RobotMotion Dataset Format

This document describes the **data organization, file structure, and episode representation** used in the **RobotMotion Humanoid Dataset**.

RobotMotion adopts a **skill–episode abstraction** that unifies heterogeneous data sources—including egocentric human demonstrations, simulation data, and teleoperated humanoid executions—under a **transparent, extensible, and learning-friendly representation**.

---

1. Dataset Abstraction

RobotMotion organizes data around two core concepts:

- **Skill**: a semantic task definition (e.g., *pick_up_cup*, *open_door*)
- **Episode**: a single execution of a skill from start to termination

Each skill contains multiple episodes collected under varying conditions, embodiments, and data sources.

---

2. High-Level Directory Structure

The dataset follows a hierarchical, skill-centric layout:

```
dataset/
├── skills/
│   ├── pick_up_cup/
│   │   ├── episodes/
│   │   │   ├── episode_0001/
│   │   │   │   ├── video/
│   │   │   │   ├── trajectory.jsonl
│   │   │   │   └── metadata.json
│   │   │   ├── episode_0002/
│   │   │   └── ...
│   │   └── skill_metadata.json
│   ├── pour_water/
│   └── ...
```

This structure enables:

- efficient indexing by task
- scalable episode-level expansion
- clean separation between skill semantics and execution data

3. Skill Metadata

Each skill is described by a dedicated metadata file (skill_metadata.json) with the following fields:

Skill Fields

- skill_id: integer identifier
- skill_name: short, machine-readable name
- skill_description: human-readable description
- total_episodes: number of episodes available for this skill

Example

```json
{
  "skill_id": 1,
  "skill_name": "pick_up_cup",
  "skill_description": "Pick up a cup from a table using one hand",
  "total_episodes": 45
}
```

4. Episode Structure

Each episode represents a **single task execution** and contains synchronized multi-modal data.

Core Episode Fields

- episode_id: unique identifier within the skill
- duration_seconds: episode length
- frame_count: number of frames or timesteps
- data_source: "human", "simulation", or "teleoperation"

Modalities (availability depends on collection type)

- RGB or RGB-D video
- Audio (optional)
- Robot proprioception (joint states, IMU, etc.)
- Actions or control commands

Each episode folder contains:

- raw sensor data (e.g., videos)
- time-series trajectories

- episode-level metadata

5. Trajectory Format

Time-series data is stored using **JSON Lines (.jsonl)**, where **each line corresponds to one timestep**.

Example

```
--------------------------------------
{
  "time": 0.033,
  "state": {
    "joint_positions": [...],
    "joint_velocities": [...]
  },
  "action": {
    "target_positions": [...]
  }
}
--------------------------------------
```

This format:

- supports high-frequency control data
- enables streaming and partial loading
- is compatible with **LeRobot** and standard robot learning pipelines

6. Annotations

Annotations are **optional** and depend on the data source and episode configuration.

Supported Annotation Types

- 2D bounding boxes
- Pixel-level segmentation
- Object tracking
- 6D object poses (simulation only)

If an episode contains no annotations:

```
"annotation": "No"
```

Annotation files, when present, are stored alongside episode data.

7. Synchronization

All modalities within an episode are synchronized using one of the following mechanisms:

- **Frame index alignment**
- **Shared timestamps**

This ensures consistent access across:

- perception
- action
- evaluation
- replay and visualization

8. Compatibility and Usage

RobotMotion is designed for seamless integration with:

- **LeRobot**
- Vision-Language-Action (VLA) pipelines
- Imitation learning frameworks
- Reinforcement learning systems

Adapters and dataset loaders will be released alongside official dataset versions.

---

9. Notes on Availability

- Not all modalities or annotations are available for every episode
- Dataset subsets may vary by **release version** and **access level**
- Public releases may contain a curated subset of the full internal dataset