Autonomous Drone Scene Reasoning Agent

Human-Aware Hazard Detection & Shared-Path Safety

---

1. Problem Statement (locked, updated)

Problem

Autonomous drones operating in real-world environments (industrial inspection, disaster response, environmental monitoring) often fail before control — at the scene understanding and decision stage, especially when humans are present.

Current systems:

- **Detect objects but do not reason about shared physical risk**
- **Evaluate safety only for the robot, not for humans following its path**
- **React to hazards but cannot explain why a route is unsafe**
- **Lack human-aware and physics-aware guidance logic**

This leads to:

- **Unsafe navigation decisions**
- **Misleading guidance in human-assisted scenarios**
- **Poor trust in autonomy**
- **High false positives / false negatives in hazard assessment**

Objective

Build a vision-based reasoning agent that enables a drone to:

1. **Understand its environment from egocentric video**
2. **Reason about physical hazards:**
    - **Terrain**
    - **Obstacles**
    - **Environmental hazards**

   - **Human presence as a traversal constraint**
3. **Evaluate shared-path safety:**
   - **Is a locally observed path safe for the drone?**
   - **Is the same locally observed path safe for a human to follow, particularly when guidance toward a safer zone is implied?**
4. **Recommend a navigation or guidance action that accounts for both agents**
5. **Explain the reasoning behind that decision**

The system focuses on reasoning and decision support, not flight control.

Explicit Non-Goals (for judges)

- **No low-level control (PID, MPC, flight stack)**
- **No real-time autonomy requirement**
- **No global mapping, SLAM, or path planning**
- **No medical, emotional, or distress inference**
- **No generative media or simulation realism focus**

Humans are treated strictly as physical agents with traversal constraints.

---

2. Core Capabilities (system guarantees)

For each input video segment or frame window:

Inputs

- **Egocentric drone video (simulated or recorded)**
- **Optional metadata (altitude, camera angle)**

Outputs

1️⃣Hazard Assessment

- **Obstacle proximity**
- **Terrain stability / passability**
- **Environmental hazards (e.g., debris zones, plumes)**
- **Human presence as a physical constraint**

2️⃣ Dual Safety Evaluation

Drone Path Safety

- **Safe / Caution / Unsafe**

Human-Follow Path Safety

- **Safe to follow**
- **Follow with caution**
- **Unsafe for human traversal**

This explicitly distinguishes:

- **Drone-only affordances (flying over gaps, tight clearances)**
- **Human affordances (walking, balance, body clearance)**

3️⃣ Optional Exit Context (Abstract, Non-Planning)

- **Represents optional directional context when a safer zone or exit is implied**
- **Not required for path safety classification**
- **Used only to interpret guidance decisions, not to generate them**
- **Does not perform exit discovery, mapping, or route selection**

> **Path safety classification is always computed independently of exit context.**

4️⃣ Risk Map (Abstract)

- **Directional risk scores:**
  - **Forward**
  - **Left**
  - **Right**
  - **Up**

Risk scores incorporate human vs drone constraints.

5️⃣ Navigation & Guidance Recommendation

- **Proceed and guide human**
- **Proceed but do not guide human**
- **Reroute before guiding**
- **Hold position**

Guidance decisions are made step-by-step.

> **Shared-path safety classification always precedes any consideration of exit or goal direction.**

6️⃣ Textual Reasoning Explanation

Clear, physically grounded explanations such as:

> **“While the drone can safely pass over the debris field, the uneven terrain presents a tripping hazard for a human. The recommended action is to reroute before guiding a person through this area.”**

---

3. Recipe-Driven Architecture (explicit mapping)

Judges should clearly see how each Cosmos Cookbook recipe is used.

🧠 1. Egocentric Social & Physical Reasoning (Core AI Reasoning)

Link:
https://nvidia-cosmos.github.io/cosmos-cookbook/recipes/inference/reason2/intbot_showcase/inference.html

📌 What this recipe *really* is

It demonstrates using **Cosmos Reason 2 in inference mode** to reason about egocentric video from a robot’s POV (e.g., gestures, spatial relationships, risk, social context) and produce structured understanding results. It focuses on embodied perception and robot-centric reasoning rather than just pixel recognition.

💡 How your project uses it (explicitly)

- You take *egocentric drone video* as input
- You feed it to **Cosmos Reason 2** to extract:
  - Hazards
  - Human presence
  - Spatial relationships
  - Traversability cues
- You apply this at every frame (or short window)

This is the **primary reasoning engine** of your system.

📌 Your usage is identical in structure and goal to this recipe — but applied to **physical safety decisions** rather than social gestures.

Judge-ready text (copy/paste):

> "We use the *Egocentric Social & Physical Reasoning* recipe from the Cosmos Cookbook to perform embodied, robot-centric reasoning over egocentric video. The agent queries Cosmos Reason 2 to extract hazards, human presence, and spatial context for physical safety evaluation."

🔄 2. Intelligent Transportation Post-Training (Optional / NOT used)

Link:
https://nvidia-cosmos.github.io/cosmos-cookbook/recipes/post_training/reason2/intelligent-transportation/post_training.html

📌 What this recipe *really* is

This shows how to **fine-tune Cosmos Reason 2 via supervised learning** on a labeled traffic dataset to improve task accuracy on problems like traffic scene understanding and pedestrian VQA. It's domain-specific post-training tailored to a dataset.

❌ How your project *does not* use it

You **do not post-train** for two reasons:

1. **General hazard reasoning** — You want open-ended physical reasoning; fine-tuning would bias the model toward specific labeled hazards.
2. **Scope and simplicity** — The challenge allows standalone inference (and you are not required to fine-tune). This keeps reasoning general and interpretable.

Judge-ready text (copy/paste):

> “We do *not* use the Intelligent Transportation post-training recipe. Post-training biases Reason 2 to a specific labeled domain (e.g., traffic), which conflicts with our requirement for open-ended physical hazard reasoning and interpretability.”

📏 3. Physical Plausibility Prediction (Contextual inspiration only)

Link:
https://nvidia-cosmos.github.io/cosmos-cookbook/recipes/post_training/reason1/physical-plausibility-check/post_training.html

📌 What this recipe *really* is

This showcases how to use Cosmos Reason 1 to **score physical plausibility** of synthetic videos on a scale with physics criteria (gravity, continuity, motion consistency). It is primarily a *benchmark* for physical realism in generated video.

🧠 How your project *uses the idea*

You **do not fine-tune** using this recipe, but you *follow its insights* on how to evaluate physical plausibility. Specifically:

- Your risk and plausibility module uses semantic cues (e.g., “gap”, “uneven ground”) and reasoning constraints, not a physics score
- You enforce physics constraints via rule logic and human vs drone affordances

Judge-ready text (copy/paste):

> “Our *physical plausibility logic* is inspired by the Physical Plausibility Prediction recipe, which demonstrates how to judge consistency with physical laws. We adopt

> this principle in our reasoning chain by applying constraint logic rather than numeric physics scoring.”

---

4. Video Search & Summarization (Alternate inference pattern)

Link: https://nvidia-cosmos.github.io/cosmos-cookbook/recipes/inference/reason2/vss/inference.html

📌 What this recipe *really* is

This shows using Cosmos Reason 2 for video summarization and search, reasoning about key moments or events in a video — essentially *video analytics*.

💡 How your project relates

Your agent is also a **video analytics system**, but with decision reasoning instead of search/summarization. You ingest a video and generate *hazard/safety outputs* rather than summaries.

Judge-ready text (copy/paste):

> “We adapt the Video Search & Summarization recipe pattern for continuous video analytics. Rather than summarizing content, we query Cosmos Reason 2 repeatedly to assess dynamic hazards and update shared safety decisions.”

🧠 HOW THESE MAP TO YOUR PIPELINE

Here’s the precise mapping between Cookbook recipes and your system modules:

Cosmos Recipe	Your Use
Egocentric Social & Physical Reasoning	Core reasoning engine (hazards, affordances, constraints)

Video Search & Summarization	Continuous video processing pattern
Physical Plausibility Prediction (inspiration)	Semantic logic for risk filtering (not numeric scoring)
Intelligent Transportation Post-Training	*Not applied* — justified by design

---

4. End-to-End Pipeline (diagram in words)

Egocentric Drone Video
↓
Video Analytics Reasoning Agent
↓
Egocentric Scene Understanding (Cosmos Reason 2)
↓
Hazard & Constraint Identification

- **Obstacles**
- **Terrain**
- **Environmental hazards**
- **Human traversal constraints**
  ↓
  Physical Plausibility & Safety Filtering
  ↓
  Dual Risk Map (Drone vs Human)
  ↓
  Optional Exit Context Update (Directional, Non-Planning)
  ↓
  Navigation & Guidance Recommendation
  ↓
  Textual Reasoning Explanation

This clearly demonstrates:

Vision → Reasoning → Shared Safety Decision → Explanation

One-Sentence Elevator Pitch (final)

“We built a drone-based reasoning agent that evaluates shared physical safety step-by-step, using non-planning, shared-safety reasoning with optional exit context to determine whether a human can safely follow a drone.”