Extended Abstract

Motivation Autonomous drones are well suited to indoor inspection and search-and-rescue, but reliable onboard navigation remains difficult. A drone must recognize a queried object, avoid clutter, and act from limited sensing and compute. We study this problem through SINGER (Adang et al., 2025), a language-conditioned navigation system in which a drone receives a natural-language goal such as "green clock" and must reach the corresponding object from monocular RGB alone. Because real flight data is expensive and collisions are costly, we train and evaluate in photorealistic 3D Gaussian Splatting simulation (Low et al., 2024; Kerbl et al., 2023), following the SINGER setting where policies are designed for zero-shot sim-to-real transfer. The original controller, trained by behavioral cloning from a geometric expert, succeeds in only about one third of our test rollouts and collides frequently. Our goal is to improve success and safety while preserving a lightweight policy suitable for onboard deployment.

Method We formulate navigation as a Markov decision process and improve SINGER through a staged training pipeline. We first train a compact visuomotor commander by behavioral cloning from RRT* trajectories tracked by a model predictive controller, which provides successful collision-free demonstrations. To make the policy more target-aware, we add centroid features derived from CLIPSeg similarity heatmaps (Lueddecke and Ecker, 2022). These features translate visual evidence for the queried object into target bearing, elevation, and apparent size without introducing additional trainable parameters. We then use DAgger (Ross et al., 2011) to reduce compounding error by relabeling the states actually visited by the learned policy. Finally, we study offline Implicit Q-Learning (IQL) (Kostrikov et al., 2022) on the remaining failure cases to test whether recovery behavior can be learned from logged trajectories.

Implementation The policy concatenates SqueezeNet visual features, drone state information, centroid features, and a short temporal history before passing the result through a two-layer commander head. The behavioral-cloning dataset contains roughly 158K observation-action pairs from 330 expert trajectories augmented by pose randomization. The vision pipeline uses CLIPSeg to produce a similarity heatmap between the language query and the current camera image. In preliminary experiments, thresholding this heatmap at the 75th percentile and adding the resulting weighted centroid as a goal condition substantially improved learning. DAgger is run for twelve aggregation rounds with the vision encoder frozen and a best-checkpoint rollback. For the offline stage, we collect seventeen hard pilot episodes and pair them with seventeen successful expert episodes on the same branches, which provides successful demonstrations for the same difficult starts before fine-tuning with IQL using expectile 0.9 and advantage-weighting temperature 0.1.

Results Goal conditioning is the largest single improvement, raising behavioral-cloning success from 36.7% to 80.7%. Adding DAgger brings the combined policy to 88.0% success on seen trajectories, with a peak of 92.0% during training and collisions reduced to 8.0%. The best policy also generalizes well to unseen launches, with per-object success close to 90%. Offline IQL improves performance on the hard branches where the policy is most likely to fail, increasing success from 35.3% to 64.7% on those specific cases. However, it reduces performance on the broader unseen benchmark to roughly 70%, indicating that the offline dataset is too narrow and perception-biased for reliable global fine-tuning.

Discussion The main remaining failures are perceptual rather than control-related. CLIPSeg returns a relative similarity heatmap even when the true object is absent or occluded, which can cause the policy to pursue false positives. Offline fine-tuning can then reinforce these misleading targets when the dataset contains blind or biased trajectories. Improving the controller alone is therefore unlikely to resolve the dominant error mode. The next step is a perception module that can distinguish confident target tracking from target absence, followed by a multi-task RL policy that switches between tracking and cautious search.

Conclusion Our extensions raise SINGER from roughly 30% to nearly 90% vision-only navigation success while reducing collisions to single digits. The project also identifies a clear bottleneck because once goal conditioning and on-policy imitation are introduced, failures are dominated by perception.

Future work should therefore prioritize robust target verification, perception-aware demonstrations, and a multi-task search-versus-track formulation for cases where the queried object is not yet visible.

Enhancing SINGER: Onboard Visual Drone Navigation through Iterative Imitation Learning and Reinforcement Fine-Tuning

Erwin Poussi
Department of Computer Science
Stanford University
erwinpi@stanford.edu

Abstract

We study language-conditioned onboard drone navigation in SINGER (Adang et al., 2025), where a quadrotor must reach a queried object from monocular RGB while avoiding obstacles. The original controller is trained by behavioral cloning from a geometric expert, but in our evaluation it succeeds in only about one third of rollouts and collides frequently. We improve the pipeline with centroid-based goal conditioning from CLIPSeg heatmaps (Lueddecke and Ecker, 2022), iterative imitation learning with DAgger (Ross et al., 2011), and offline Implicit Q-Learning (Kostrikov et al., 2022) on collected failure cases. Centroid features raise behavioral-cloning success from 36.7% to 80.7%, and DAgger further improves success to 88.0% while reducing collisions to 8.0% and preserving generalization to unseen launches. Offline fine-tuning recovers many hard branches but reduces performance on the broader benchmark, showing that the logged failures are too narrow and perception-biased for global policy improvement. Qualitative analysis shows that the remaining errors are dominated by perception because the semantic detector can produce confident false targets when the queried object is absent or occluded. These results suggest that robust target verification and a future multi-task search-versus-track policy, rather than more aggressive control fine-tuning alone, are the critical next steps for deployable language-conditioned drone navigation.

1 Introduction

Autonomous drones could make indoor inspection, inventory search, and search-and-rescue faster and safer, but these settings require more than low-level obstacle avoidance. The robot must understand a semantic goal, recognize the corresponding object from onboard sensing, and navigate through clutter without relying on external infrastructure. This is especially challenging on small aerial platforms, where sensing, compute, and data collection are all constrained.

We build on SINGER (Adang et al., 2025), a language-conditioned onboard navigation system developed at the Stanford Multi-Robot Systems Lab. In SINGER, a drone receives a natural-language query such as "green clock" and must fly toward the named object using monocular RGB observations. Training is performed in photorealistic 3D Gaussian Splatting simulation (Low et al., 2024; Kerbl et al., 2023), which enables safe data generation and repeated evaluation before real-world deployment. The broader SINGER setting is motivated by zero-shot sim-to-real transfer, so simulation quality and onboard compatibility are central design constraints rather than secondary implementation details.

The original SINGER training pipeline relies on behavioral cloning from a geometric expert. This provides a scalable source of demonstrations, but it has two important limitations. The expert plans collision-free paths in the reconstructed scene without considering what the onboard camera can actually perceive, making its trajectories physically feasible but not always informative for visual navigation. Behavioral cloning also suffers from covariate shift because once the learned policy deviates from the expert trajectory, it enters states that were not present in the training data and errors can accumulate.

We address these limitations through three additions. We first introduce centroid-based goal conditioning derived from CLIPSeg heatmaps, giving the policy an explicit estimate of where the queried object appears in the image. We then apply DAgger (Ross et al., 2011) to relabel the states visited by the learned policy and reduce compounding error. Finally, we evaluate offline reinforcement-learning fine-tuning with Implicit Q-Learning (IQL) (Kostrikov et al., 2022) on collected failure cases. Goal conditioning and DAgger substantially improve performance, while offline fine-tuning helps on targeted hard branches but hurts broader generalization. This contrast motivates a failure analysis that separates control errors from perception errors and points toward multi-task search-versus-track policies as a natural next direction.

Concretely, this project augments SINGER with lightweight centroid features computed directly from CLIPSeg heatmaps, introduces an iterative imitation-learning pipeline that corrects the distribution shift induced by behavioral cloning, and studies whether offline IQL can recover from hard failures. The resulting analysis shows that once the policy receives a reliable target cue and is trained on its own state distribution, target verification becomes the limiting factor for deployment.

2 Related Work

Imitation learning for control. Behavioral cloning has a long history in robot and vehicle control, from early neural driving systems such as ALVINN (Pomerleau, 1988) to modern visuomotor policies. Its simplicity is also its main weakness because training is performed on expert states, whereas deployment occurs on states induced by the learned policy. Small prediction errors can therefore compound and drive the agent outside the training distribution. DAgger (Ross et al., 2011) addresses this issue by repeatedly rolling out the current policy, querying the expert on the visited states, and aggregating the newly labeled data. We use this idea to expose the drone policy to its own state distribution while preserving the original visuomotor architecture.

Offline reinforcement learning. Imitation alone cannot improve beyond the expert and provides limited supervision for recovery behavior. Offline reinforcement learning offers a complementary route by optimizing a reward from fixed logged data, which is important in SINGER because generating additional trajectories is expensive and the state-action space is high-dimensional. We focus on Implicit Q-Learning (Kostrikov et al., 2022), which avoids explicit maximization over actions that are poorly represented in the dataset and is therefore well suited to narrow offline logs. We also draw conceptually on advantage-weighted updates such as AWAC (Nair et al., 2020). Instead of treating all logged actions equally, these updates give more influence to actions whose estimated return is higher than the current value baseline while reducing the influence of weaker actions. This smooth weighting is preferable to directly copying only the best-looking samples because it preserves useful variation in the dataset and avoids unstable jumps toward actions that may be overestimated.

Semantic visual navigation. Language-conditioned navigation requires linking a text query to visual evidence. We use CLIPSeg (Lueddecke and Ecker, 2022), which produces a dense similarity heatmap conditioned on a text prompt and gives the policy a target-dependent cue without training an object-specific detector. Our simulation setup follows the SOUS VIDE line of work (Low et al., 2024), which trains visual drone navigation policies in Gaussian Splatting scenes (Kerbl et al., 2023). The remaining failure modes connect to recent work on robust RGB obstacle reasoning and temporal prediction. CARE (Kim et al., 2025) explores visually generated trajectories and depth-aware image analysis for collision avoidance, while DreamZero (Ye et al., 2026) learns a world-action model that predicts future actions and states over a receding horizon rather than outputting only one-step controls. These ideas are complementary to a multi-task search-versus-track policy because obstacle-aware prediction could support safe exploration when target verification indicates that the queried object is absent or occluded.

Figure 1 summarizes the SINGER system that we extend. The simulator renders language-embedded scenes and semantic maps; the RRT* expert produces collision-free geometric trajectories; and the onboard policy maps RGB observations to control commands. The central difficulty is that the expert is geometric while the deployed policy is visual: a path can be collision-free in the reconstruction yet still provide poor supervision for a camera-limited controller.

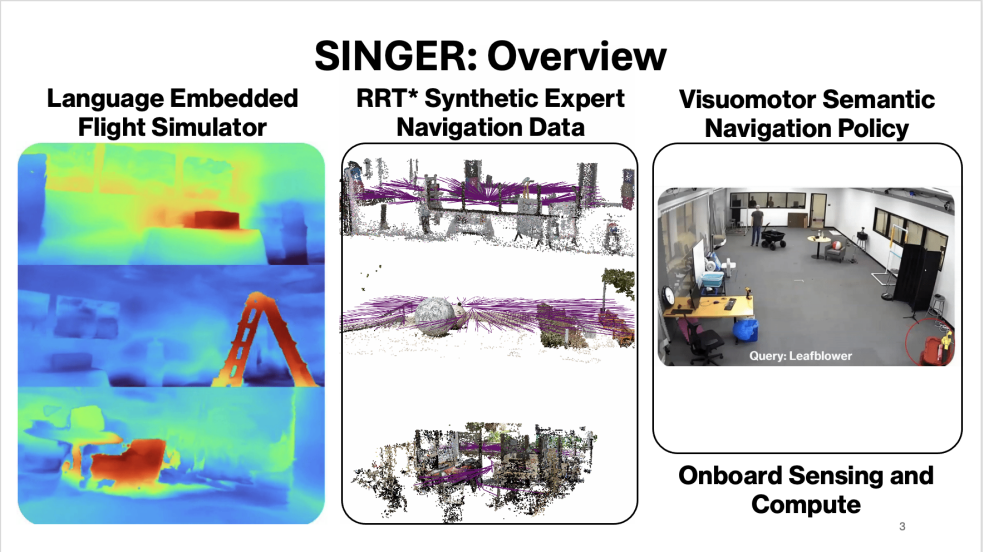


Figure 1: Overview of the SINGER system. The simulator renders language-embedded scenes and semantic maps; the RRT* expert produces geometric collision-free trajectories; and the onboard visuomotor policy navigates from RGB observations toward the queried object.

3 Method

Our approach learns a compact language-conditioned visuomotor policy entirely from synthetic data. The training pipeline is shown in Figure 2. We begin with behavioral cloning, add explicit goal conditioning, correct the policy distribution with DAgger, and finally test offline RL fine-tuning on hard failures.

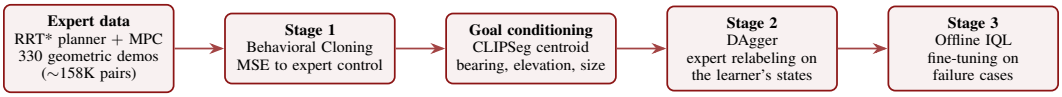


Figure 2: Training pipeline. We first imitate a geometric expert, then add centroid-based goal conditioning, use DAgger to correct distribution shift, and evaluate offline IQL fine-tuning on collected failure cases.

Design rationale. The stages are ordered by sample efficiency and deployment risk. Behavioral cloning provides a stable initialization from inexpensive synthetic demonstrations, while centroid conditioning injects semantic goal information without increasing model size. DAgger then corrects the policy on the states it actually visits, which is safer and cheaper than online reinforcement learning on a physical drone. Offline IQL is used last to test whether the collected failures contain enough reliable signal to teach recovery behavior without additional rollouts.

Problem formulation. We model navigation as a discrete-time Markov decision process. The underlying state is $x_k = (\boldsymbol{p}_W, \boldsymbol{v}_W, \boldsymbol{q}_{BW}, \boldsymbol{\omega}_B)$, where $\boldsymbol{p}_W$ and $\boldsymbol{v}_W$ are world-frame position and velocity, $\boldsymbol{q}_{BW}$ is the body orientation, and $\boldsymbol{\omega}_B$ denotes body rates. The policy does not receive privileged geometric target information. Instead, it observes $o_k = (\mathcal{I}_k, b_k, e_k, s_k, h_k)$, consisting of an encoded RGB frame $\mathcal{I}_k$, centroid features for target bearing b_k, elevation e_k, apparent size s_k, and a short history h_k of recent observations and actions. The action $u_k = (f_{\text{th}}, \boldsymbol{\omega}_B^{\text{cmd}})$ specifies collective

thrust and three body-rate setpoints. The simulator advances the system through discrete quadrotor dynamics $x_{k+1} = f_d(x_k, u_k)$ inside the Gaussian Splatting scene.

Behavioral cloning. The initial dataset is generated with an RRT* planner tracked by a model predictive controller, ensuring that the demonstrations reach the queried objects without collisions. It contains 330 expert trajectories across three targets, which become approximately 158K observation-action pairs after fourfold pose-randomization augmentation. The policy uses SqueezeNet visual features (Iandola et al., 2016), drone state information, and temporal history as inputs to a lightweight commander head with two hidden layers of width 100. We train this head by mean-squared error to the expert action:

$$\mathcal{L}_{\mathrm{BC}}(\theta) = \mathbb{E}_{(o,\, u^\star)\sim\mathcal{D}_{\mathrm{BC}}}\left[\, \left\| \pi_\theta(o) - u^\star \right\|_2^2 \right]. \tag{1}$$

Centroid-based goal conditioning. A vision-only policy must know which object is being queried. We obtain this information from CLIPSeg (Lueddecke and Ecker, 2022), which produces a semantic similarity heatmap $\boldsymbol{H}$ for the text prompt relative to the image currently seen by the drone. We threshold the heatmap at the 75th percentile, compute the intensity-weighted centroid of the activated region, and convert it into normalized image coordinates:

$$\begin{aligned} M &= \big\{(x,y) \;\big|\; \boldsymbol{H}(x,y) > P_{75}(\boldsymbol{H})\big\}, \\ c_x &= \frac{\sum_{(x,y)\in M} x\, \boldsymbol{H}(x,y)}{\sum_{(x,y)\in M} \boldsymbol{H}(x,y)}, \qquad c_y = \frac{\sum_{(x,y)\in M} y\, \boldsymbol{H}(x,y)}{\sum_{(x,y)\in M} \boldsymbol{H}(x,y)}, \\ b &= 2\, c_x / W - 1, \qquad e = 2\, c_y / H - 1. \end{aligned} \tag{2}$$

Figure 3 illustrates the construction. The heatmap localizes the most semantically relevant region for the query, the weighted centroid summarizes that region, and the displacement from the image center becomes a bearing-elevation cue for the commander. The activated area is also used as a crude apparent-size feature. In preliminary experiments, adding this simple goal condition substantially improved learning while adding no trainable parameters.

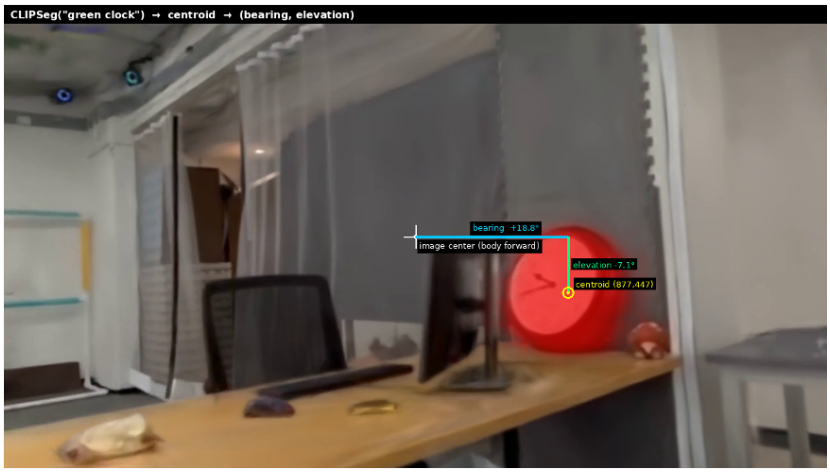


Figure 3: Centroid-based goal conditioning. CLIPSeg produces a similarity heatmap for the query "green clock." We threshold the heatmap, compute its weighted centroid, and convert the offset from the image center into target bearing and elevation features.

DAgger. Goal conditioning improves target acquisition but does not remove distribution shift. We therefore roll out the learned policy in simulation and query the MPC expert for the action it would take at each visited state. These relabeled samples are aggregated with the previous dataset and used to retrain the commander. During DAgger, the vision encoder remains frozen and only the commander head is updated. We run twelve aggregation rounds and keep the best validation checkpoint, preventing later rounds from overwriting a stronger policy.

Offline IQL fine-tuning. We then test whether offline RL can recover from hard failures. The dataset contains seventeen pilot episodes on difficult branches, of which four succeed, eleven collide, and two miss the target. We also collect seventeen successful expert episodes on the same branches, giving the learner successful demonstrations for the same difficult starts. To make the reward easier to interpret, we use a sparse terminal reward with a small dense shaping term:

$$r_k = \begin{cases} +1000, & \text{if the episode ends in success,} \\ -1000, & \text{if the episode ends in collision,} \\ -1 + 0.5\, v_k, & \text{otherwise,} \end{cases} \qquad v_k = \begin{cases} 1, & \text{if the target is visible in frame,} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The terminal values are deliberately on the order of 10^3: trajectories typically last around one hundred steps, so the accumulated step penalty is only on the order of 10^2. This scale makes success clearly preferable and collisions clearly catastrophic. The -1 per-step term encourages faster convergence to the goal, while the $+0.5$ visibility bonus encourages the drone to keep the queried object in view during flight. IQL fits a value function by expectile regression and an action-value function with a temporal-difference target:

$$\begin{aligned} \mathcal{L}_V(\psi) &= \mathbb{E}_{(s,a)\sim\mathcal{D}}\big[\, L_2^\tau\big(Q_{\hat{\theta}}(s,a) - V_\psi(s)\big)\,\big], \quad L_2^\tau(u) = \big|\tau - \mathbb{1}(u<0)\big|\, u^2, \\ \mathcal{L}_Q(\theta) &= \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\Big[\, \big(r + \gamma\, V_\psi(s') - Q_\theta(s,a)\big)^2 \,\Big]. \end{aligned} \tag{4}$$

The actor is then updated by advantage-weighted regression,

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[\, \exp\big(\beta\,[\,Q_\theta(s,a) - V_\psi(s)\,]\big)\, \big\|\pi_\phi(o) - a\big\|_2^2 \,\Big]. \tag{5}$$

We use expectile $\tau = 0.9$ and advantage temperature 0.1. We compare two variants: one updates the full actor for 100K steps, and the other updates only the output layer for 10K steps. This comparison measures how far the policy can move from its imitation initialization before generalization degrades.

4 Experimental Setup

All experiments are performed in the SOUS VIDE Gaussian Splatting simulator (Low et al., 2024; Kerbl et al., 2023). We evaluate three target objects consisting of a clock, a leafblower, and a set of boxes. The evaluation is designed to answer whether the drone reaches the queried object, whether it avoids damaging collisions, and whether the learned behavior transfers beyond the exact branches used for expert supervision.

Evaluation protocol and metrics. We evaluate in two regimes. The seen regime uses trajectory branches available during training, with fifty rollouts per object. This setting is still nontrivial because many collision-free action sequences can connect the same start and goal, and small deviations can lead to different visual observations. The unseen regime launches the drone from new start poses, with eleven rollouts per object, to measure robustness beyond the exact starts used for supervision. A rollout is counted as a success when the queried target remains in the final field of view and the drone ends within 2 m of the target. It is counted as a collision when the drone strikes the scene; otherwise, it is categorized as a goal-out-of-sight or timeout failure. We also report final distance to the goal, which separates near misses from failures in which the policy navigates to the wrong region. Success rate is the primary task metric, collision rate is the primary safety metric, and final distance is used as a continuous diagnostic.

Architecture and training details. The commander is intentionally small for onboard use. SqueezeNet visual features, drone state, centroid features, and a short temporal history are concatenated into a 147-dimensional input, followed by two hidden layers of width 100 and four control outputs. Behavioral cloning is trained for 150 epochs with a held-out validation split. DAgger uses twelve aggregation rounds, ten epochs per round, learning rate 2×10^{-5}, a frozen vision encoder, and best-checkpoint rollback. Offline IQL uses the hard-branch dataset described above and the two fine-tuning budgets introduced in the method section.

5 Results

Our evaluation supports two conclusions. First, goal-conditioned imitation learning is highly effective because it raises success to nearly 90% and reduces collisions to single digits. Second, offline RL

improves targeted failure cases but does not improve the global policy, because the logged failures are too biased. We present the quantitative results first and then use qualitative examples to explain the remaining failures.

5.1 Quantitative Evaluation

Table 1 compares the four main policy variants on seen test flights. Plain behavioral cloning without centroid features reaches only 36.7% success and collides in 21.3% of rollouts. DAgger without centroid features improves success to 52.3%, but the largest gain comes from goal conditioning: centroid behavioral cloning reaches 80.7% success before any on-policy aggregation. Combining centroid features with DAgger gives the best policy, with 88.0% success, 8.0% collisions, and an average final goal distance of 1.65 m.

Table 1: Benchmark across the four configurations on seen test flights, with fifty rollouts per object at a fixed seed. The best value in each row is shaded.

Metric	Pre-Centroid BC	Pre-Centroid DAgger	Centroid BC	Centroid DAgger
Success rate	36.7%	52.3%	80.7%	**88.0%**
Collision rate	21.3%	∼20%	13.3%	**8.0%**
Goal distance	3.71 m	∼2.5 m	1.84 m	**1.65 m**

Table 2: Per-object performance of the best policy, centroid features plus DAgger. Performance is strong across all three targets; boxes remain the most collision-prone due to tighter surrounding geometry.

Object	Success	Collision	Runs
Clock	**90%**	8%	50
Leafblower	84%	**4%**	50
Boxes	**90%**	15%	41

Table 2 shows that the improvement is not driven by a single easy object. The clock and boxes reach 90% success, while the leafblower reaches 84%. The outcome breakdown in Figure 4 shows the same trend: centroid conditioning both increases the success region and reduces the failure mass, especially the goal-out-of-sight and timeout failures that dominate the pre-centroid policies.

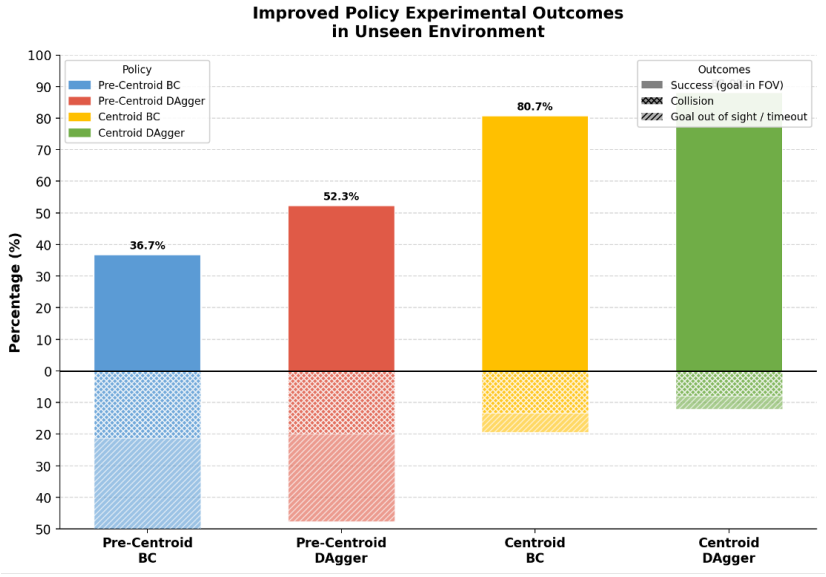
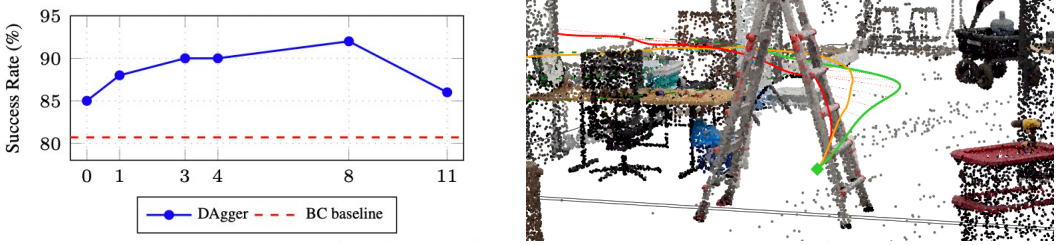


Figure 4: Outcome breakdown across policy variants. Bars above the axis indicate success; hatched regions below the axis separate collisions from goal-out-of-sight or timeout failures.

Figure 5 illustrates the effect of DAgger. Success rises above the behavioral-cloning baseline, peaks at 92% on iteration eight, and later declines toward 86%. This degradation suggests that continued aggregation can overfit the particular relabeled states, which motivates the best-checkpoint rollback. The trajectory visualization shows the practical effect: from the same start state, the DAgger policy recovers around the ladder and returns toward the target, whereas the behavioral-cloning policy drifts away from the intended route.



(a) DAgger success rate across iterations.

(b) DAgger correction in the scene.

Figure 5: DAgger improves robustness by correcting the states induced by the learned policy. The best policy occurs at iteration eight, before later aggregation begins to degrade performance.

Offline IQL on hard branches. Table 3 evaluates IQL only on the seventeen hard branches used for fine-tuning, which are precisely the branches on which the policy is more likely to fail. On this restricted set, full-actor IQL increases success from 35.3% to 64.7% and reduces collisions from 47.1% to 29.4%. The output-layer-only variant also improves over the DAgger baseline, though by a smaller margin. These results show that the offline dataset contains useful local recovery information for common failure cases.

Table 3: Offline IQL fine-tuning evaluated on the seventeen hard training branches. Full-actor IQL recovers many cases that the DAgger baseline fails.

Model	Success	Collision	Runs
DAgger baseline	35.3%	47.1%	17
IQL, full actor	**64.7%**	**29.4%**	17
IQL, output layer only	52.9%	41.2%	17

However, the same fine-tuned policy drops to roughly 70% success on the broader unseen benchmark, well below the 88% DAgger policy. The improvement is therefore local rather than general. The qualitative analysis below explains why many hard-branch failures are driven by incorrect or missing target detections, so fine-tuning on them can reinforce misleading behavior.

5.2 Qualitative Analysis

The remaining failures are dominated by perception. CLIPSeg produces a relative similarity map for every frame, even when the queried object is absent. The policy is trained to move toward the strongest activation; therefore a false positive becomes an actionable target rather than a harmless detector error. Figure 6 shows three representative cases.

In the first case, CLIPSeg activates on an object that is not the true target, and the drone follows that spurious region. In the second, the queried green clock is not initially visible, so a visually similar green spray is detected instead and the policy navigates confidently to the wrong object. In the third, the target is visible at first but later becomes occluded by a ladder; once the detector loses the target, the policy latches onto another region and drifts away. These examples show that the controller often behaves sensibly given the target signal it receives, while the failure comes from the target signal itself being wrong.

(a) False-positive perception. (b) Target never seen. (c) Loss of sight.

Figure 6: Perception-driven failure modes. CLIPSeg can activate on false positives, substitute a visible distractor when the true target is absent, or lose the object after occlusion.

The expert introduces a second limitation. As shown in Figure 7, the RRT* planner sometimes takes unnecessarily indirect paths. These trajectories are geometrically valid, but they are not necessarily natural or perception-aware. A learned controller trained on such demonstrations may inherit unnecessary detours, especially when a more direct visual route is available.

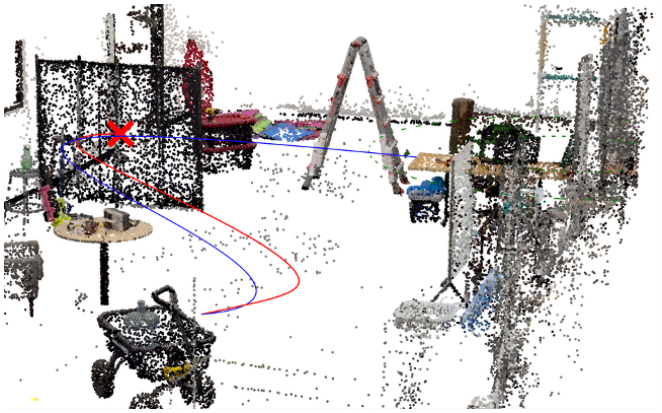


Figure 7: Structural limitation of the geometric expert. The RRT* path is collision-free but unnecessarily indirect, giving the learner supervision that is physically valid but not perception-aware.

These observations explain why IQL improves the hard branches but hurts the broader benchmark. Offline RL can only improve a policy when the dataset encodes the desired behavior. In our hard cases, many trajectories are generated under ambiguous or incorrect perception. Fine-tuning on them teaches the policy to commit more strongly to biased target signals, which helps some memorized branches but reduces generalization.

6 Discussion

The main lesson is that the bottleneck shifts as the policy improves. Behavioral cloning initially fails because it lacks a target-conditioned visual signal and is vulnerable to distribution shift. Centroid features and DAgger address these problems directly, raising success to nearly 90%. After that point, most remaining failures are caused by perception because the policy is often given a confident but incorrect target location.

This has two implications. First, offline RL is not automatically beneficial. In this project, IQL extracts useful behavior from the hard-branch dataset, but that dataset is biased by the same perception errors that caused the failures. The resulting policy overfits local recovery patterns and loses global robustness when evaluated on the broader benchmark. This does not mean that offline RL is inappropriate for language-conditioned flight; it means that the usefulness of offline RL is bounded by the quality, coverage, and perceptual correctness of the logged data.

Second, future improvements should prioritize target verification and uncertainty. The policy should distinguish between tracking a visible target and searching when the target is absent or occluded, because these are different control problems with different risk profiles. A multi-task search-versus-track policy would make this distinction explicit. It could track confidently when the perception stack provides calibrated evidence that the queried object is present, and switch to cautious exploration when the visual evidence is weak, ambiguous, or inconsistent across time. Such a multi-task formulation is meaningful only if the perception module can first estimate whether the target is actually present, rather than always returning the most similar region in the image.

This framing also clarifies the role of the methods tested here. Goal-conditioned imitation with DAgger provides a simple, stable, and interpretable baseline for vision-only navigation, while offline IQL is better viewed as a tool whose success depends on the structure of the recovery dataset. The next stage should therefore combine perception-aware demonstrations, calibrated target confidence, and a multi-task policy that can choose between tracking and searching instead of committing blindly to every heatmap activation.

7 Conclusion

We improved SINGER in both sample efficiency and navigation performance. In the original training regime, reaching roughly 80% success requires a much larger behavioral-cloning dataset, on the order

of one million observation-action pairs. In this project, we restrict training to approximately 158K pairs, under which the vanilla controller reaches only about 30% success. Adding centroid-based goal conditioning raises success to 80.7% on the same smaller data regime, showing that the policy was missing a simple but essential semantic navigation cue. DAgger then improves robustness by training on the states induced by the learned policy, bringing success close to 90% and reducing collisions to single digits.

Offline IQL recovers targeted failures but does not improve the general policy, because the failure dataset is narrow and perception-biased. The main conclusion is therefore diagnostic as much as algorithmic. Once the policy receives a reliable target cue and is trained on its own state distribution, perception becomes the limiting factor. Future work should focus on robust target verification, perception-aware demonstrations, and a multi-task search-versus-track policy that can switch between confident tracking and safe exploration when the queried object is not visible.

References

Adang et al. SINGER: Semantic In-situ Navigation and Guidance for Embodied Robots. *arXiv preprint arXiv:2509.18610*, 2025.

Low et al. SOUS VIDE: Cooking Visual Drone Navigation Policies in a Gaussian Splatting Vacuum. *arXiv preprint arXiv:2412.16346*, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics (SIGGRAPH)*, 2023.

Dean A. Pomerleau. ALVINN: An Autonomous Land Vehicle in a Neural Network. *Advances in Neural Information Processing Systems*, 1988.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. *AISTATS*, 2011.

Timo Lueddecke and Alexander S. Ecker. Image Segmentation Using Text and Image Prompts. *CVPR*, 2022.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline Reinforcement Learning with Implicit Q-Learning. *ICLR*, 2022.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating Online Reinforcement Learning with Offline Datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient Online Reinforcement Learning with Offline Data. *ICML*, 2023.

Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level Accuracy with 50x Fewer Parameters and less than 0.5MB Model Size. *arXiv preprint arXiv:1602.07360*, 2016.

Kim et al. CARE: Confidence-Aware RGB Obstacle Reasoning for Embodied Navigation. *arXiv preprint arXiv:2506.03834*, 2025.

Ye et al. DreamZero: Temporal Prediction for Visual Navigation. *arXiv preprint arXiv:2602.15922*, 2026.