🧠 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

---

📜 Judge-Safe Summary for README

Copy/paste this into your README under "External Resources / Cosmos Usage":

Cosmos Cookbook Integration

This project uses the **Egocentric Social & Physical Reasoning** recipe from the Cosmos Cookbook to power the core reasoning pipeline. We continually query Cosmos Reason 2 on egocentric drone video to extract hazard and spatial context for shared human–robot safety evaluation. Our system adopts the video analytics pattern from the Video Search & Summarization recipe for frame-by-frame reasoning.

We intentionally do *not* apply the post-training recipe because our goal is **generalized physical safety reasoning**, not domain-specific fine-tuning. Post-training biases the model to specific labeled examples, limiting its hazard vocabulary and interpretability. Our physical plausibility

logic is inspired by the Physical Plausibility Prediction concepts, implemented through explicit constraint logic rather than numeric physics scoring.