Id	Objectives	Task Persistance * Denotes questions, where designer may opt to answer sub-questions to address concern	Response	Example Response
ASK 1	Accuracy	**Does the RL agent's design accurately reflect real-world dynamics? ***		
ASK 1.1	Accuracy	How does the learning process incorporate domain knowledge or physical constraints to improve alignment with reality?		Physical constraints, such as maximum velocity, acceleration limits, and actuator capabilities, are enforced through hard constraints within the RL formulation, preventing unrealistic actions during training. Domain knowledge is integrated via expert-designed reward functions that guide the agent toward desirable behaviors, penalizing unsafe or inefficient actions. Additionally, the system leverages imitation learning from expert demonstrations, allowing the agent to learn from real-world trajectories before fine-tuning through reinforcement learning.
ASK 1.2	Accuracy	Could the current reward design inadvertently incentivize undesired behaviors? What measures are in place to identify and mitigate such risks?		The reward function prioritizes efficient pathfinding which might lead the agent to fly too aggressively, taking shortcuts that lead to unsafe behaviors like flying too close to obstacles. To mitigate this, a penalty is applied for actions that bring the drone within a certain distance of obstacles, ensuring safety is prioritized. Additionally, a smoothness term is added to the reward function to discourage erratic movements, promoting more controlled flight paths.
ASK 1.3	Accuracy	If multiple agents are used, how does the reward structure balance global and local objectives to accurately attribute actions to individual agents?		Each agent receives a local reward for completing its assigned task efficiently (i.e. delivering an item to a designated location within a specific time frame). Simultaneously, a global team reward is given based on the overall throughput of the system (i.e. the total number of items delivered without collisions or bottlenecks). To ensure effective collaboration, a credit assignment mechanism which calculates each agent's contribution to the global reward by isolating the impact of its actions on the team's overall performance.
ASK 2	Accuracy	**How is the learning architecture optimized for the available computational resources? ***		
ASK 2.1	Accuracy	If applicable, how does the system handle high-dimensional state and action spaces to balance accuracy and computational efficiency?		Autoencoders and feature extraction networks compress raw state representations into lower-dimensional latent spaces, preserving critical information while reducing computational complexity. Actor-critic architectures with shared feature representations enable efficient learning in large state-action spaces.
ASK 2.2	Accuracy	If operating in real-time, how does the system ensure timely decision-making to meet operational constraints without compromising accuracy?		The policy network is optimized using neural architecture search (NAS) and quantization techniques to reduce inference latency while maintaining decision quality. Multi-threaded execution and GPU acceleration enable fast computations, allowing the agent to process high-dimensional inputs with minimal delay. Additionally, hierarchical decision-making is used, where high-frequency low-level actions are handled by precomputed controllers or fast heuristic policies, while higher-level planning occurs at a lower frequency to balance speed and accuracy. The system also incorporates early-exit strategies, where low-uncertainty decisions are executed immediately, while more complex scenarios invoke additional processing only when necessary.
ASK 3	Accuracy	**How is accuracy assessed in the system?**		
ASK 3.1	Accuracy	What metrics are used to evaluate performance and how are acceptable ranges for them chosen?		Task completion accuracy measures the percentage of correctly executed tasks, with an acceptable range of ≥98% based on historical human performance benchmarks. Pick-and-place precision is assessed using the Mean Absolute Error (MAE) of item placements, with an acceptable deviation of ≤2 cm to ensure correct item handling. Navigation accuracy is measured by the Deviation from Optimal Path (DOP), where the agent's route must stay within 5% of the shortest possible path to maintain efficiency. These thresholds are determined through warehouse operational standards and expert-defined benchmarks.
ASK 3.2	Accuracy	Are there validation procedures to compare RL-generated actions with those of expert human operators or traditional controllers?		This is done using Action Agreement Rate (AAR), measuring the percentage of RL decisions that align with expert actions, with an acceptable threshold of ≥90%. Additionally, Task Efficiency Ratio (TER) compares RL task completion times to those of human operators, ensuring the agent performs within ±5% of expert efficiency. Finally, Deviation from Optimal Policy (DOP) is used to assess how closely the RL agent's actions follow those of traditional warehouse controllers, with a target of ≤3% deviation to ensure alignment with best operational practices.
ASK 4	Availability + Quality of Data, Robustness/Generalizability	**Have edge cases, rare scenarios, or perturbations been identified and prioritized during training/testing? ***		
ASK 4.1	Availability + Quality of Data, Robustness/Generalizability	How are variations in the environment's dynamics (e.g., changes in weather, unexpected obstacles) modeled during training and testing?		During training, a domain randomization technique is used to simulate diverse conditions, such as varying weather (rain, fog, snow), lighting (day, night, twilight), and road conditions (wet, icy, or uneven surfaces). Additionally, unexpected obstacles, like pedestrians or debris, are dynamically introduced into the simulation to ensure the agent learns robust avoidance strategies.
ASK 4.2	Robustness/Generalizability, Availability + Quality of Data	How are high-risk or low-probability scenarios identified and prioritized into testing and replay strategies?		The system employs a prioritized experience replay (PER) mechanism, where transitions are sampled based on their associated temporal-difference (TD) error, which often highlights challenging or novel situations. To further emphasize edge cases, high-risk scenarios—such as near-collision events, sudden wind gusts, or GPS signal loss—are tagged during training and given a higher priority in the replay buffer. Additionally, a separate buffer is maintained specifically for rare or critical events, ensuring that these scenarios are revisited frequently to reinforce the agent's ability to handle them effectively.
ASK 5	Robustness/Generalizability	**How does the system maintain robust state representations despite uncertainties or limitations in observations? ***		
ASK 5.1	Robustness/Generalizability, Availability + Quality of Data	How is data augmented or adapted during training to account for sensor noise, missing inputs, or corrupted observations?		During training, sensor data is augmented with Gaussian noise to simulate inaccuracies common in real-world scenarios, such as GPS drift or camera blurring. Missing data scenarios are modeled by randomly masking inputs from specific sensors, such as temporarily disabling GPS or introducing dropout in LiDAR point clouds, to teach the agent to rely on redundant sensory modalities. Sensor failures, like a complete camera outage, are simulated in the test environment to evaluate the agent's ability to adapt by leveraging fallback mechanisms.
ASK 5.2	Robustness/Generalizability	How does the system infer missing or uncertain state information when observations are incomplete or unreliable?		A Kalman filter is used to estimate hidden states in scenarios with sensor noise or dropped data, providing a more reliable belief of the system's current condition. For high-dimensional or multi-sensor inputs, Bayesian state estimation integrates uncertainty-aware models to infer likely states when observations are ambiguous.
ASK 6	Robustness/Generalizability	**How is the decision-making process designed to handle uncertainty and inconsistencies in action execution? ***		
ASK 6.1	Robustness/Generalizability, Availability + Quality of Data	How are action execution errors, such as actuator noise and power limitations, incorporated into training and testing?		Actuator errors, such as overshooting or undershooting a target position due to mechanical inaccuracies, are simulated during training by adding stochastic noise to the executed actions. Communication delays are modeled by introducing random time lags between the policy output and the execution of actions in the environment.
ASK 6.2	Robustness/Generalizability	How does the system adapt its action selection when execution outcomes deviate from expected behavior, preventing small execution errors from compounding into unstable behaviors?		To adapt in such cases, the system employs an action correction mechanism using a model-predictive control (MPC) layer that recalibrates actions in real-time based on updated sensor feedback.
ASK 7	Robustness/Generalizability	**How does the system ensure that rewards provide reliable and informative feedback to guide learning effectively? ***		
ASK 7.1	Robustness/Generalizability	How does the system ensure stable and efficient learning when rewards are sparse or inconsistently observed during training?		Reward shaping introduces intermediate rewards based on domain knowledge, guiding the agent toward long-term goals without compromising optimality. Intrinsic motivation mechanisms, such as curiosity-driven exploration using prediction error or information gain, encourage the agent to explore meaningful states even when external rewards are infrequent.
ASK 7.2	Robustness/Generalizability	How does the system mitigate the impact of noise or corruption in the reward signal to improve learning outcomes?		To address noisy rewards, a smoothing filter (e.g., exponential moving average) is applied to reward signals during training.
ASK 7.3	Robustness/Generalizability	How are delays in reward signals accounted for to prevent misalignment between actions and long-term outcomes?		Eligibility traces (λ-return TD learning) help propagate reward information across multiple time steps, allowing actions taken earlier in an episode to receive proper credit.
ASK 8	Robustness/Generalizability	**Have differences between the simulated and real environments been identified and addressed? ***		
ASK 8.1	Robustness/Generalizability	What strategies have been designed to incor… refine the model and reduce the sim2real gap?		Sensor data from real-world trials, such as force-torque readings and motion trajectories, were used to update the parameters of the simulated objects and dynamics. We also leveraged Bayesian optimization to tune the physics engine parameters (e.g., friction coefficients, damping factors) to better align simulated outcomes with real-world observations
ASK 8.2	Robustness/Generalizability	What techniques have been employed to transfer policies or adapt models developed in simulation to perform effectively in real-world environments?		We used domain adaptation techniques like feature alignment networks to ensure the sensory input distributions between simulation and reality were matched, allowing the agent to interpret real-world data seamlessly.

ASK 9	Robustness/Generalizability, Availability + Quality of Data	**In multi-agent settings, are failures of other agents simulated during training to improve resilience in cooperative and competitive settings?**		Failures of other agents are introduced during training through domain randomization and adversarial testing technique and agents are trained to reroute their paths dynamically to avoid congestion caused by an immobilized robot or take over tasks that a failed robot could not complete. During testing, stress scenarios explicitly include multi-agent failures, and the system's ability to maintain operational efficiency and avoid bottlenecks under such conditions is evaluated.
ASK 10	Availability + Quality of Data, Robustness/Generalizability	**How does the system improve training efficiency to maximize learning and generalizability from limited data?**		Off-policy learning methods, such as Soft Actor-Critic (SAC) or Q-learning with experience replay, enable the agent to reuse past experiences rather than relying solely on fresh interactions. Meta-learning (e.g., MAML) is employed to accelerate adaptation to new tasks by training the model on a distribution of environments.
ASK 11	Robustness/Generalizability	**How are hyperparameters chosen to balance stability, generalization, and performance across different environments?**		Bayesian optimization and Population-Based Training (PBT) are used to explore the hyperparameter space efficiently, adapting values dynamically based on performance metrics. Hyperparameters are validated through cross-environment evaluation, where models are tested on unseen variations of the environment to ensure adaptability without overfitting to specific conditions.
ASK 12	Robustness/Generalizability	**What metrics are used to evaluate robustness and how are acceptable thresholds for them determined?**		Generalization Error quantifies the agent's performance drop when exposed to unseen scenarios, with an acceptable degradation of ≤3% compared to training conditions. Performance Degradation Under Perturbation evaluates task efficiency under perturbed conditions (e.g., dynamic obstacles, varying load weights) and must remain within ±7% of nominal performance to guarantee stability. Thresholds are determined based on historical warehouse failure rates, expert-defined tolerances, and stress-test simulations.
ASK 13	Robustness/Generalizability, Accuracy	**How do the chosen approaches balance the performance-reliability trade-off, ensuring that higher performance does not compromise reliability and vice-versa?**		Regularization methods, like dropout and L2 weight decay, are used to prevent overfitting, ensuring that the model maintains generalization across diverse environments without sacrificing reliability. At the same time, performance-enhancing strategies like curriculum learning and reward shaping are employed to boost task efficiency and task completion speed, while maintaining a safety layer that temporarily overrides high-risk actions.
ASK 14	Validity	**Are formal verification methods used to ensure the correctness and safety of the RL system? If so, which specific methods are applied?**		Model checking is employed to formally verify that the drone's decision-making processes respect critical safety properties, such as maintaining safe altitude and avoiding obstacles. This is done by exhaustively checking the policy against a set of formal specifications for safe flight paths. Additionally, reachability analysis is used to verify that the drone can always reach a safe state (e.g., returning to a safe landing zone in case of a failure or sudden obstacle), even when faced with noisy sensor inputs or sudden environmental changes, such as gusts of wind.
ASK 15	Safety	**What constraints or penalties are incorporated into the training process to ensure safety?**		Collision avoidance penalties are applied whenever the drone gets too close to obstacles or buildings, encouraging it to maintain a safe distance. To further ensure safety, no-fly zones (e.g., areas near airports or restricted zones) are integrated into the reward function, with heavy penalties for any attempt to enter these areas. Emergency landing rewards are added, promoting the drone to seek safe landing zones if its battery is running low or if critical sensor failures occur.
ASK 15.1	Safety	If multiple constraints or penalties are used, how are they prioritized or managed in situations where they might conflict?		Collision avoidance is given the highest priority, with a strong penalty for any proximity to obstacles, as this directly impacts the drone's safety. No-fly zone penalties are given lower priority but are enforced in all cases to comply with regulations. To handle conflicts, a hierarchical penalty system is employed where violations of higher-priority constraints (e.g., collisions) override lower-priority penalties (e.g., altitude or minor boundary deviations).
ASK 15.2	Safety	How are the constraints or penalities balanced with the reward so the agent is not tempted toward a higher reward at the cost of safety?		A safe exploration bonus is incorporated, rewarding the drone for taking conservative, safe routes that avoid high-risk areas, even if they may result in slightly slower task completion times. The reward structure is designed to emphasize long-term safety by gradually adjusting the learning process so that safety concerns are prioritized first, and unsafe behaviors are discouraged even if they offer short-term rewards.
ASK 16	Safety	**What mechanisms ensure safety during execution and in response to failures or unexpected scenarios?**		
ASK 16.1	Safety	What predefined safety constraints are implemented in the system to prevent unsafe actions during execution?		Altitude limits are used to ensure the drone remains within a safe operating height range, above obstacles and within regulatory requirements. No-fly zones are defined around sensitive areas like airports or restricted zones, and the drone is not allowed to enter these areas under any circumstances.
ASK 16.2	Safety, Security	What fallback systems, predefined controllers, or recovery strategies are implemented to address situations where the RL agent fails, behaves unpredictably, or enters an unsafe state? (note: may use ASK 8.2.1 to partially address this, but elaborate on any other strategies that do not involve human intervention)		Inverse kinematics-based controllers are used to take over in case the RL agent's planned trajectory leads to an unsafe configuration, such as an unintended collision with nearby objects or workers. If the robot starts behaving unpredictably, such as making erratic movements or deviating from its task path, recovery strategies like trajectory smoothing or position correction are applied to bring the robot back to a safe, predefined state. Additionally, the system employs graceful degradation, where if certain sensors (like vision or force sensors) fail, the robot switches to a fallback mode that relies on redundant sensors (e.g., using encoders and accelerometers instead of cameras) and slows down its operations
ASK 16.2.1	Safety	What mechanisms are available for human intervention, and under what circumstances is intervention allowed or expected?		Emergency stop buttons are placed within the workspace, allowing operators to immediately halt the robot's operation if an unsafe situation arises, such as an unexpected collision or if the robot deviates from its designated path. Additionally, a manual override mode is provided, which allows the operator to take control of the robot via a joystick or remote interface, particularly in cases where the robot's movements become erratic or if it encounters unexpected obstacles that require human judgment to resolve. Intervention is expected when the system detects abnormal conditions, such as a significant drop in task accuracy or if safety protocols (like proximity to human workers) are breached.
ASK 16.3	Safety, Security	How does the system detect out-of-distribution (OOD) inputs and other anomalies during deployment?		For detecting sensor anomalies, the system uses autoencoders trained on normal sensor data (such as force, torque, and visual input). To detect OOD visual inputs, the system uses CNN-based OOD detector which measures the confidence of the CNN output and flags inputs with low confidence as potentially out-of-distribution.
ASK 16.4	Safety	How does the system handle failures in other agents within the multi-agent environment to ensure its own safety and prevent cascading failures?		The drones rely on real-time communication to detect issues with other drones and reroute to avoid the failed drone's location. Task redistribution ensures that the delivery mission is reassigned to other drones, maintaining operational efficiency.
ASK 17	Safety	**How does the system validate and manage safety during ongoing adaptation and exploration in deployment? ***		
ASK 17.1	Safety	If the system is designed to adapt further during deployment, how are the adapted policies validated for safety? Is there a mechanism to revert to a baseline policy in the event of a failure in the adapted policy?		If the adapted policy leads to behaviors that breach predefined safety constraints, it is flagged as unsafe. Additionally, if the policy results in increased risk of collisions between drones, fails to adapt appropriately to environmental changes (like sudden weather conditions), or causes drones to deviate from optimized paths in unsafe ways (such as flying too close to buildings or other drones), it is considered unsafe. When such issues are detected, the system automatically triggers a rollback mechanism that reverts to the baseline policy. This can occur through automatic shutdown of the adapted policy, followed by reinitializing the baseline control system, or if necessary, a manual override can be performed by an operator.
ASK 17.2	Safety	How does the system manage the trade-off between exploration and safety, particularly during adaptation in a deployment environment?		The system uses a conservative exploration strategy where drones are only allowed to explore new behaviors or routes within safe boundaries, ensuring that exploration doesn't jeopardize safety. Additionally, adaptive risk functionsa are used that dynamically adjust exploration rates based on the drone's confidence in its current policy, increasing exploration only when safety can be guaranteed, and reducing it when uncertainty is high
ASK 18	Safety	**How does the system model and quantify uncertainty in decision-making, and how are uncertainty estimations or confidence intervals used to enhance reflect limitations and enhance safety?**		When a drone encounters high uncertainty—such as detecting ambiguous objects through its vision system or operating in complex weather conditions—it increases the safety margin by adopting more cautious behavior, like reducing speed or increasing distance from potential hazards. Uncertainty estimates are used to adjust the exploration-exploitation trade-off, ensuring that exploration is minimized in high-risk situations and the drone prioritizes safe actions.
ASK 19	Safety	**What metrics are used to evaluate safety (e.g. constraint violation rate) and how are acceptable thresholds for these metrics determined?**		The constraint violation rate tracks how often drones exceed predefined safety boundaries, such as altitude limits or speed limits. Acceptable thresholds for these metrics are determined based on industry safety standards. The system allows no more than a 0.1% violation rate in high-priority zones, with stricter thresholds in densely populated or critical delivery areas. Collision frequency and no-fly zone breaches are set to near-zero tolerances, while emergency stops are monitored to ensure they occur only in exceptional situations, indicating potential system failure or unsafe conditions. These thresholds are continuously refined through simulation-based testing.
ASK 19.1	Safety, Maintainability	Are these safety metrics logged during deployment? If so, what events trigger the logging, how frequently are they recorded, and how often are they reviewed to ensure continuous safety?		Constraint violation rate, collision frequency, no-fly zone breaches, and emergency stop occurrences are logged at high frequency (e.g., every second or after each action taken) to ensure precise tracking of safety-related events. The logs are then reviewed every 10 minutes if operating in highly dynamic urban areas. Otherwise, logs are monitored once per day in more predictable environments. Additionally, automated monitoring systems analyze these logs in real time to trigger alerts if the metrics exceed predefined thresholds.
ASK 20	Maintainability	**What mechanisms are in place to detect when performance/behaviors degrade to a point that requires policy updates or retraining?**		Performance degradation is detected using an LSTM network trained on historical data, which track key performance indicators, including task completion time, delivery accuracy, and safety violations. If the system detects a sustained drop in performance beyond a threshold, it triggers an alert to initiate policy updates or retraining.
ASK 21	Transparency + Explainability	**What data or metrics will be provided to the human overseer during run-time?**		The overseer will be shown no-fly zone violations, collision warnings, and emergency stop occurrences. Battery levels and sensor health status for each drone will be continuously displayed. Performance KPIs, such as delivery time and any deviations from the expected route, will be provided as well.

ASK 21.1	Transparency + Explainability, AI Expertise	How will the information be presented to align with the overseer's expertise and cognitive load?		High-priority alerts (e.g., safety violations, battery failure, or drone malfunctions) will be prominently displayed with color-coded warnings (e.g., red for urgent, yellow for caution) to ensure immediate attention. Key performance metrics, like delivery success rate and task completion times, will be shown in summary form with easy-to-read charts, such as bar graphs or trend lines, highlighting any deviations from the expected behavior.
ASK 21.2	Transparency + Explainability	What conditions or thresholds will trigger alerts, and how will the system communicate these to the overseer?		If an anomaly (see ASK 8.3), safety violation (see ASK 8.1), or sensor failure is detected, the overseer will receive immediate alerts with details of the issue. This will include a visual indicator (red, flashing alerts for critical and yellow for warnings) as well as a sound alert in the case of a critical scenario.
ASK 22	**Transparency + Explainability, AI Expertise**	**What tools or information will ensure the human overseer has an adequate expertise in the system before using the system in high-risk environment? ***		
ASK 22.1	Transparency + Explainability, AI Expertise	How will the system demonstrate examples of expected, degraded, and failure behaviors in a way that is interpretable for the overseer?		The system will provide training simulations that present various scenarios, including normal operations, minor issues (e.g., delayed deliveries, sensor malfunctions), and critical failures (e.g., collisions, emergency stops). The overseer will interact with these scenarios in real-time, receiving guidance on interpretation and response strategies through step-by-step tutorials.
ASK 23	**Transparency + Explainability**	**To what extent can the human overseer be actively involved in the training process? Are they able to provide feedback or corrections during key stages?**		During training, the overseer can annotate scenarios and review model outputs to flag suboptimal behaviors or correct agent actions.
ASK 24	**Transparency + Explainability, System Accountability**	**How will the system record and log critical interactions and decisions?**		Timestamped logs will include drone movements, safety violations, manual interventions, and any alerts or changes in system state.
ASK 24.1	Transparency + Explainability, System Accountability	Does the system enable detailed post-mortem analysis, including replaying scenarios to trace the reasoning behind decisions?		The logs will specifically include the state, action, and reward at each decision point so that decision-making can be reconstructed during post-mortem analysis.
ASK 25	**Privacy**	**Does the agent's training data include any sensitive information, such as proprietary operational data, personally identifiable information (PII), or safety-critical system logs? If so, how is privacy protected? ***		Yes, the agent may be trained on proprietary operational data, including system performance logs, sensor data, and potentially employee movement patterns.
ASK 25.1	Privacy	What security measures ensure sensitive data is securely collected, and how is it sanitized or anonymized during preprocessing?		Data is collected through secure, encrypted channels (TLS 1.2+), and any sensitive identifiers are removed or anonymized using hashing and differential privacy techniques before entering the training pipeline.
ASK 25.2	Privacy	How is sensitive data securely stored, and what access controls are in place to prevent unauthorized retrieval?		All stored data is encrypted using AES-256, and access is restricted through role-based access control (RBAC) with multi-factor authentication (MFA). Audit logs track all access requests, and regular security audits ensure compliance with data protection policies. Sensitive datasets are stored in segregated environments, with strict least-privilege access policies enforced.
ASK 25.3	Privacy	What safeguards prevent data leakage or misuse during model training and inference?		The training environment is sandboxed and monitored for unauthorized access, and federated learning is used when possible to keep sensitive data on-premises. Additionally, differential privacy techniques ensure individual data points cannot be reconstructed from model outputs, and secure multi-party computation (SMPC) is used for collaborative training without direct data exposure.
ASK 26	**Fairness**	**How are biases detected and mitigated?***		
ASK 26.1	Fairness	How does the system ensure equitable treatment across different groups during decision making?		When bias is detected, the agent applies re-weighting strategies to its learning process and incorporates fairness constraints into the objective function using Fairness Constraints Optimization (FCO), directly adjusting the reward structure to penalize unfair actions.
ASK 26.2	Fairness	What methods/metrics are used to identify biases in the agent's decision-making processes?		To identify biases, the system implements Demographic Parity as a fairness metric, ensuring that decisions result in outcomes that are statistically equal across demographic groups (e.g., task assignments are distributed proportionally to group representation). Additionally, the system uses Equalized Odds to ensure that the true positive and false positive rates for any group are equal.
ASK 27	**Fairness, Availability + Quality of Data**	**How is the diversity and representativeness of training data assessed to prevent biases or blind spots in learned behaviors?**		State and action space coverage analysis ensures that the training dataset spans the full range of expected operating conditions, avoiding overfitting to narrow scenarios. Clustering techniques and density estimation methods, such as k-means clustering or Gaussian Mixture Models (GMMs), are used to detect underrepresented regions in the data. Additionally, domain experts review critical decision boundaries to ensure real-world alignment, and active data collection strategies prioritize sampling from uncertain or poorly performing areas.