Team Division Plan for Remaining AI Oil Spill Project Tasks (English Version)

Below is a clean and efficient way to divide the remaining workload between two people. The project naturally splits into two major branches:

Overall Remaining Tasks

Branch A — Physics + AI Prediction

Branch B — Chemistry/Biology Modeling + Validation & Reporting

Role A: Physics & AI Lead

> *Focus: building realistic simulation data, strengthening the AI predictor, running physical/movement scenarios*

A1. Improve Physics-Based Synthetic Data Generator

- Update `utils/physics_ops.py` and `data/make_synthetic_data.py`
- Add:
  - Time-varying U, V fields (currents/wind)
  - Randomized environmental variability
  - Parameterized constants (D, beta, wind factor, current noise, grid size, T_total)
- Ensure reproducibility (set seeds, configs)

A2. Enhance the ConvLSTM AI Model

- Update:
  - `ai_predictor/model_conv_lstm.py`
  - `ai_predictor/train_predictor.py`
- Add:
  - train/validation/test split
  - multi-step prediction (`T_pred > 1`)
  - loss metrics (MSE, MAE, IoU)
  - YAML/JSON config file for training settings

A3. Taean Oil Spill Scenario Script

- Create new file: `scenarios/taean_case.py`
- Implement:
  - Grid representing Taean Manripo Coast (your X-Y setup)
  - Initial oil spill location from your real coordinates
  - Wind/current forcing similar to 2007 Hebei Spirit accident
- Run ConvLSTM predictions for future spill movement
- Output severity maps & visualizations

A4. Code Refactoring

- Clear documentation + function explanations
- Standardize tensor shapes & naming
- Organize demo and scenario scripts:
    - `run_demo.py`
    - `run_taean_case.py`

Role B: Bio–Chemical Modeling & Analysis Lead

> *Focus: extracting scientific constants from papers, ecological modeling, result interpretation, documentation*

B1. Build Parameter Table From Scientific Papers

Create `doc/bio_chem_params.md` summarizing:

- DO consumption rate (k_{consume})
- DO reaeration rate (k_{reaer})
- Plankton mortality, recovery time
- Oil toxicity thresholds
- Biological oxygen demand parameters
- Units + source paper references

B2. Improve Biology & Chemistry Mini-Models

Update:

- `utils/biology_ops.py`
- `chemistry_ops.py` (if needed)

Replace placeholder constants with real scientific values.
Add proper documentation such as:

# Based on Newsted et al. (1987)

# sens_coeff = 0.0001 (in response to PAH concentration)

Refine:

- `update_DO()`
- `plankton_response()`
- `ecological_recovery_index()`

Make sure inputs/outputs are clearly defined.

B3. Ecological Impact Reporting System

- Expand the reporting section in `run_demo.py`
  - Average recovery index
  - Worst impacted zone index
  - Final DO levels
  - Plankton survival %
- Automatically save reports to:
  - `reports/demo_report.md`

B4. Experimental Design & Validation

- Use A’s improved predictor to run multiple scenarios:
    - Vary D (diffusion), wind, initial spill size
    - Compare ecological outcomes (DO, plankton, recovery)
- Prepare scientific summary paragraphs for:
    - poster
    - paper-style documentation
    - midterm report

Joint Tasks (A & B) — Must Be Agreed Together

Shared Interface

1. **Data Format Definition**
    - Standardize the channel meaning:
        - C0 = oil
        - C1 = U (x-current)
        - C2 = V (y-current)
    - Document this in `doc/data_format.md`.
2. **Parameter Naming Convention**

   - Both must use the same naming:
      - D, beta (physics)
      - k_consume, k_reaer, k_bio (biology)
      - Units marked clearly

3. **Unified Execution Scripts**
   - Both `run_demo.py` and `run_taean_case.py` must use the same API
   - Avoid breaking each other’s code
4. **Git Branch Strategy**
   - A: `feature/physics-model`, `feature/taean-scenario`
   - B: `feature/bio-chem`, `feature/reporting`

Concise Summary (One Line Each)

- **A**: “Physics simulation, AI prediction, scenario modeling.”
- **B**: “Scientific parameters, biology/chemistry modeling, ecological analysis.”