Morris Validation Project - File Deployment Guide

🎯 File Structure Organization

Current Status

- The uploaded "test_model_architecture.py" contains **source code** (not tests)
- Need to reorganize files into correct directory structure
- Several modules are correctly positioned, others need deployment

File Deployment Actions Required

1. Create Directory Structure

```
bash

python setup_project.py
```

2. Deploy Source Files to `src/`

✅ Already Implemented (from existing artifacts):

- `src/data_generation.py`
- `src/logging_utils.py`
- `src/checkpoint_manager.py`
- `src/memorization_metrics.py`

📁 Need to Deploy:

- `src/model_architecture.py` ← Use content from artifact "model_architecture_fixed"

3. Deploy Test Files to `tests/`

✅ Already Implemented:

- `tests/test_data_generation.py`
- `tests/test_logging_utils.py`
- `tests/test_checkpoint_manager.py`
- `tests/test_memorization_metrics.py` (with fixed SimplePerfectMemoryModel)

📁 Need to Deploy:

- `tests/test_model_architecture.py` ← Use content from updated artifact "test_model_architecture"

4. Root Files

✅ Already Positioned:

- `function_declarations.py`
- `requirements.txt`
- `run_tests.py`
- `setup_project.py` (updated)

Complete File Mapping

```
morris_validation/
├── src/
│   ├── __init__.py                [auto-created by setup]
│   ├── model_architecture.py      [Deploy from artifact]
│   ├── data_generation.py         [✅ Existing]
│   ├── logging_utils.py           [✅ Existing]
│   ├── checkpoint_manager.py      [✅ Existing]
│   └── memorization_metrics.py    [✅ Existing]
├── tests/
│   ├── __init__.py                [auto-created by setup]
│   ├── test_model_architecture.py [Deploy from artifact]
│   ├── test_data_generation.py    [✅ Existing]
│   ├── test_logging_utils.py      [✅ Existing]
│   ├── test_checkpoint_manager.py [✅ Existing]
│   └── test_memorization_metrics.py [✅ Existing]
├── logs/                          [auto-created by setup]
├── data_cache/                    [auto-created by setup]
├── function_declarations.py       [✅ Existing]
├── requirements.txt               [✅ Existing]
├── run_tests.py                   [✅ Existing]
└── setup_project.py               [✅ Updated]
```

🔧 Deployment Steps

Step 1: Run Setup

```
bash

python setup_project.py
```

Step 2: Save Artifact Files

Save these artifact contents to their respective files:

1. **src/model_architecture.py** ← Copy from "model_architecture_fixed" artifact
2. **tests/test_model_architecture.py** ← Copy from updated "test_model_architecture" artifact

Step 3: Verify Installation

```bash
# Install dependencies
pip install -r requirements.txt

# Run all tests
python run_tests.py
```

Step 4: Expected Test Results

After proper deployment, all tests should pass:

- `test_data_generation.py` ✅
- `test_logging_utils.py` ✅
- `test_checkpoint_manager.py` ✅
- `test_memorization_metrics.py` ✅ (fixed perfect memory model)
- `test_model_architecture.py` ✅ (proper transformer tests)

📊 Verification Checklist

- [ ] Directory structure created by setup_project.py
- [ ] All source files in `src/` directory
- [ ] All test files in `tests/` directory
- [ ] Dependencies installed: `pip install -r requirements.txt`
- [ ] All tests passing: `python run_tests.py`
- [ ] Ready for training_loop.py implementation

🚀 Next Phase

Once file structure is organized and tests pass:

1. **Implement training_loop.py** for complete Morris validation
2. **Run experiments** across GPT model sizes

3. **Validate 3.6 bits-per-parameter scaling law**

🐛 Troubleshooting

If tests fail:

1. Check that `src/` contains source files (not test files)
2. Check that `tests/` contains test files (not source files)
3. Verify imports in test files point to `src/` directory
4. Ensure all dependencies installed

If memorization test fails:

- Verify `SimplePerfectMemoryModel` gives perfect autoregressive predictions
- Check that conditional entropy < 0.01 for perfect memory model