRINE completes 62/72 tasks, compared with 48/72 and 52/72, respectively. Relative to reactive control, collisions fall from 12 to one. The paired success differences remain significant after Holm correction against reactive control ($p = 0.0020$) and always-on deliberation ($p = 0.0234$); the collision difference against reactive control also remains significant ($p = 0.0127$). Normalized goal progress improves by 0.196 against

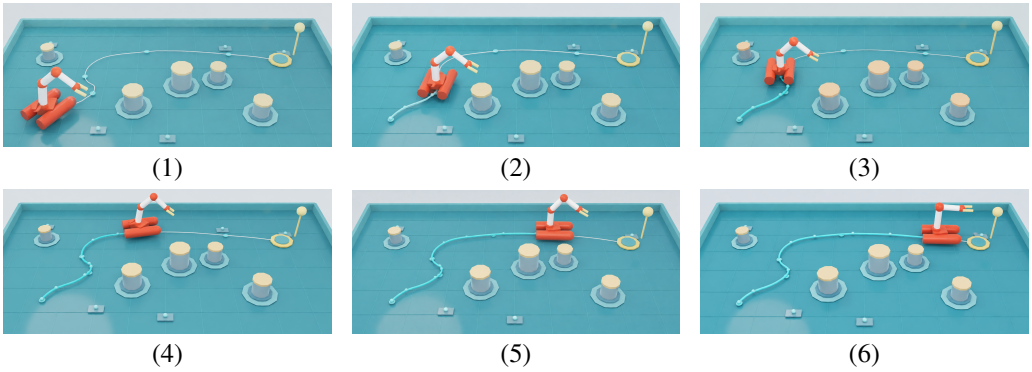

Figure 2: Six successive views of a successful navigation episode, ordered from left to right and top to bottom. Blue denotes the executed trajectory up to each view; the pale path shows the complete recorded trajectory. The robot and scene objects are schematic visualizations.

reactive control (Holm-adjusted $p = 0.0017$), and success-weighted path efficiency (SPL) increases from 0.667 to 0.829. BRINE's measured expert-forward duty is 0.611, compared with 1.010 for the always-on system, corresponding to a 39.5% reduction by this metric. Mean decision latency is 287.9 versus 328.9 ms for always-on deliberation, a 12.5% reduction in this deployment comparison.

Table 2: Paired 72-task system comparison. Progress is normalized goal progress; SPL is full-population success-weighted path efficiency. Duty counts SR, SV, and CP forwards per decision, normalized by three.

Controller	Success↑	Collisions↓	Progress↑	SPL↑	Duty↓	Latency↓ (ms)
Reactive controller	48/72	12/72	0.348	0.667	0.333	202.9
Always-on deliberation	52/72	0/72	0.540	0.652	1.010	328.9
BRINE (FP32)	**62/72**	1/72	**0.544**	**0.829**	0.611	287.9

Table 3: Task-difficulty breakdown of the paired 72-task system comparison. Each row contains 48 simple and 24 complex tasks. Routing configurations are identical to Table 2.

	Simple tasks		Complex tasks			
Controller	Success↑	SPL↑	Success↑	Collision↓	Progress↑	SPL↑
Reactive controller	48/48	1.000	0/24	12/24	0.027	0.000
Always-on deliberation	40/48	0.729	12/24	0/24	0.552	0.497
BRINE (FP32)	**48/48**	0.958	**14/24**	1/24	**0.610**	**0.571**

The effect is concentrated where recovery and role coordination are needed. Table 3 separates simple and complex tasks. On the 24 complex tasks, BRINE completes 14, reactive control completes 0, and always-on deliberation completes 12; its SPL is 0.571, versus 0 and 0.497. On the 48 simple tasks, BRINE and reactive control both complete all tasks. Appendix Table 15 shows that BRINE's success advantage over reactive control occurs in all three scene families.

Closed-loop routing diagnostics further characterize when deliberation is selected. A beneficial opportunity is a paired task on which BRINE improves completion, collision, or progress over reactive control; selected deliberation identifies these opportunities with precision 0.611 and recall 1.000. Deliberation is also selected on 44.3% of pooled simple-task decisions and captures every beneficial opportunity on the complex tasks. The denominators and complete diagnostic values are given in Appendix Table 14; they differ from the 37-state classification results in Table 5.

We also compare with generalist vision-language-action policies: Octo (Ghosh et al., 2024), OpenVLA (Kim et al., 2025), SpatialVLA (Qu et al., 2025), π_0 and $\pi_{0.5}$ (Black et al., 2025b;a), SmolVLA (Shukor et al., 2025), and X-VLA (Zheng et al., 2026). The uniform comparison freezes every upstream VLA and fits the same ridge-regression recipe from 2,800 causal training anchors to the

canonical two-dimensional command. All policies execute the same 72 task instances from matched physical initial states, use bounded body-frame forward-speed and yaw-rate commands, and receive the same 960-decision budget. Each VLA consumes the causal observation channels supported by its architecture, while BRINE executes its complete multimodal policy. Table 4 therefore measures end-to-end closed-loop behavior under native policy interfaces with a common task and action protocol. Appendix Table 10 gives the complete protocol, and Appendix Tables 11–12 report a complementary native-adaptation study.

Table 4: End-to-end closed-loop comparison on the shared 72-task benchmark. "2D head" denotes the common frozen-backbone ridge adapter to body-frame forward speed and yaw rate. Simple and complex columns show successes among 48 and 24 tasks. The VLA policies directly produce bounded control commands, whereas BRINE uses its adaptive M_0–M_5 cognitive-routing policy.

Policy	Success↑	Simple↑	Complex↑	Collisions↓	Progress↑	SPL↑
Octo-Base-1.5 + 2D head	23/72	22/48	1/24	33/72	0.287	0.319
OpenVLA-7B + 2D head	24/72	24/48	0/24	34/72	0.284	0.333
SpatialVLA-4B + 2D head	24/72	24/48	0/24	33/72	0.293	0.333
π_0-Base + 2D head	8/72	8/48	0/24	45/72	0.166	0.111
$\pi_{0.5}$-Base + 2D head	2/72	2/48	0/24	30/72	0.012	0.000
SmolVLA-Base + 2D head	5/72	5/48	0/24	57/72	0.150	0.068
X-VLA-Base + 2D head	4/72	4/48	0/24	35/72	0.156	0.051
BRINE (FP32)	**62/72**	**48/48**	**14/24**	**1/72**	**0.544**	**0.829**

4.3 Counterfactual supervision and router refinement

The counterfactual study contains 192 states with complete six-branch outcomes: 155 for fitting and 37 held out by task group. We compare a risk-and-teacher-supervised router with a matched router trained using Eq. 4; initialization, optimization budget, and non-target regularization are shared. In Table 5, exact six-mode accuracy and binary upgrade precision, recall, and F1 each reach 1.000 under counterfactual supervision. Expected return rises from 0.6488 to 0.9038, a paired difference of $+0.2550$ (95% cluster-bootstrap confidence interval $[+0.1392, +0.3771]$; 10,000 resamples), while mean regret falls from 2.9723 to approximately zero. Appendix Table 20 reports the corresponding calibration and expected-cost measures.

Table 5: Router results on 37 held-out counterfactual states. Both routers select over the complete M_0–M_5 route space. Accuracy scores the exact six-mode choice; precision, recall, and F1 score the binary upgrade decision. Return and expert count are expectations under the router policy; regret uses the selected mode.

Router training	Accuracy↑	Precision↑	Recall↑	F1↑	Return↑	Regret↓	Experts↓
Risk and teacher	0.459	0.364	0.308	0.333	0.6488	2.9723	0.466
Counterfactual supervision	**1.000**	**1.000**	**1.000**	**1.000**	**0.9038**	**0.00001**	**0.351**

Starting from the risk-and-teacher-supervised router, full-feedback refinement raises fitting-state policy expected return from 0.7538 to 0.8513 and changes 30 deterministic route choices while leaving every guarded fitting context unchanged. We then evaluate it on a fresh paired 24-task closed-loop set. Table 6 shows that actual expert-forward duty falls from 0.6094 to 0.4820, a relative reduction of 20.9% and a paired difference of -0.1273 (95% confidence interval $[-0.1795, -0.0739]$; paired permutation $p = 0.00020$, Holm-adjusted $p = 0.00080$ across four prespecified endpoints). The refined router completes 24/24 tasks, versus 23/24 for the baseline, and both policies incur zero collisions. Normalized progress changes by $+0.0011$ (95% confidence interval $[-0.0314, +0.0306]$), while SPL rises from 0.8685 to 0.9566. Across 1,542 routed decisions, the shield preserves all 287 high-risk routes; on the 1,255 low-risk decisions, the refined route differs 1,013 times. The refinement therefore changes the deployed allocation where permitted and realizes a statistically supported reduction in expert computation under high-risk authority.

Table 6: Closed-loop evaluation of contextual-bandit refinement over M_0–M_5; the shield retains the reference route on high-risk decisions. Progress is normalized goal progress; duty is actual expert forwards per decision, normalized by three.

Router	Success↑	Collisions↓	Progress↑	SPL↑	Expert duty↓
Risk-conditioned reference	23/24	0/24	0.5636	0.8685	0.6094
BRINE contextual bandit	**24/24**	**0/24**	**0.5647**	**0.9566**	**0.4820**

4.4 Role-aware precision and expert ablations

On expert calibration inputs, role-aware mixed precision preserves FP32 routing and role decisions with zero measured action-output error while selected SR, SV, and CP neural operations use lower precision. Peak *temporary allocated* memory falls by 7.13%; both configurations complete 24/24 paired closed-loop tasks with zero collisions (Table 7).

Table 7: Precision study with a shared routing policy and route semantics. Temporary peak, routing agreement, and action error use expert calibration inputs; completion and collisions use paired closed-loop evaluation. Agreement and error are relative to FP32.

Configuration	Temp. peak↓ (MiB)	Route agree↑	Max action error↓	Success↑	Collisions↓
FP32 reference	7.024	1.000	0	24/24	0/24
BRINE mixed precision	**6.523**	**1.000**	**0**	**24/24**	**0/24**

Table 8: Role removal on the 24 complex tasks; all variants use standard precision.

Variant	Success↑	Collisions↓	Goal progress↑	SPL↑
SR removed	9/24	0/24	0.550	0.374
SV removed	11/24	4/24	0.589	0.428
CP removed	12/24	0/24	0.592	0.487
BRINE	**14/24**	1/24	**0.610**	**0.571**

The role-removal study in Table 8 uses the same 24 complex tasks with legal route remapping and global action projection. Removing SR produces the largest success decrease and removing SV the highest collision count, while full BRINE leads in success, goal progress, and SPL. Capability-transition records verify CP consumption at the next decision.

Appendix Tables 18 and 19 report matched role-specific challenges: SR and SV improve their targeted settings, memory enables ten additional temporal-handoff successes out of 16, and CP revisions are consumed at the next decision. Compound challenges further demonstrate context-dependent role composition.

5 Conclusion

BRINE allocates cognitive computation among state refinement, safety verification, and capability planning according to risk, capability, progress, and memory, with legal role combinations and safety-constrained action projection governing execution. Six-branch matched continuations supervise this allocation, full-feedback contextual-bandit refinement reduces low-risk expert computation under the risk shield, and role-aware mixed precision retains FP32 semantic authority. The standard-precision system completes 62/72 paired tasks, compared with 48/72 for reactive control and 52/72 for always-on deliberation; counterfactual supervision improves held-out expected return by 0.255. In their paired evaluations, refinement reduces expert-forward duty by 20.9% while completing every task with zero collisions and preserving every high-risk route; mixed precision reduces peak temporary allocated memory by 7.13% while preserving complete task success. Together, these results establish efficient, safety-conscious cognitive allocation learned from matched consequences.

AI USE STATEMENT

Generative AI tools were used solely for language polishing and reviewer-style critique to identify clarity and presentation issues. The research design, system implementation, experiments, analysis, and conclusions were produced and verified by the authors.

REFERENCES

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 2669–2678, 2018. doi: 10.1609/aaai.v32i1.11797.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.010.

Jeremy Paul Coffelt, Peter Kampmann, and Bilal Wehbe. SAVOR: Sonar-aided visual odometry and reconstruction for autonomous underwater vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1236–1243, 2024. doi: 10.1109/IROS58592.2024.10801846.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090.

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

Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T. Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do As I Can, Not As I Say: Grounding language in robotic affordances. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318. PMLR, 2023. URL https://proceedings.mlr.press/v205/ichter23a.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. 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 *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. OneTwoVLA: A unified vision-language-action model with adaptive reasoning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ebfabf372037aaa4a8d92c9b457ece3e-Abstract-Conference.html.

Xiaomin Lin, Nare Karapetyan, Kaustubh Joshi, Tianchen Liu, Nikhil Chopra, Miao Yu, Pratap Tokekar, and Yiannis Aloimonos. UIVNAV: Underwater information-driven vision-based navigation via imitation learning. In *2024 IEEE International Conference on Robotics and Automation*, pp. 5250–5256, 2024. doi: 10.1109/ICRA57147.2024.10611203.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1gs9JgRZ.

Xiaohuan Pei, Yuxing Chen, Siyu Xu, Yunke Wang, Yuheng Shi, and Chang Xu. Action-aware dynamic pruning for efficient vision-language-action manipulation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/1223ca2b14513106ad55db6a537337f9-Abstract-Conference.html.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Jiayuan Gu, Zhigang Wang, Yan Ding, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action models. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.011.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. MemoryVLA: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/1fa3d6ccbcd15f7285fee666b2bd57be-Abstract-Conference.html.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025. URL https://arxiv.org/abs/2506.01844.

Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine Learning Proceedings 1990*, pp. 216–224, 1990. doi: 10.1016/B978-1-55860-141-3.50030-4.

Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, 16(52):1731–1755, 2015. URL https://jmlr.org/papers/v16/swaminathan15a.html.

Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. HAQ: Hardware-aware automated quantization with mixed precision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8604–8612, 2019. doi: 10.1109/CVPR.2019.00881.

Yuhao Xu, Yantai Yang, Zhenyang Fan, Yufan Liu, Yuming Li, Bing Li, and Zhipeng Zhang. QVLA: Not all channels are equal in vision-language-action model's quantization. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/fa064215307efaad75bebc7a2e3194a4-Abstract-Conference.html.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal Patel, Paul Liang, Daniel Khashabi, Cheng Peng, Rama Chellappa, Tianmin Shu, Alan Yuille, Yilun Du, and Jieneng Chen. World-In-World: World models in a closed-loop world. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5b4263be85820683d78675cc18d2efc7-Abstract-Conference.html.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Tai Wang, Ya-Qin Zhang, Jingjing Liu, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/630d293833e09e1ecd892a898a20b074-Abstract-Conference.html.

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. WMPO: World model-based policy optimization for vision-language-action models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/6605c7d1c1ae1521b328d036df5be1ae-Abstract-Conference.html.

A CAUSAL EXECUTION AND COGNITIVE ROUTING

A.1 NETWORK INSTANTIATION AND OPTIMIZATION

The multimodal state constructor maps RGB and three-channel sonar through convolutional encoders, valid point-cloud features through a pointwise encoder and masked pooling operation, and observed state through a feed-forward encoder. A task-token GRU conditions four-head multimodal attention followed by two Transformer fusion layers, producing a 128-dimensional state representation. The reactive controller combines this representation with local grid, scan, and goal-relative navigation features in a 256-dimensional GRU with four-head causal attention. Its two-dimensional action head predicts bounded forward speed and yaw rate; a separate head estimates action uncertainty. The state constructor is optimized with masked reconstruction and cross-modal contrastive losses (weights 1.0 and 0.1), followed by reactive-policy optimization with task-action and uncertainty losses. The training population contains 700 task episodes and 198,721 frames; the state constructor and reactive controller use 20 and 30 epochs, respectively, with AdamW, learning rate 3×10^{-4}, and batches of 32 per GPU. A separate 100-task development population is used for evaluation.

SR uses separate RGB and sonar convolutional encoders and point-cloud pooling, each with 32-dimensional modality features. A 256-dimensional feed-forward trunk combines these features with the frozen 128-dimensional state representation and masked risk and capability inputs; its output heads predict the state correction and modality confidences. SV and CP use distinct 256-dimensional feed-forward trunks. SV predicts current safety, next-decision veto, risk at horizons of 1, 5, 10, and 20 decisions, and up to eight action constraints. CP predicts a revision event, one of six capability classes, and a revised local subgoal. Their learned predictions are consumed by the FP32 safety and capability rules described above. SR is trained for eight epochs with batch size 64 and AdamW at learning rate 3×10^{-4} using state-correction mean-squared error and confidence binary cross-entropy with weight 0.25. SV and CP use the same batch size, epoch count, and learning rate; their losses supervise safety, veto, horizon risk, feasible constraints, revision events, legal successor capabilities, and revised subgoals.

The router uses separate encoders for the state–capability–memory representation and the risk diagnostics, each with width 128. A learned fusion gate combines them before binary upgrade, five-way conditional route, and benefit-prediction heads. Episodic memory stores at most 128 realized entries per capability and retrieves up to 16 relevant entries for a decision. The risk-and-teacher-supervised router uses eight epochs of AdamW at learning rate 3×10^{-4}, batch size one, and gradient-norm clipping at one. For the matched supervision comparison, three random seeds share the same initialization and use 1,024 optimizer steps on the 155 fitting states; only the supervision target changes between conditions. The 37 held-out states are reserved for evaluation.

Observations and control state. At each control decision, the deployed policy receives the current robot state, RGB image, point-cloud observation, sonar measurement, task command, and the outcome history available up to that decision. The state constructor records observation availability and derives reliability, modality-conflict, action-uncertainty, failure, and risk-trend features. The task command enters through the goal interface. Episodic memory is queried using the current causal representation and capability context, and each memory entry is created after its corresponding action and outcome have been observed.

The capability graph restricts which revisions the capability planner may propose. A confirmed revision becomes visible at the next decision. The safety verifier can restrict or veto the current action, and the global action projection remains active for every route. When the feasible action set is empty, the controller holds a safe action and obtains a new observation before trying another route. These ordering rules give each confirmed subgoal revision a clear causal boundary.

Algorithm 1 Causal routed control

Require: Causally available observations and task command; action history; current capability; realized episodic memory; router and safety rules.
Ensure: An executed action and an updated causal state.

Construct state h_t and reliability and risk diagnostics from current measurements.
Run the early SR diagnostic; compute the reactive action proposal and retrieve eligible memory.
Form routing context z_t and propose $m_t \in \mathcal{M}$ with the six-mode router.
Apply the high-risk override, emergency rule, and capability-graph legality checks to m_t.
if SR participates in the effective mode **then**
    Use its refined representation for the current action proposal.
end if
if CP participates **then**
    Propose a legal revision for decision $t + 1$.
end if
if SV participates **then**
    Obtain its constraints and veto for the current action.
end if
Fuse action proposals and project onto global limits and any active SV constraints.
if the feasible action set is empty **then**
    Execute a safe hold.
else
    Execute the projected action.
end if
Observe the actual transition and update causal history and eligible memory.
Expose any confirmed CP revision at decision $t + 1$ only.

The six routes are listed in Table 1. Their labels denote which role outputs participate in the current branch. The SR diagnostic can run before route selection, and measured expert-forward duty includes every executed role forward. Equation 4 uses the branch role count as its target-side complexity cost.

B Counterfactual Supervision and Router Optimization

B.1 Paired counterfactual continuations

Each counterfactual training item begins with one saved snapshot. All six branch continuations reuse that snapshot's scene, robot and controller state, causal sensor history, and exogenous disturbance schedule. The continuation is limited to 20 decisions. Its purpose is to compare cognitive choices at a common state: branch outcomes provide offline training targets. The continuation records capability completion, clipped goal progress, unsafe collision, selected roles, and a dedicated safe-abort outcome for infeasible safety constraints. The branch value uses coefficients $\alpha = 0.1$, $\beta = 10$, and $\gamma = 0.01$ in Eq. 4. Target selection uses a value tolerance of 0.05 and an upgrade-improvement margin of 0.1 over M_0.

Algorithm 2 Six-branch counterfactual target construction

Require: Saved snapshot s_i; legal modes $\mathcal{M} = \{M_0, \ldots, M_5\}$; shared disturbance schedule; horizon and value weights.
Ensure: Complete branch-value vector, preferred mode, and upgrade and role targets.

for each $m \in \mathcal{M}$ **do**
  Restore s_i and the same exogenous disturbance schedule.
  Continue the simulator for at most 20 decisions under mode m.
  Record completion $C_i(m)$, progress $P_i(m)$, collision $U_i(m)$, and role count $c(m)$.
  Set $J_i(m) \leftarrow C_i(m) + \alpha P_i(m) - \beta U_i(m) - \gamma c(m)$.
end for
Retain all six outcomes for supervision and reserve deployment memory for realized outcomes.
if a collision-free branch exists **then**
  Restrict the target search to collision-free branches.
else retain all six branches for target search.
end if
Let $\mathcal{B}_i$ contain branches within the specified value tolerance of the best.
Select $m_i^\star$ from $\mathcal{B}_i$, preferring fewer roles.
Derive the upgrade target from safety rescue or sufficient improvement over M_0.
Derive the conditional role target from $m_i^\star$ and return the branch-value vector and targets.

The counterfactual collection has 192 states with six complete outcomes per state, or 1,152 branch continuations. Training uses 155 states; 37 states are held out by task group. The three compared router variants use matched initialization, optimizer budget, and non-target regularization. The paired return effect is assessed with 10,000 bootstrap resamples clustered by held-out group.

B.2 Risk-shielded contextual-bandit refinement

The full branch-value vector supports an offline contextual-bandit objective. We denote the risk-and-teacher-supervised router by π_{ref} and update only the proposal router; the state constructor, reactive controller, experts, memory readout, action prior, and action aggregation retain their calibrated parameters. The objective uses the normalized six-branch returns on the 155 fitting states, an entropy coefficient $\beta_H = 0.005$, a high-risk reference-policy KL coefficient $\lambda_H = 0.1$, learning rate 10^{-3}, and 1,201 optimization steps. A separate closed-loop population evaluates the resulting policy.

Algorithm 3 Risk-shielded router refinement

Require: Fitting states with normalized six-branch returns; router π_{ref}; high-risk predicate.
Ensure: A selected router and a shielded effective route at deployment.

Initialize the proposal router π_θ from π_{ref}.
Partition fitting contexts into $\mathcal{L}$ and $\mathcal{H}$ with the high-risk predicate.
for each optimization step **do**
  Compute the six-mode proposal distribution $\pi_\theta(\cdot \mid z_i)$.
  On $\mathcal{L}$, maximize expected normalized branch return and entropy.
  On $\mathcal{H}$, minimize $D_{\mathrm{KL}}(\pi_{\mathrm{ref}} \| \pi_\theta)$.
  Update only the proposal router with Eq. 5.
end for
if the high-risk predicate is active at deployment **then**
  Use the reference router's effective mode.
else
  Use the refined router's nominal mode.
end if
Apply capability legality, emergency handling, and global action projection.

The risk shield activates when current conflict, anomaly, failure, or inconsistency exceeds its threshold of 0.5, or when the existing emergency predicate is active. The emergency predicate combines failure, inconsistency, and uncertainty with weights 0.4, 0.3, and 0.3, respectively, and threshold 0.8. The paired study compares the refined router with its risk-conditioned baseline. Its 287 shield-active decisions preserve the baseline route; the refined router selects a different deterministic mode on 1,013 of the 1,255 lower-risk decisions.

C ROLE-AWARE PRECISION ASSIGNMENT

Weights are retained in FP32. Selected neural operations use CUDA FP16 autocast, while router and memory readout, semantic thresholds, feasibility geometry, capability legality, revision lifecycle, and the final action projection remain in FP32. Expert outputs are returned to FP32 before they influence these decisions. The selected assignment retains lower-precision computation in early RGB and sonar encoding within SR, a horizon-risk prediction operation within SV, and a revised-subgoal prediction operation within CP. Thus each role contains genuine lower-precision neural computation even though its decision interface remains FP32.

Algorithm 4 Semantic precision calibration and execution

Require: FP32 experts; 256 calibration inputs; ordered candidate precision assignments.
Ensure: Selected assignment and FP32-authoritative closed-loop execution.

Set the selected assignment $p^\star \leftarrow \mathrm{FP32}$.
for each candidate assignment p in the predetermined order **do**
  Evaluate the same calibration inputs with FP32 and assignment p.
  Check finite outputs, exact hard route and role agreement, and unchanged parameter identity.
  Measure maximum action error, fallback count, and peak temporary allocated memory.
  if all semantic checks hold, error $\leq 5 \times 10^{-4}$, fallback is zero, and temporary memory falls **then**
    Set $p^\star \leftarrow p$ and **break**.
  end if
end for
At execution, run only the selected neural operations under FP16 autocast.
Return expert outputs to FP32 before routing, capability, safety, and action decisions.
if a mixed-precision expert output is non-finite **then**
  Reevaluate that expert in FP32.
end if
Apply the unchanged FP32 semantic and safety rules and execute the projected action.

Calibration measures peak *temporary allocated* memory with a batch-one standardized pass through SR, SV, and CP. The selected assignment reduces this peak from 7,365,120 to 6,840,320 bytes, or 7.13%, while retaining exact hard semantic agreement and zero measured action-output error on the calibration inputs. The paired closed-loop evaluation confirms 24/24 completion and zero collisions for both mixed precision and FP32.

D EXPERIMENTAL ENVIRONMENT AND EVALUATION PROTOCOL

D.1 SERVER AND SIMULATOR

All reported experiments, training, and GPU measurements were produced on a Linux server with two NVIDIA GeForce RTX 3090 GPUs (24 GB each), NVIDIA driver 535.309.01, Python 3.10.20, PyTorch 2.5.1 with CUDA 12.1 support, and NumPy 1.26.4. The environment uses Isaac Sim 4.5. The final router and precision confirmation uses one designated RTX 3090 for neural inference and expert calibration. Hardware identity is recorded for each study, and the paired FP32/mixed-precision microbenchmark uses the same device and inputs.

The benchmark uses rendered RGB, geometric, point-cloud, and sonar observations with a kinematic continuation model. The nominal control interval is 50 ms (20 Hz); RGB is sampled at 20 Hz, while sonar, point-cloud, scanning, and map-fusion updates are specified at 10 Hz. A control decision uses the most recent causally available measurement from each modality. The maximum closed-loop episode length is 960 decisions. Task completion requires stable goal acquisition, including the 0.78 m goal-distance and 0.08 rad heading criteria for eight decisions. These criteria and the collision and safety rules are shared across paired variants.

The three evaluation scene families are industrial frames, open tanks, and pipe corridors. The primary 72-task population contains 24 parent tasks from each family and is partitioned into 48 simple and 24 complex tasks. A parent task specifies the initial scene and task conditions; paired policies are evaluated from the same parent conditions. The standard-precision full-system comparison uses this population, and Table 9 identifies the population used by each study.

Table 9: Experimental populations supporting the system, counterfactual-supervision, contextual-bandit, and mixed-precision evaluations.

Study	Population	Purpose
System, scene, and VLA comparisons	72 paired tasks	Closed-loop completion, collision, progress, path efficiency, and computation.
Role-specific route challenges	Seven groups of 16 paired tasks	Targeted SR, SV, CP, composition, and memory behavior.
Expert-removal comparison	24 complex parent tasks	Targeted removal of each expert from the full system.
Six-branch router supervision	155 fit states; 37 held-out states	Offline branch-choice learning and paired counterfactual evaluation.
Bandit-refined router	24 paired simple tasks	Closed-loop task, safety, routing, and expert-forward comparison.
Role-aware mixed precision	256 calibration inputs; 24 paired simple tasks	Numerical and semantic calibration followed by closed-loop confirmation.

D.2 Policy comparisons and role interventions

The reactive comparator uses the fast controller as its complete policy. The always-on comparator activates deliberative roles unconditionally, subject to the same action limits and safety authority. Standard-precision BRINE uses risk- and capability-conditioned routing. The external comparison includes Octo, OpenVLA, SpatialVLA, π_0, $\pi_{0.5}$, SmolVLA, and X-VLA on the same 72 parent tasks. In the uniform comparison, every VLA backbone is frozen and the same closed-form ridge recipe fits a canonical two-dimensional action head from four causal anchors per training episode. OpenVLA and SpatialVLA use RGB and language; Octo uses RGB and language; SmolVLA, π_0, $\pi_{0.5}$, and X-VLA additionally use their native state interface. BRINE uses RGB, state, task, sonar, geometric observations, memory, and role-specific experts. Table 10 records the controlled physical and evaluation factors.

Table 10: Protocol for the VLA comparisons. Canonical-head and native-adaptation policies use their architecture-supported causal inputs. Both VLA studies and BRINE are evaluated on the same 72 physical tasks.

Protocol factor	Canonical-head VLAs	Native-adapted VLAs	BRINE (FP32)
Evaluation population	72 tasks: 48 simple, 24 complex	Same 72 tasks	Same 72 tasks
Initial conditions	Matched manifest slots, seeds, poses, and disturbances	Same physical initial conditions	Same physical initial conditions
Command interface	$[v_{\mathrm{body}}, \omega_{\mathrm{yaw}}]$	$[v_{\mathrm{body}}, \omega_{\mathrm{yaw}}]$	$[v_{\mathrm{body}}, \omega_{\mathrm{yaw}}]$
Action bounds	$v \in [0, 0.85]$ m/s; $\omega \in [-1.2, 1.2]$ rad/s	Same bounds	Same bounds
Control and budget	First action, then replanning; 960 decisions	Same execution and budget	Same execution and budget
Stable completion	$d \leq 0.78$ m, $\lvert\Delta\psi\rvert \leq 0.08$ rad for 8 decisions	Same criterion	$d \leq 0.78$ m, $\lvert\Delta\psi\rvert \leq 0.08$ rad for 8 decisions
Model fitting	700 tasks, 2,800 anchors; common ridge recipe	700 tasks, 11,200 anchors; 100-task development selection	Risk- and capability-conditioned cognitive policy
Inference	Deterministic, batch one	Deterministic, batch one	Deterministic, batch one

The canonical-head comparison isolates the upstream representation under one adaptation rule. The complementary study uses each model family's native adaptation mechanism: QLoRA over the OpenVLA language-action policy, SmolVLA action-expert tuning, and an Octo two-dimensional action head with native non-transformer adaptation modules. All three use 11,200 causal training

anchors, checkpoint selection on 400 development anchors, and the held-out 72-task population for closed-loop evaluation.

Table 11: Complementary native-adaptation results on the shared 72-task benchmark. Simple and complex columns show successes among 48 and 24 tasks.

Policy and adaptation	Success↑	Simple↑	Complex↑	Collisions↓	Progress↑	SPL↑
OpenVLA-7B + QLoRA action policy	11/72	11/48	0/24	20/72	0.200	0.153
SmolVLA + native action-expert tuning	11/72	11/48	0/24	52/72	0.166	0.113
Octo-Base-1.5 + native 2D action head	5/72	5/48	0/24	53/72	0.143	0.066
BRINE (FP32)	**62/72**	**48/48**	**14/24**	**1/72**	**0.544**	**0.829**

Table 12 verifies the native adaptation runs through their optimization records, selected development objectives, parameter deltas, and complete closed-loop evaluation coverage. Development objectives follow the native loss of each model family and select its checkpoint.

Table 12: Native-adaptation training health checks. Changed tensors are reported over trainable tensors; CE denotes action-token cross-entropy.

Adaptation	Train	Dev	Steps	Trainable	Changed	Selected dev objective
OpenVLA QLoRA	11,200	400	700	110.83M	852/878	0.222636 CE
SmolVLA action expert	11,200	400	700	99.88M	122/155	0.063862 flow loss
Octo native action head	11,200	400	700	7.09M	15/17	0.478079 action loss

The final audit verifies 216/216 native-adapted VLA evaluation episodes, three distinct checkpoint identities, and parameter changes for every adaptation. Together, the protocol and health checks connect the reported closed-loop outcomes to completed, model-specific adaptation procedures.

The expert-removal study uses the same 24 complex parent tasks for full BRINE and each removal condition. Each intervention disables the selected role and remaps the route to a legal combination of the remaining roles, while capability restrictions and global action projection remain in force. CP-dependent routes preserve their required safety verification. The role-specific challenge study complements these architectural interventions with selected route behaviors on seven 16-task groups.

D.3 Metrics and inference

Success and collision are episode-level binary outcomes. Normalized goal progress measures advancement toward the goal under the common task definition. SPL averages success-weighted path efficiency over the full evaluation population; unsuccessful episodes contribute zero, while successful episodes with zero shortest-path and executed lengths contribute one. Selected-deliberation duty is the fraction of decisions routed to a deliberative mode. Actual expert-forward duty counts completed SR, SV, and CP neural forwards and divides their sum by three times the number of decisions; because a role may be called more than once per decision, this quantity can exceed one. Decision-latency summaries are measured from executed decisions, and the system comparison averages per-episode latency summaries.

For paired binary outcomes, we use exact McNemar tests. For paired continuous differences, we use bootstrap confidence intervals or paired permutation tests as specified for each study. Reported system-level p values are adjusted with Holm's procedure within the prespecified comparison family. The counterfactual return comparison uses a 10,000-resample cluster bootstrap over held-out task groups. Raw endpoint values and paired differences jointly describe task, safety, and computation effects.

E Additional Experimental Results

E.1 Decision allocation and closed-loop routing diagnostics

Table 13 gives decision-allocation and latency summaries for the paired 72-task comparison; the primary success, safety, and mean-computation endpoints appear in Table 2. Table 14 reports complementary closed-loop routing diagnostics with the denominators stated in its caption.

Table 13: Decision allocation and latency on the same 72 tasks. Selected duty is the episode-averaged fraction of decisions routed to deliberation; forward duty counts actual expert calls. Latency columns average the corresponding per-episode decision-latency summaries in milliseconds.

Controller	Selected duty↓	Forward duty↓	Mean↓	95th pct.↓	99th pct.↓
Reactive controller	0.000	0.333	202.9	242.6	257.7
Always-on deliberation	1.000	1.010	328.9	432.9	472.2
BRINE (FP32)	0.644	0.611	287.9	425.8	469.1

Table 14: Closed-loop routing diagnostics for standard-precision BRINE on the 72 paired tasks. The final two metrics use simple decisions and complex parent tasks, respectively.

System	Upgrade precision↑	Upgrade recall↑	Simple upgrade rate	Missed complex↓
BRINE (FP32)	0.611	1.000	0.443	0.000

E.2 Task-difficulty and scene strata

The simple and complex task outcomes appear in Table 3 in the main text. The scene-level completion and collision counts appear in Table 15; Table 16 adds progress and SPL for each scene family. Each scene contributes 24 paired tasks.

Table 15: Success and collision counts by scene family, 24 paired tasks per family.

	Industrial frame		Open tank		Pipe corridor	
Controller	Success↑	Collision↓	Success↑	Collision↓	Success↑	Collision↓
Reactive controller	16/24	5/24	16/24	0/24	16/24	7/24
Always-on deliberation	15/24	0/24	21/24	0/24	16/24	0/24
BRINE (FP32)	**20/24**	1/24	**22/24**	**0/24**	**20/24**	**0/24**

Table 16: Scene-level normalized goal progress and SPL on the paired 72-task population; each scene contributes 24 tasks.

	Industrial frame		Open tank		Pipe corridor	
Controller	Progress↑	SPL↑	Progress↑	SPL↑	Progress↑	SPL↑
Reactive controller	0.363	0.667	0.338	0.667	0.343	0.667
Always-on deliberation	0.498	0.583	0.639	0.789	0.484	0.583
BRINE (FP32)	**0.538**	**0.829**	0.584	**0.914**	**0.509**	**0.744**

E.3 Paired system effects

Table 17 reports the paired effects underlying selected system-level claims. Effects are BRINE minus the indicated comparator. A positive success or progress difference and a negative collision difference favor BRINE. The intervals and adjusted tests are computed on the full paired population with scene and task strata preserved in the paired design.

Table 17: Selected paired 72-task effects with 95% confidence intervals and Holm-adjusted p values.

Comparator	Endpoint	Difference	95% interval	Adjusted p
Reactive controller	Success	$+0.1944$	$[+0.1111, +0.2917]$	0.00195
Reactive controller	Collision	-0.1528	$[-0.2361, -0.0694]$	0.01270
Reactive controller	Goal progress	$+0.1956$	$[+0.1163, +0.2842]$	0.00170
Always-on deliberation	Success	$+0.1389$	$[+0.0694, +0.2222]$	0.02344

E.4 ROLE-SPECIFIC CHALLENGE MATRIX

The role-challenge population contains seven mechanistically selected groups of 16 paired tasks each. Each contrast in Table 18 is computed within its matched group. The SR comparison shows strong gains in its targeted condition, the SV comparison reduces observed collisions, and CP revisions are consumed by the next decision as designed. The compound comparisons demonstrate context-dependent role selection. The temporal-memory handoff comparison measures the contribution of realized episodic memory to the full system.

Table 18: Complete directional contrast summary for the seven role-challenge groups ($n = 16$ paired tasks per row). Positive success and progress and negative collision differences favor the first route. Multiple rows within a group reuse its 16 parents.

Challenge context	Paired contrast	Δ success↑	Δ collision↓	Δ progress↑
State-refinement demand	$M_1 - M_0$	$+0.875$	-0.313	$+0.864$
Safety-verification demand	$M_2 - M_0$	$+0.438$	-0.313	$+0.647$
Capability revision	$M_4 - M_2$	$+0.063$	0.000	$+0.029$
Joint SR and SV demand	$M_3 - M_0$	$+0.500$	-0.563	$+0.627$
Joint SR and SV demand	$M_3 - M_1$	-0.375	0.000	-0.271
Joint SR and SV demand	$M_3 - M_2$	-0.063	0.000	-0.060
Joint SV and CP demand	$M_4 - M_2$	$+0.063$	0.000	-0.025
Three-role demand	$M_5 - M_0$	$+0.313$	-0.438	$+0.535$
Three-role demand	$M_5 - M_3$	-0.063	0.000	-0.055
Three-role demand	$M_5 - M_4$	-0.125	0.000	-0.117
Temporal memory handoff	BRINE–memory ablated	$+0.625$	0.000	$+0.729$

Table 19 adds path efficiency and computation measures for selected contrasts from the complete matrix.

Table 19: Paired role-specific challenge results ($n = 16$ per comparison) across seven matched challenge groups. Values are first-controller minus second-controller differences. Latency is mean decision latency in milliseconds; duty is actual expert-forward duty.

Comparison	Δ success↑	Δ collision↓	Δ progress↑	Δ SPL↑	Δ duty↓	Δ latency↓
SR versus reactive	+0.875	-0.313	+0.864	+0.859	+0.051	+168.7
SV versus reactive	+0.438	-0.313	+0.647	+0.438	+0.315	+249.7
CP with SV versus SV	+0.063	0.000	+0.029	+0.062	+0.348	-5.5
BRINE vs. memory ablation	+0.625	0.000	+0.729	+0.615	+0.212	+7.6

The temporal-memory condition completes 10/16 tasks with memory and 0/16 in the ablated condition; normalized progress is 0.847 versus 0.118, and SPL is 0.615 versus 0. The corresponding expert-forward duty is 0.726 versus 0.514. In the state-refinement challenge, nominal M_1 completion is 14/16 versus 0/16 for M_0. Across the relevant challenge contrasts, safety-driven routing adapts the effective mode in six of the 32 M_1-labelled episodes.

E.5 ROUTER PREDICTION AND RISK SHIELDING

Table 20 adds probability-calibration and expected-cost information to the 37-state counterfactual-supervision results in Table 5. The Brier score and expected calibration error (ECE) concern the binary upgrade probability. Table 6 reports the contextual-bandit closed-loop results.

Table 20: Held-out upgrade calibration and policy expectations on 37 counterfactual states. Lower Brier, ECE, regret, and expected role count are preferable; return is higher-is-better.

Router training	Brier↓	ECE↓	Return↑	Regret↓	Roles↓
Risk and teacher	0.361960	0.373330	0.648771	2.972311	0.465657
Counterfactual supervision	**0.00000009**	**0.000060**	**0.903803**	**0.000012**	**0.351297**

The closed-loop confirmation in Table 6 contains 1,542 routed decisions. The risk shield is active for 287 decisions and preserves the baseline route in every case. Among 1,255 lower-risk decisions, 1,013 deterministic routes differ. Fitting-state expected return rises from 0.75385 to 0.85132; the deployed duty reduction remains significant after Holm correction across progress, duty, success, and collision endpoints ($p = 0.00080$). Both policies record zero collisions, and minimum clearance is 1.5682 m and 1.5543 m, respectively.