📊 TEST → FINE-TUNE → TEST Workflow

✅ PHASE 1 — Baseline Testing (No Fine-Tuning)

1. Prepare Evaluation Dataset

Collect a labeled dataset appropriate to your safety task.

Dataset structure example:

Image / Video	Label	Category
real_floor_crack1.jpg	cracked	hazard
simulation_floor_crack1.png	cracked	hazard
real_floor_clean1.jpg	not cracked	safe
real_shadow1.jpg	not cracked	ambiguous

Labeling guidelines:

- Clear ground truth: “cracked” / “not cracked”
- Include edge cases (shadows, partial cracks, texture noise)

---

2. Define Evaluation Prompts

Create question prompts the model will answer per sample:

Binary safety:

```
“Is this floor cracked? Yes or No.”
```

Task-oriented safety:

```
“Is this surface safe for robot traversal? Yes or No.”
```

Severity classification:

```
“How would you rate the hazard level of this surface from 1 (safe) to
5 (dangerous)?”
```

---

3. Run Inference with Base Model

Call Cosmos Reason 2’s inference endpoint on each sample with your prompts.

Log outputs:

- Model answer
- Reasoning text
- Confidence (if available)
- Timestamp, prompt used

---

4. Evaluation Metrics

Calculate the following to quantify baseline performance:

Metric	Description
Accuracy	Correct classification rate
Precision / Recall	Especially on “hazard” classes
Confusion matrix	Where errors occur
Uncertainty analysis	Cases of low-confidence answers

A simple accuracy formula:

```
Accuracy = (# correct predictions) / (total samples)
```

If the base model performs *well enough* (e.g., >85–90% on your key tasks), you may not need fine-tuning.

📈 PHASE 2 — Fine-Tune Training (Optional)

Proceed only **if baseline performance is insufficient.**

1. Select Training Data

Use your labeled dataset from Phase 1.

For static reasoning tasks:

- You can use **images + prompt/response pairs**
- Simulation or real images are OK, but *include some real data if possible*

Example training sample format:

```
{
  "image_path": "floor_crack1.jpg",
  "prompt": "Is this floor cracked? Yes or No.",
  "target": "Yes"
}
```

Pack your dataset in a filesystem that resembles Cosmos Cookbook custom dataset format.

---

2. Fine-Tune Execution

Use Cosmos post-training workflows:

- Fine-tune Cosmos Reason 2 using supervised examples above
- Use **domain randomization** if using synthetic data

If your dataset is small, consider:

- Few-shot tuning
- Regularization (e.g., dropout)
- Mix real and synthetic

---

3. Training Validation

During training, evaluate on a held-out validation set:

Epoch	Val Loss	Val Accuracy
1	0.83	71.2%
2	0.45	86.7%
3	0.35	89.3%

Stop when:

- Validation accuracy plateaus
- Loss stops decreasing

📊 PHASE 3 — Post-Fine-Tuning Testing

1. Run Inference with Fine-Tuned Model

Use the same test set from Phase 1.

Log the same outputs:

- Predictions
- Reasoning text
- Confidence

2. Compare Metrics

Compute the same metrics:

Metric	Base Model	Fine-Tuned Model
Accuracy	74.6%	88.9%
Precision	69.2%	83.5%
Recall	75.0%	90.1%

Improvement here justifies fine-tuning.

🧠 Decide and Document

📌 If Base Model Was Enough:

Show a side-by-side table comparing base model scores with “target performance thresholds.” Explain why you *chose not to fine-tune*.

For example:

> “Base model accuracy of 92% on real images met our safety requirements (≥90%). Fine-tuning would risk overfitting to synthetic images and offered minimal improvement.”

📌 If Fine-Tuning Helped:

Show:

- Before vs. after metrics
- Qualitative error analysis (e.g., example where base model fails but fine-tuned succeeds)
- Why improvements matter (real-world risk reduction, etc.)