

# The Agentic Research Migration Kit

**Subject:** Standard Operating Procedures for organizing long-term, agent-driven ML Research.

**Goal:** Convert chaotic "scratchpad" repositories into structured, navigable "Scientific Archives".

## 1. The Philosophy: "Experiment-First"

Agents struggle with deep nesting and scattered contexts.

**The Gold Standard:** Every experiment is a **Self-Contained Unit**.

### The Anti-Pattern (Do Not Do This)

```
/models
  model_v1.py
  model_v2.py
/scripts
  train_v1.py
/logs
  v1_log.txt
```

*Why it fails:* An agent looking at `model_v2.py` doesn't know which log, config, or training script belongs to it.

### The Standard Structure

```
/experiments
  /v01_baseline_mlp
    /models
      model.py          <-- The exact code used
    /logs
      train.log
    /audits
      audit_results.json
    /checkpoints
    /tests
  journal.md           <-- The narrative history
  train.py             <-- The exact entry point
```

## 2. Migration Protocol (The "Deep Clean")

Drop this Python script into the root to organize loose files.

```
# cleanup_experiments.py (Generalized)
import os, shutil
```

```

ROOT = "experiments"
SUBDIRS = ["models", "tests", "scripts", "logs", "audits", "plots",
"checkpoints"]

for d in os.listdir(ROOT):
    path = os.path.join(ROOT, d)
    if not os.path.isdir(path) or not d.startswith("v"): continue

    # Create standard subdirs
    for s in SUBDIRS:
        if not os.path.exists(os.path.join(path, s)):
            os.makedirs(os.path.join(path, s))

    # Catalog and Move
    for f in os.listdir(path):
        src = os.path.join(path, f)
        if not os.path.isfile(src): continue

        # Rules
        if f.endswith((".png", ".pdf", ".jpg")): shutil.move(src,
os.path.join(path, "plots", f))
        elif "test" in f or "diag" in f:          shutil.move(src,
os.path.join(path, "tests", f))
        elif "results" in f.lower():              shutil.move(src,
os.path.join(path, "audits", f))
        elif "model" in f or "net" in f:          shutil.move(src,
os.path.join(path, "models", f))
        # Keep train.py and journal.md in root

```

### 3. Document Templates (The "Constitution") 📜

Create these 4 files in the project root to ground any future agents.

#### A. **README.md** (The Dashboard)

```

# [Project Name]: Current Status

> **SOTA**: V[XX] ([Description])
> **Focus**: [Current Goal]

## 🚀 Active Experiments
| Experiment | Status | Score | Notes |
|---|---|---|---|
| `vXX_description` | ● Running | 95.2% | Current leader. |

## 📖 Documentation
- [Protocols](PROTOCOLS.md): How to work here.

```

- [Metrics](METRICS\_METHODODOLOGY.md): What "Success" means.
- [Debugging](DEBUGGING\_GUIDE.md): Common fixes.

## B. PROTOCOLS.md (The Law)

### # Engineering Protocols

#### ## 1. The Directory Law

- **Strict Isolation**: Never import code from `v1` into `v2`. Copy-paste and refactor.
- **Root Hygiene**: The root folder contains *only* Documentation. All code lives in `experiments/`.

#### ## 2. The Agentic Override

- **Descriptive Names**: `v5\_attention\_fix` > `v5`.
- **Journaling**: Every experiment MUST have a `journal.md`. Update it *before* and *after* every major run.

#### ## 3. Deployment

- **Mini-Train First**: Run for 100 steps on local CPU before launching huge GPU jobs.
- **Audit First**: Write the test *before* the fix.

## C. METRICS\_METHODODOLOGY.md (The Ruler)

### # Metrics & Standards

#### ## North Star Metrics

1. **[Metric A]** (e.g., Accuracy): Exact match % on validation set.
2. **[Metric B]** (e.g., Latency): Tokens/sec on Reference Hardware (L4/A100).

#### ## The Audit Suite

Level	Name	Description	Pass Criteria
1	<b>Smoke Test</b>	Does it run?	Exit Code 0
2	<b>Convergence</b>	200 step mini-train	Loss < Initial Loss
3	<b>SOTA Certification</b>	Full evaluation	> Previous Best

## D. PROJECT\_MANIFEST.md (The Index)

(Optional: Auto-generated list of all experiments and their one-line summaries)

## 4. The "Handoff" Ritual 🤝

When an agent finishes a session, they must:

1. **Update the Journal:** "I tried X. It obtained Y logs. I recommend Z next."
2. **Clean the Room:** Move artifacts to `plots/` or `audits/`.
3. **Update Leaderboard:** If a new high score was set, log it in `README` or `LEADERBOARD.md`.

*This framework ensures that even if 10 different agents work on the project, the structure remains coherent and monolithic history is preserved.*