CPNAV: Calibrated Vision–Language Navigation for Object-Goal Search in Unseen Environments

Qizhao Chen[1], Yuanhong Zeng[2], Shoh Nishino[1], and Anushri Dixit[1]

***Abstract*— Directing a robot to locate a target object in unseen environments remains a central challenge in embodied AI. Vision–Language Model (VLM)–based methods provide rich semantic reasoning but often produce uncalibrated and overconfident action decisions. We introduce CPNAV, a navigation framework that combines collision-aware planning with statistically calibrated VLM action scoring for map-free object-goal navigation. From RGB-D input, CPNAV constructs a navigability map, generates collision-free motion primitives, and queries two VLMs for goal detection and action scoring. The resulting scores are calibrated offline via conformal prediction (CP) using oracle actions obtained from a modified tree search, providing finite-sample guarantees at a user-specified risk level. The CP layer is model-agnostic and can wrap any VLM policy. Online, the calibrated threshold prunes the search tree by filtering unreliable actions with backtracking. Experiments on the HM3D object-goal navigation benchmark show that CPNAV improves over the 4 Lets look at the main problem of the baseline uncalibrated method. The agent is placed at a random location in an apartment, the goal is to find the plant a baseline in both reliability and efficiency. In simulation, the proposed method achieves a relative improvement of 6.17% in the success rate and 11.34% in the success weighted by path length (SPL) compared to the non-calibrated baselines. In real-world deployment on a modified Hiwonder MentorPi robot, CPNAV achieves relative improvements of 24.99% in Success Rate and 15.20% in SPL compared to baselines.**

I. INTRODUCTION

Robotic navigation in complex unseen indoor environments remains challenging because effective decision making must tightly couple low-level path planning with high-level semantic understanding, while generalizing to novel layouts without prior maps or goal location information [1]. Traditional rule-based planners can handle geometric obstacle avoidance but rely on hand-tuned rules and expert knowledge, making them brittle in unforeseen situations [2], [3]. In contrast, learning-based navigation planners, particularly those based on reinforcement learning (RL), often require painstaking reward shaping and large-scale training data [4]. Such RL-based planning approaches still struggle to generalize across diverse and unseen environments [5], [6].

A key limitation of both rule-based and purely learning-based navigation methods is the absence of semantic understanding, which is essential for efficient goal-directed behavior in realistic environments [7]. Object-goal navigation benefits from reasoning about object–object and object–scene relationships and from leveraging commonsense priors about given environment layouts. By integrating these semantic priors, the robotic agent can effectively constrain the search process, significantly reducing the branching factor of the search tree and prioritizing paths that align with contextual expectations. Semantic information and common sense reasoning also allow the agent to infer the probable locations of goal objects even when they are not immediately present within the agent's current field of view [8]–[10]. Recent VLMs (Gemini, LLaVA, GPT-4o) enable injection of commonsense and scene priors into navigation, bridging geometric perception and high-level planning [11], [12]. Through large-scale multimodal pretraining, these models acquire semantic priors that support generalization to previously unseen environments, reducing reliance on environment-specific training [13]–[15].

Although pretrained VLMs supply strong semantic cues, their internal heuristics, which are the learned rules that guide their behavior, are not calibrated for sequential decision making [16]. VLM outputs are optimized for token prediction or image–text alignment rather than long-horizon action selection under physical constraints. As a result, they often assign non-trivial confidence to unsafe, redundant, or goal-irrelevant actions [17], [18]. In this work, we tackle the problem of unreliable high-level action scoring in unseen environments. Without an explicit mechanism to quantify and control confidence, VLM-driven navigation can assign non-trivial scores to semantically plausible but suboptimal or unsafe actions. Over long horizons, such miscalibration can manifest as inefficient exploration, excessive backtracking, and action loops.

Conformal Prediction (CP) provides a principled statistical mechanism to address this issue by converting uncalibrated model scores into prediction sets [19], [20]. Rather than trusting raw VLM confidences, CP uses a held-out calibration set to compute thresholds that control the error rate at a user-specified risk level. Under the assumption of exchangeability, this ensures that the probability of excluding the correct action is bounded by the calibration threshold, independent of the VLM architecture or training process.

Building on this principle, we introduce CPNAV, illustrated in Fig. 1, a conformal prediction–augmented navigation framework that integrates semantic reasoning with statistically calibrated decision-making. The main contributions of this work are as follows: (1) We introduce a conformal calibration layer that converts uncalibrated VLM action scores into statistically valid safety sets, guaranteeing that at

[1]Qizhao Chen, Shoh Nishino, and Anushri Dixit are with the Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, Los Angeles, CA, USA. qizhaoc@ucla.edu, shohnishino@ucla.edu, anushridixit@ucla.edu

[2]Yuanhong Zeng is with the Department of Electrical and Computer Engineering, University of California, Los Angeles, Los Angeles, CA, USA. yuanhongzeng@ucla.edu

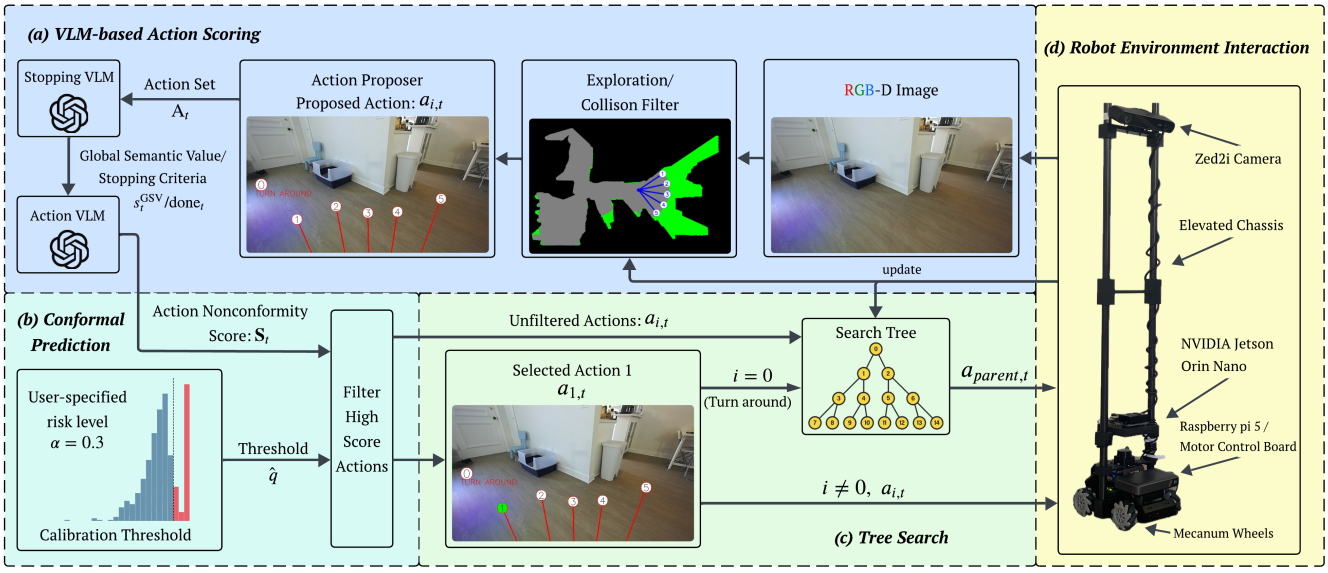


Fig. 1: Overview of the CPNAV pipeline. (a) The action proposer constructs candidate actions from RGB-D input, prioritizing unexplored regions while enforcing collision safety. (b) The threshold obtained via offline conformal calibration is used to filter out actions whose scores fall below the calibrated bound. (c) The highest-scoring untried action that satisfies the threshold is selected for expansion in the search tree and executed accordingly. (d) Hardware experiments are conducted on a modified Hiwonder MentorPi robot platform.

risk level α, the oracle action is excluded with probability at most α under exchangeability. (2) An offline bottleneck path search is formulated to extract oracle-aligned trajectories, enabling conformal prediction for long-horizon embodied navigation. (3) Calibrated safety sets are integrated into a backtracking tree-search policy that prunes low-confidence branches and improves efficiency under fixed step budgets. (4) Extensive simulation and hardware experiments validate the approach. On HM3D, CPNAV improves Success Rate by 6.17% and SPL by 11.34%; on a physical modified MentorPi robot platform, it achieves 24.99% and 15.20% relative improvements in Success Rate and SPL, respectively.

II. Related Work

A. *VLM-Augmented Vision-Language Navigation*

Recent research has shifted from reinforcement learning toward leveraging the zero-shot reasoning of Vision-Language Models (VLMs) for embodied navigation [1], [11], [13]. Recent advancements in VLM-augmented navigation have expanded the paradigm of spatial integration by utilizing large-scale pre-trained models for end-to-end navigational QA and spatial reasoning [21]–[24], as well as hierarchical world-model and long-horizon planning [25]–[28]. These frameworks leverage diverse strategies such as task-preferenced grounding for flexible objective specification [12], [26] and zero-shot semantic localization that merges scene priors with frontier-based exploration policies [8], [13], [29]. Despite these gains, a persistent challenge remains: these methods often treat VLM outputs as absolute truths. As highlighted in recent literature, this leads to catastrophic failure modes where agents are misled by hallucinated semantic relevance or overconfident scores for physically inaccessible paths [16], [17]. This vulnerability underscores the critical need for a formal statistical mechanism to quantify and calibrate VLM uncertainty during sequential decision-making.

B. *Conformal Prediction for Reliable Decision-Making*

Conformal prediction (CP) provides a distribution-free framework for uncertainty quantification with finite-sample coverage guarantees [19], [20]. By converting model scores into calibrated prediction sets, CP bounds the probability of excluding the correct label at a user-specified risk level, making it appealing for safety-critical robotics applications.

Building on these theoretical foundations, several recent works have successfully integrated conformal prediction into the robotics pipeline to enhance safety and reliability. CP has been utilized in perception and object detection to transform standard classification outputs into class prediction sets, preventing autonomous systems from making overconfident but incorrect categorical decisions [30]. Beyond perception, CP has been applied to motion prediction and trajectory forecasting, where it constructs uncertainty-aware prediction tubes around learned dynamics models, enabling downstream planners to account for bounded future-state uncertainty [31].

Despite these successes, a significant gap remains in high-level navigation decision-making for object-goal navigation. Existing research primarily focuses on uncertainty-aware stopping criteria or confidence estimation for downstream tasks, rather than calibrating the sequential action-selection process itself [32]. While such methods determine when an agent should terminate exploration, they do not provide

statistical guarantees over which navigation actions should be executed during long-horizon object-goal search. In contrast, our work applies conformal prediction directly to VLM-based action scoring, constructing calibrated safety sets that govern the navigation policy throughout the exploration process.

III. Method

A. Problem Definition

We consider object-goal navigation in previously unseen indoor environments. An embodied agent is initialized at an unknown 2D pose with position and orientation, $\mathbf{x}_0 = (x_0, y_0, \theta_0)$, within a continuous environment. The goal is to navigate to an instance of a target object category g (e.g., chair, bed, toilet) without prior knowledge of the map or goal location. At each time step t, the agent receives an RGB-D observation $I_t = (I_t^{\text{RGB}}, I_t^{\text{depth}})$ and its estimated pose $\mathbf{x}_t = (x_t, y_t, \theta_t)$. Based on the current observation history, the agent selects a continuous low-level polar action $a_t = (\theta_t, r_t)$. Here θ_t denotes the heading angle of the action, and r_t denotes the travel distance. By executing these actions, the agent aims to find the shortest path while following a semantically reasonable path to find the goal object.

B. VLM-based Action Scoring

CPNav generates collision-aware motion primitives from RGB-D input and queries VLMs for stopping decisions and per-action semantic confidences. The resulting scene-aware scores guide tree-based exploration and provide the calibrated signals used by the conformal prediction layer.

*1) **Action proposer**:* The action proposer generates a set of collision-aware motion primitives from the current RGB-D observation (Fig. 1a). Building upon the strategy of VLM-Nav [21], we integrate a depth-based collision filter to ensure safe exploration. First, the agent generates a navigability mask from the depth image, identifying traversable regions with heights below 5 cm. The agent then samples rays within the horizontal field of view, marching each ray to determine the maximum collision-free travel distance $d_{i,t}$. This calculation accounts for the agent's physical footprint, defined by its radius r_{agent} and a safety clearance $r_{\text{clearance}}$.

The resulting distances form a set of candidate motion primitives. To prioritize efficient exploration, we prune trivial or excessively long motions and filter out actions that lead to previously explored regions. Additionally, we include an auxiliary turn-around action $a_{0,t} = (\pi, 0)$, which executes a physical in-place rotation at the root step or serves as a backtracking signal to the parent node. The final candidate set is defined as:

$$\mathrm{A}_t = \{a_{i,t} = (\theta_{i,t}, d_{i,t})\}_{i=1}^{N}, \tag{1}$$

where N denotes the number of feasible motion primitives remaining after collision filtering and exploration prioritization. These candidates are subsequently proposed to the VLM for final selection.

*2) **VLM Scoring and Stopping**:* At each action step t, the agent queries a VLM to determine if the agent has found the desired goal and what action to take to move to the goal.

*a) **Goal Reaching Determination and Global Semantic Scoring**:* Given the current RGB observation I_t^{RGB} and the object-goal description g, the stopping VLM outputs

$$(\text{done}_t, s_t^{\text{GSV}}) = f_{\text{stop}}(I_t^{\text{RGB}}, g), \tag{2}$$

where $f_{\text{stop}}(\cdot)$ denotes the stopping VLM query. $\text{done}_t \in \{0,1\}$ indicates whether the agent sees the goal. $s_t^{\text{GSV}} \in [0,1]$ represents the global semantic value (GSV) [32]. GSV measures how promising the current view is for exploration, reflecting the availability of navigable free space. Higher values indicate stronger scene-level cues, and we use GSV to weight per-action confidence scores.

*b) **Action Selection and Confidence Scoring**:* For each candidate motion primitive $a_{i,t} \in \mathrm{A}_t$, the action VLM assigns a semantic confidence score

$$s_{i,t}^{\text{action}} = f_{\text{action}}(I_t^{\text{RGB}}, a_{i,t}, g), \tag{3}$$

where $s_{i,t}^{\text{action}} \in [0,1]$ reflects how likely executing $a_{i,t}$ will lead toward the goal under the current observation.

To incorporate scene-level navigation potential, we fuse the per-action confidence with the global semantic value:

$$s_{i,t} = s_t^{\text{GSV}} \cdot s_{i,t}^{\text{action}}. \tag{4}$$

The final confidence vector used for decision-making at an observation step is

$$\mathbf{S}_t = [s_{1,t}, \ldots, s_{N_t,t}]. \tag{5}$$

The fused score $s_{i,t}$ is used for tree-based exploration, oracle path search, and conformal calibration. Concise versions of the prompts and their corresponding JSON response examples are shown in Fig. 2.

C. Tree-Based Exploration

A tree-structured search policy is used to systematically explore candidate actions while enabling efficient backtracking.

*1) **Search Tree Formulation**:* The agent's exploration is modeled as a dynamically growing tree. Each node corresponds to a visited state of the robot, and each edge represents an executed action that connects a parent state to a newly reached state. Branching occurs when multiple candidate actions are available from the same state.

The robot maintains an exploration log:

$$\mathrm{H}_t = \{(\mathbf{x}_k, \mathrm{A}_k, \mathbf{S}_k)\}_{k=1}^{t}, \tag{6}$$

where $\mathbf{x}_k$ is the simulator state, A_k the candidate actions, and $\mathbf{S}_k = \{s_{i,k}\}$ their VLM scores. When transitioning from $\mathbf{x}_k$ to $\mathbf{x}_t$, we assign the parent of node t to be k. The tree structure is defined by the parent link, while H_t stores the associated data.

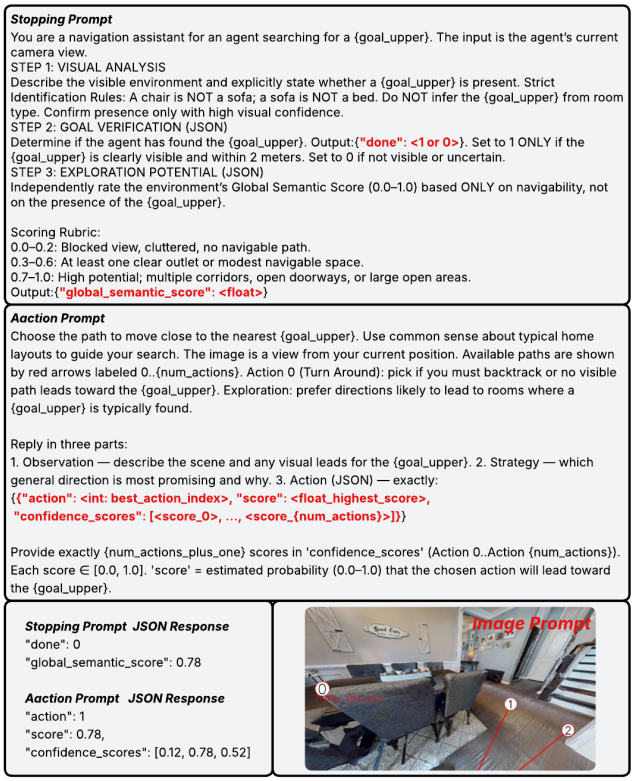


Fig. 2: Concise stopping and action prompts with their JSON responses. The stopping prompt queries the stopping condition and GSV; the action prompt queries the per-action confidence score. The manually normalized confidence score is multiplied by GSV to form the final per-action score.

*2) **Local Backtrack**:* If the VLM selects the turn-around action (index 0), the robot reverts to the parent node k^* of current timestep t and retrieves its stored state and action set. The remaining untried actions are

$$\mathrm{R}_{k^*} = \{a_{i,k^*} \in \mathrm{A}_{k^*} \mid a_{i,k^*} \notin \mathrm{T}_{k^*}\}. \tag{7}$$

where T_k denote the set of actions already attempted from step k.

If R_{k^*} is nonempty, the remaining actions are sorted by their semantic confidence scores and the highest-ranked untried action is executed from the parent state $\mathbf{x}_{k^*}$. Otherwise, the agent initiates a global backtracking procedure. At the root node of step 0, the turn-around action is executed directly.

*3) **Global Backtrack**:* If local backtrack exhaust recent branches, a global backtrack searches the full history up to the current time step t for any node that still contains untried actions:

$$\mathrm{R}_t^{\text{global}} = \bigcup_{k=1}^{t} \left\{ a_{i,k} \in \mathrm{A}_k \,\middle|\, a_{i,k} \notin \mathrm{T}_k,\ i \neq 0 \right\}. \tag{8}$$

If $\mathrm{R}_t^{\text{global}}$ is empty, exploration terminates. Otherwise, the action with the highest semantic score among $\mathrm{R}_t^{\text{global}}$ is selected. The agent is restored to the corresponding state $\mathbf{x}_k$ and resumes exploration by executing that action.

D. Offline Best Path Search

After the goal is reached for the first time, we build a nominal path to the goal. However, the initial successful trajectory may be circuitous or include low-confidence edges. To establish a robust baseline for calibration, we refine this history into an optimal "Oracle" path by solving for the max-min Path, whcih represent a sequence of edges (u,v) with confidence scores $s_{u,v}$ that maximizes the minimum edge weight:

$$P^* = \arg\max_{P} \left(\min_{(u,v)\in P} s_{u,v} \right). \tag{9}$$

We use the max–min criterion because calibration is performed over the whole trajectories. P^* yields the path's bottleneck confidence, providing a conservative, trajectory-level threshold instead of per-step guarantees. We implement this via a search algorithm modified from Breadth-First Search, which prioritizes paths with higher bottleneck scores. The resulting path P^* provides the minimum confidence threshold experienced during a successful goal-reaching mission.

To further optimize the Oracle path, the agent re-enters a global backtracking phase where exploration is constrained to untried actions $a_{i,k}$ with scores strictly exceeding the current bottleneck $s(P^*)$:

$$\mathrm{R}_t^{\text{refine}} = \{a_{i,k} \in \mathrm{R}_t^{\text{global}} \mid s_{i,k} > \min_{(u,v)\in P^*} s_{u,v}\}. \tag{10}$$

The agent continues this search until it reaches the goal again, triggers a turn-around, or converges to an existing node. In any such event, the search algorithm is re-executed until the total step limit is met. If a higher bottleneck score is found, P^* is updated. This iterative refinement ensures the final threshold S^* used for Conformal Prediction represents the most reliable path discovered.

E. Offline Calibration with Conformal Prediction

To provide rigorous guarantees in unseen environments, we employ a calibration framework that aligns the VLM's internal confidence with actual success rates. This process ensures that the robot is aware of its own uncertainty and only executes actions when they fall within a statistically backed safety set.

*1) **Non-conformity Scoring**:* We begin by establishing a calibration dataset consisting of 1000 independent and identically distributed scenarios, where each scenario contains an environment observation and its corresponding ground-truth action. For each scenario, we compute a non-conformity score E_i, which quantifies the model's error relative to the oracle action a_i^* identified during our offline path search:

$$E_i = 1 - s_{a^*,i} \tag{11}$$

where $s_{a^*,i}$ is the semantic confidence score assigned by the VLM to the ground-truth action. The VLM is tasked to give unnormalized scores, which reduces bias. We then normalize the scores for cross-step consistency. High E_i values indicate that the model was "surprised" by the correct action, while low values indicate high alignment.

*2) **Beta-Calibration for Uncertainty Alignment**:* Because the calibration set $\mathscr{D}_{cal}$ is finite, the empirical quantile alone may not guarantee the target success rate $1-\alpha$ on new data. To account for this finite-sample uncertainty, we use a confidence level $\delta = 0.99$. We define a "stricter" adjusted risk level $\hat{\alpha}$ such that the following condition holds via the Beta distribution:

$$\text{Beta}^{-1}(\delta; n+1-v, v) \geq 1-\alpha \tag{12}$$

where $v = \lfloor (n+1)\hat{\alpha} \rfloor$ and Beta^{-1} is the quantile function of the Beta distribution. This adjustment allows us to state with 99% confidence that the robot will achieve at least a $1-\alpha$ success rate during online deployment.

*3) **Quantile Application**:* Using the adjusted risk $\hat{\alpha}$, we compute the calibrated threshold $\hat{q}$ from our distribution of non-conformity scores:

$$\hat{q} = \text{Quantile}\left(\{E_1, \ldots, E_n\}, \frac{\lceil (n+1)(1-\hat{\alpha}) \rceil}{n}\right) \tag{13}$$

The calibrated quantile $\hat{q}$ is then used to construct the admissible prediction set during online deployment.

F. Online Deployment with Calibrated Safety Sets

During online inference, the agent uses the offline-calibrated threshold $\hat{q}$ to manage uncertainty. Instead of a greedy execution of the highest VLM-scored action, the agent filters candidate actions into a "safety set" to maintain the success guarantee.

*1) **Prediction Set Construction**:* At each time step t, the agent receives candidate actions A_t and their semantic scores $\mathbf{S}_t$. It constructs a prediction set $C(\xi_t)$ containing only actions that satisfy the safety threshold:

$$C(\xi_t) = \{a_{i,t} \in \mathrm{A}_t \mid s_{i,t} \geq 1-\hat{q}\} \tag{14}$$

If $C(\xi_t) = \emptyset$, the agent triggers the turn-around action $a_{0,t}$ to initiate backtracking. Otherwise, the agent selects the action with the highest semantic score within $C(\xi_t)$ for execution.

Let a_t^* denote the oracle action at time t from the offline oracle path search. Under exchangeability between calibration and deployment episodes, the conformal threshold $\hat{q}$ ensures

$$\mathbb{P}(a_t^* \in C(\xi_t), \forall t > 0) \geq 1-\alpha \tag{15}$$

Because calibration is performed over complete navigation episodes using the bottleneck oracle path, this guarantee applies at the trajectory level. With probability at least $1-\alpha$, the oracle-consistent path is not pruned during online exploration. The Beta correction further ensures this bound holds with confidence $1-\delta$.

*2) **Tree-Search Exploration**:* The calibrated safety set governs which actions are eligible for execution. Among the admissible actions, the agent always selects the highest-scoring untried action with backtracking, forming a tree-structured search process (Fig. 1(c)).

If the selected action corresponds to index 0 (the turn-around action), the agent rewinds to its parent node and executes the next untried action with the highest score from that parent state. This same procedure is triggered when all admissible actions along the current branch have been exhausted.

When a branch of the search tree is fully explored, the agent performs a global backtracking step. It traverses the tree to locate the closest ancestor node that still contains untried admissible actions and selects the highest-scoring action at that node. This strategy ensures systematic exploration of the search tree while prioritizing semantically confident decisions.

IV. Experimental Setup

A. Simulation and Hardware Configuration

We evaluate CPNAV using the Habitat-Sim platform with the HM3D dataset [33], [34]. We consider two experimental settings: a simulation setup and a hardware setup. The simulation experiment represents an idealized evaluation within the Habitat simulator. The agent is equipped with a camera having a high horizontal field of view (HFoV), and a maximum budget of 200 steps is allowed. The hardware calibration experiment constrains the simulator to match physical hardware limitations. The agent uses a reduced camera HFoV and is given a much lower step limit. In addition, we conduct a simulation parameter sensitivity analysis to evaluate the robustness of CPNAV with respect to sensing configuration and model capacity. To ensure computational efficiency, we implement a parallelized execution pipeline on an NVIDIA RTX 4090 GPU.

For the hardware validation, we employed a Hiwonder MentorPi M1 mobile platform equipped with a Mecanum drivetrain for omnidirectional mobility. State estimation was performed using the SLAM Toolbox, which fused data from a 2D LiDAR scanner, wheel odometry, and an Inertial Measurement Unit (IMU) to provide robust localization. We used a ZED 2i stereo camera to obtain rectified RGB-D data for environmental perception. The specific experimental parameters and sensor configurations are detailed in Table I.

TABLE I: Experimental Parameters

Parameter	Simulation	Hardware
r_{agent}, $r_{\text{clearance}}$	0.1 m, 0.05 m	
Resolution	640×480 RGB-D	
Camera Height	0.9m	
HFoV / Down Pitch	131°/0.45 rad	101°/0.4 rad
VLM Model	GPT-5 Nano	
Step Limit	200	100
Target Risk (α)	0.45	0.3
Confidence (δ)	0.99	
Episodes (Cal/Eval)	500 / 200	500 / 24 (Real-world)

B. Baselines and Metrics

To evaluate the effect of conformal calibration on action selection, we compare CPNAV against alternative action-selection strategies that differ in how uncertainty is incorporated [16].

- **No Calibration:** The agent always executes the action with the highest raw VLM score $s_{i,t}$, and all actions are considered valid.

- **Simple Set:** Uses a fixed, manually-tuned threshold τ. Any action where $s_{i,t} > \tau$ is considered valid.
- **Ensemble Set:** Estimates uncertainty by querying the VLM multiple times and uses result frequency to construct confidence vector $\mathbf{S}_t$.
- **Direct Set:** The VLM is explicitly prompted to list all feasible actions that could lead to the goal relying on the model's internal reasoning.

We evaluate each baseline over 200 runs, matching the online simulation deployment setting. Performance is measured by Success Rate (SR) and Success weighted by Path Length (SPL) [35]. Success is manually verified: the target must be visible in the final observation with an unobstructed path; stopping at an incorrect target is counted as failure. For successful episodes, SPL is computed using the geodesic distance from the start pose to the verified goal via the Habitat API. We additionally report SR and SPL under fixed step budgets, along with mean distance traveled, to further assess efficiency and exploration behavior.

For the hardware experiments, we compare CPNAV against the no calibration baseline in two different apartment environments. Under the baseline, no confidence threshold is applied, and all proposed actions are considered admissible and executed in descending order of their raw VLM scores. For each trial, the agent is randomly initialized within the apartment, and the target object category is randomly selected from the same goal set used in simulation. The baseline is evaluated over 24 trials, matching the number of hardware runs conducted for CPNAV. We also use SR, SPL, and mean distance traveled for evaluation.

V. RESULTS

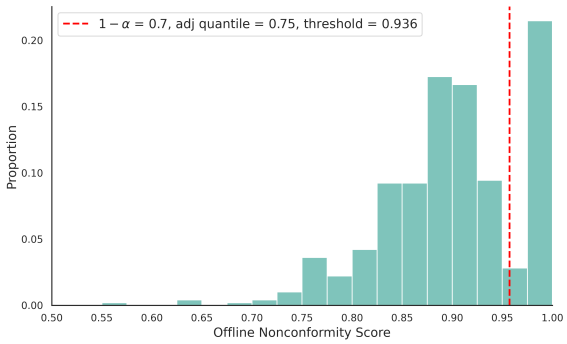
A. *Simulation Results*

*1) **Offline Calibration Threshold Analysis**:* We compute the conformal threshold from the VLM confidence assigned to the oracle action in each calibration episode; these oracle-action scores form the calibration set. Fig. 3 shows the empirical distributions for the two regimes (simulation and hardware), with the dashed red line marking the calibrated threshold $\hat{q}$ (Sec. IV-F). With a Beta correction at confidence $\delta = 0.99$, the resulting thresholds are

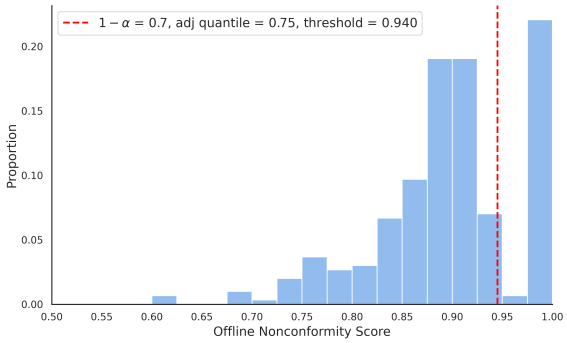
$$\hat{q}_{\text{sim}} = 0.060, \qquad \hat{q}_{\text{real}} = 0.064,$$

corresponding to adjusted quantile levels of approximately 0.25.

Two observations follow. First, oracle-action scores are positively skewed and frequently modest, indicating that even correct actions do not always receive high raw VLM confidence in unseen environments. Second, because calibration is performed in the nonconformity space, larger values of E_i correspond to lower semantic confidence. The calibrated threshold $\hat{q}$ therefore lies in the upper tail of the E_i distribution, meaning that conformal prediction removes only the highest nonconformity (i.e., lowest-confidence) fraction of oracle-aligned actions while retaining the majority of reliable decisions. During online deployment, the calibrated threshold $\hat{q}$ determines the admissible action set through the safety-set construction defined in Eq. (14), thereby filtering out low-confidence actions from execution.



(a) Simulation calibration histogram. The cutoff $\hat{q} = 0.936$.



(b) Hardware calibration histogram. The cutoff $\hat{q} = 0.94$.

Fig. 3: Empirical distributions of nonconformity score used for conformal calibration. The hardware calibration is done in the simulation setting. Each panel shows the distribution of nonconformity scores E_i. The dashed red line indicates the calibrated score threshold $\hat{q}$ upper bound of $1-\bar{\alpha}$, where $\bar{\alpha}$ is the Beta-distribution adjusted risk level.

*2) **Simulation Results**:* Table II reports the online evaluation results in the simulation setting. The CPNAV policy uses the offline-calibrated quantile $\hat{q}_{\text{sim}} = 0.94$ to construct admissible action sets at every decision step, which corresponds to a risk level of $\alpha = 0.3$. The Simple Set and Ensemble Set baselines instead use a fixed confidence bound of $1-\alpha = 0.7$. The Prompt Set directly asks for the action set without using a bound.

Compared to all baselines, CPNAV achieves the highest Success Rate (SR) of 86.00%, corresponding to a relative improvement of 6.17% over the no-calibration baseline. Path efficiency is also improved, with a relative SPL improvement of 11.34% and a relative reduction in mean travel distance of 17.45%.

Among heuristic threshold-based methods, the best-performing variant achieves an SR of 76.67% and an SPL of 0.3950. CPNAV surpasses this manually tuned approach with a relative improvement of 12.17% in SR and a relative improvement of 13.65% in SPL, demonstrating that data-

TABLE II: simulation online evaluation (200 episodes). SR<50 denotes Success Rate within 50 steps

Policy	Threshold	SR	SR<50	SR<100	SPL	SPL<50	SPL<100	Mean Dist (m)
Simple Set	0.7	76.67%	59.67%	72.33%	0.3950	0.3742	0.3919	32.79
Ensemble Set	0.7	74.00%	63.33%	70.00%	0.3908	0.3803	0.3879	28.01
Prompt Set	N/A	79.67%	58.33%	70.67%	0.3871	0.3564	0.3784	37.40
No Calibration	0.0	81.00%	60.33%	75.67%	0.4032	0.3771	0.3982	35.38
CPNAV	**0.936**	**86.00%**	**73.00%**	**80.00%**	**0.4489**	**0.4323**	**0.4442**	**29.21**

driven conformal calibration provides more reliable and principled action filtering than fixed heuristic thresholds.

These improvements are consistent with the intended effect of conformal calibration. By filtering out low-confidence actions, which often correspond to semantically plausible but suboptimal branches, the calibrated policy reduces unnecessary exploration and prioritizes higher-quality trajectories. Importantly, under a fixed step budget, this selective filtering prevents the agent from wasting steps on low-value detours and preserves more of the available interaction budget for high-score actions. As a result, the policy is better able to allocate its limited steps toward trajectories that are more likely to successfully reach the goal, leading to improved Success Rate, higher path efficiency, and greater robustness.

B. *Simulation Parameter Sensitivity Analysis*

We analyze the sensitivity of CPNAV to sensing configuration and model capacity by varying one parameter at a time from the CPNAV setting, which is using GPT-5o-nano, height = 0.9 m, FOV = 131°, calibrated threshold $\hat{q} = 0.064$, and voxel map is enabled to encourage exploration. Table III reports Success Rate, SPL, and mean travel distance.

TABLE III: Parameter sensitivity analysis in simulation.

Variant	SR	SPL	Mean Distance (m)
CPNAV ($\hat{q} = 0.064$)	**86.00%**	**0.4489**	**29.21**
No Calibration ($\hat{q} = 0.0$)	81.00%	0.4032	35.38
Height = 0.45 m	67.00%	0.3139	44.63
Height = 1.70 m	79.33%	0.4053	30.40
FOV = 69°	40.00%	0.1871	44.82
FOV = 101°	72.67%	0.3048	40.92
Model = GPT-4o-mini	71.33%	0.3366	34.40
No Voxel Map	66.00%	0.3644	36.47

The table highlights four main trends. First, sensing geometry strongly influences performance. Lower camera height or a narrower field of view substantially reduces SR and SPL while increasing travel distance, reflecting less informative observations and less efficient exploration. Second, model capacity matters. Replacing GPT-5o-nano with a smaller VLM degrades both reliability and path efficiency, indicating that calibration effectiveness depends on the semantic strength of the underlying model. Third, conformal calibration consistently outperforms the uncalibrated variant by filtering low-confidence actions and thereby improving both success rate and trajectory efficiency. Finally, maintaining a voxel-based occupancy map further enhances performance by preventing redundant revisits.

C. *Hardware Results*

We evaluated CPNAV on a physical Hiwonder MentorPi M1 platform across 24 randomized object-goal episodes in two residential apartment layouts. For hardware calibration we used the threshold $\hat{q}$ of 0.936, corresponding to a target risk level of $\alpha = 0.30$. Table IV summarizes the primary metrics comparing the uncalibrated baseline to CPNAV with the calibrated threshold.

TABLE IV: Hardware results.

Policy	SR	SPL	Mean Distance (m)
No Calibration	66.67%	0.4204	9.0330
CPNAV	**83.33%**	**0.4843**	**6.2949**

CPNAV substantially improved reliability and efficiency on the physical robot. The Success Rate increased from 66.67% to 83.33%, corresponding to a relative improvement of 24.99%. Path efficiency improved with a relative SPL increase of 15.20%, while mean distance traveled decreased with a relative reduction of 30.31%. In both simulation and hardware experiments, CPNAV consistently increased Success Rate and SPL while reducing unnecessary exploration. Under the stricter step budget and real-world execution noise of the hardware trials, these gains were more pronounced. Conformal calibration systematically filters low-confidence actions that are often semantically plausible but suboptimal, producing more direct trajectories and improved robustness to sensing and actuation uncertainty.

Differences between simulation and hardware results stem from deliberate configuration changes and environment complexity. The hardware experiments used a rectified HFoV of 101°, whereas 131° in simulation, and a modified camera pitch. This reduces visible semantic context and alters viewing geometry. In addition, state-estimation drift during long-horizon backtracking can degrade voxel-map consistency in the real system. Finally, the physical apartments tested were structurally simpler and smaller than many HM3D scenes, which helps explain the higher hardware success rates despite more constrained sensing and noisier odometry.

VI. Conclusion and Limitations

Conclusion. In this work, we proposed CPNAV, a conformal prediction-based calibration framework for vision-language navigation. By filtering low-confidence actions

through data-driven conformal calibration, CPNAV improves both reliability and path efficiency without modifying the underlying policy architecture. Across simulation and real-world experiments, CPNAV consistently achieves higher Success Rate and SPL while reducing unnecessary exploration and travel distance. The improvements are particularly pronounced in hardware deployments, where strict step budgets and real-world execution noise amplify the cost of suboptimal decisions. These results demonstrate that conformal calibration provides a principled and effective mechanism for action filtering.

Limitations. Despite its effectiveness, two limitations remain. First, during offline calibration with the refined tree search, nodes are defined solely based on spatial proximity without accounting for orientation, which may introduce poor estimation. Second, the method relies on scores from a vision-language model accessed via the ChatGPT API; its inherent stochasticity and occasional inconsistencies can introduce variability in action scoring and affect calibration stability. Future work will explore orientation-aware state representations and improved robustness to VLM variability.

REFERENCES

[1] R. Alqobali, M. Alshmrani, R. Alnasser, A. Rashidi, T. Alhmiedat, and O. M. Alia, "A survey on robot semantic navigation systems for indoor environments," *Applied Sciences*, vol. 14, no. 1, 2023.

[2] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[3] S. Tellex *et al.*, "Understanding natural language commands for robotic navigation and manipulation," in *National Conference on Artificial Intelligence (AAAI)*, 2011.

[4] G. Dulac-Arnold, D. Mankowitz, and T. Hester, "Challenges of real-world reinforcement learning," *arXiv preprint arXiv:1904.12901*, 2019.

[5] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," in *International Conference on Learning Representations (ICLR)*, 2020.

[6] X. Zhao *et al.*, "On evaluation of embodied navigation agents in new environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[7] F. Giustra, S. Fontana, G. Mongardi, and A. Giusti, "Semantic navigation: A survey of methods and challenges," *Frontiers in Robotics and AI*, vol. 10, 2023.

[8] D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," in *Neural Information Processing Systems (NeurIPS)*, 2020.

[9] E. Deng *et al.*, "Grounded language models for robotic navigation," *arXiv preprint arXiv:2305.12345*, 2023.

[10] A. Cowley *et al.*, "Searching for objects with semantic search spaces," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[11] H. Ha and S. Song, "Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models," in *Conference on Robot Learning (CoRL)*, 2023.

[12] Y. Zhang *et al.*, "A survey of vision-language-action models for embodied manipulation," in *arXiv preprint arXiv:2510.24795*, 2025.

[13] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Clip on wheels: Zero-shot object navigation as object localization and exploration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[14] D. Hendrycks, S. Basart *et al.*, "The many faces of robustness: A critical analysis of out-of-distribution generalization," *ICCV*, 2021.

[15] R. Gandikota *et al.*, "Distilling vision-language models for robust embodied generalization," *arXiv preprint arXiv:2310.15878*, 2023.

[16] A. Z. Ren *et al.*, "Robots that ask for help: Uncertainty alignment for large language model planners," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[17] M. Ahn *et al.*, "Do as i can, not as i say: Grounding language models in robotic affordances," in *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[18] S. Kadavath *et al.*, "Language models (mostly) know what they know," *arXiv preprint arXiv:2207.05221*, 2022.

[19] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*, ser. Springer Series in Statistics. Springer, 2005.

[20] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," *Foundations and Trends in Machine Learning*, vol. 16, no. 4, pp. 494–591, 2023.

[21] D. Goetting, H. G. Singh, and A. Loquercio, "End-to-end navigation with vision-language models: Transforming spatial reasoning into question-answering," in *arXiv preprint arXiv:2411.05755*, 2024.

[22] D. Shah, B. Osinski, B. Ichter, and S. Levine, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Conference on Robot Learning (CoRL)*, 2023.

[23] G. Zhou, Y. Hong, and Q. Wu, "Navgpt: Explicit reasoning in vision-and-language navigation with large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024, pp. 7641–7649.

[24] G. Zhou, Y. Hong, Z. Wang, X. E. Wang, and Q. Wu, "Navgpt-2: Unleashing navigational reasoning capability for large vision-language models," *arXiv preprint arXiv:2407.12366*, 2024.

[25] T. Wang, X. Li, F. Lu, T. Gong, J. Dong, W. Xue, S. Qu, C. Bai, and G. Chen, "Conavbench: Collaborative long-horizon vision-language navigation benchmark," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

[26] S. Li, D. Huang, Y. He, Y. Fu, Y.-G. Jiang, and X. Xue, "Tp-mddn: Task-preferenced multi-demand-driven navigation with autonomous decision-making," in *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2025.

[27] L. Zhang *et al.*, "Mem2ego: Empowering vision-language models with global-to-ego memory for long-horizon embodied navigation," *arXiv preprint arXiv:2502.14254*, 2025.

[28] Y. Cai, X. He, M. Wang, H. Guo, W.-Y. Yau, and C. Lv, "Cl-cotnav: Closed-loop hierarchical chain-of-thought for zero-shot object-goal navigation with vision-language models," *arXiv preprint arXiv:2504.09000*, 2025.

[29] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "VLFM: vision–language frontier maps for zero-shot semantic navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 42–48.

[30] Z. Mei, A. Dixit, M. Booker, E. Zhou, M. Storey-Matsutani, A. Z. Ren, O. Shorinwa, and A. Majumdar, "Perceive with confidence: Statistical safety assurances for navigation with learning-based perception," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 14 150–14 157.

[31] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, "Safe planning in dynamic environments using conformal prediction," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5116–5123, 2023.

[32] A. Z. Ren, J. Clark, A. Dixit, M. Itkina, A. Majumdar, and D. Sadigh, "Explore until confident: Efficient exploration for embodied question answering," in *Robotics: Science and Systems (RSS)*, 2024.

[33] M. Savva *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9339–9347.

[34] K. Yadav *et al.*, "Habitat-matterport 3d semantics dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 4927–4936.

[35] P. Anderson, A. X. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.