# **Agentic AI for NeuroConv and Lab-to-NWB Pipelines**

## **1. Background**

### **Neurodata Without Borders (NWB)**

* NWB is a standardized file format for storing and sharing neurophysiology data.
* It unifies raw data, metadata, and analysis results into a single structure.
* Goal: make datasets comparable across labs, reproducible, and FAIR (Findable, Accessible, Interoperable, Reusable).

### **NeuroConv**

* Python package developed to facilitate conversions from various proprietary data formats into NWB.
* Provides **interfaces** (e.g., for SpikeGLX, NeuroExplorer, SLEAP, DeepLabCut, etc.).
* Handles raw data extraction, metadata mapping, and file writing.
* Flexible: can support multiple modalities (electrophysiology, imaging, behavior).

### **CatalystNeuro**

* A company that collaborates with labs to standardize their data pipelines.
* Goes beyond NeuroConv by building **lab-to-NWB repositories**:  
  + Examples: mease-lab-to-nwb, stavisky-lab-to-nwb, churchland-lab-to-nwb.
  + Each repo is an **end-to-end pipeline** customized for a lab’s workflow.

## **2. What Lab-to-NWB Repos Contain**

CatalystNeuro builds these repos to adapt NeuroConv to each lab’s unique data environment.  
 Key components:

1. **Conversion Interfaces Config**
   * Which NeuroConv interfaces to load.
   * Example: SpikeGLX (electrophysiology) + DeepLabCut (behavior).
2. **Metadata Templates**
   * Lab-specific JSON/YAML schemas.
   * Pre-filled fields for experiment type, rig info, subject species, electrode maps.
3. **Preprocessing Steps**
   * Clock synchronization between devices.
   * Renaming/relabeling channels.
   * Data cleaning or trial segmentation.
4. **Main Entry Script**
   * CLI or Python script: python convert\_session.py --session 2023-07-14.
   * Automates batch conversion.
5. **Validation**
   * Runs NWB schema validation.
   * Ensures output is FAIR-compliant.

## **3. The Agentic AI Vision**

We want to build an **agentic AI system** that can **replicate CatalystNeuro’s lab-to-NWB creation process**:

* **Step 1: Input raw data**
  + AI receives files with no prior knowledge of the lab.
* **Step 2: Detect and classify data formats**
  + Use NeuroConv interfaces (format-aware tools).
  + Example: detect SpikeGLX files → use SpikeGLXRecordingInterface.
* **Step 3: Construct metadata schema**
  + Propose templates (species, task, rig configuration).
  + Allow user to fill missing fields interactively.
* **Step 4: Generate a conversion pipeline (lab-to-NWB repo)**
  + Create repo with:  
    - Interface loading script.
    - Metadata YAML.
    - Main conversion CLI.
    - Validation hooks.
* **Step 5: Execution & Reuse**
  + Repo can be run repeatedly on future datasets.
  + Mimics CatalystNeuro’s deliverable to labs.

## **4. Giving Context to the Agent**

To teach the agent **how to build lab-to-NWB repos automatically**, we need to provide context:

1. **Knowledge Base**
   * Summaries of existing CatalystNeuro lab-to-NWB repos.
   * Show typical structures: src/, metadata/, convert\_session.py.
2. **NeuroConv Documentation**
   * Explain how to use interfaces (.run\_conversion(), metadata injection).
3. **Design Patterns**
   * Hooks for preprocessing (clock alignment, channel mapping).
   * YAML-driven configs.
   * Script scaffolds for reproducibility.
4. **Few-shot Examples**
   * Example repo scaffolds the agent can copy/adapt.

## **5. Proposed Workflow for Research**

1. Collect and summarize 40+ CatalystNeuro lab-to-NWB repos.
2. Build a **meta-schema**: what common elements appear across repos.
3. Train the agent to:  
   * Identify formats.
   * Assemble interfaces.
   * Propose metadata.
   * Write scaffold code.
4. Validate output using NWB tools.

## **6. Expected Outcome**

* An **agentic AI assistant** that can:  
  + Take any new lab’s raw data.
  + Build a custom lab-to-NWB repo automatically.
  + Follow CatalystNeuro’s methodology (interfaces + metadata + preprocessing + validation).
* Long-term: reduce time/cost for labs to adopt NWB.