

# Foqus ML Task — Runbook

Foqus ML Task — Runbook (Fresh Setup)

Generated: September 16, 2025

This document provides a clean, reproducible setup and run guide for the repository, including:

- Full environment setup (venv),
- Editable install,
- Formatting/linting,
- CLI smoke test,
- Unit tests, and
- Notes for CI and troubleshooting.

## Step 0 — Clone

0) Clone (if you don't already have the repo)

```
$ git clone https://github.com/msoltanpour/foqus-ml-task.git
```

```
$ cd foqus-ml-task
```

## Step 1 — Cleanup

1) Optional cleanup of caches (safe)

```
$ find . -name "__pycache__" -type d -prune -exec rm -rf {} +
```

## Step 2 — Create Virtualenv

2) Create a fresh virtual environment (Python 3.10+ recommended)

Option A — Python 3.11:

```
$ python3.11 -m venv .venv
$ source .venv/bin/activate
$ python -V
$ python -m pip install -U pip
```

Option B — Use your available Python 3.10+:

```
$ python3 -V    # ensure >= 3.10
$ python3 -m venv .venv
$ source .venv/bin/activate
$ python -V
$ python -m pip install -U pip
```

## Step 3 — Editable Install

3) Install the project in editable (dev) mode

```
$ pip install -e ".[dev]"
```

This installs runtime deps (torch, numpy, matplotlib, tqdm, pyyaml, einops) and dev tools (pytest, ruff, black, pre-commit, etc.).

## Step 4 — Pre-commit (Optional)

4) Pre-commit (format & lint on commit) – optional but recommended

```
$ pre-commit install
```

```
$ pre-commit run -a
```

If the hook modifies files, run it again until it reports no changes.

## Step 5 — CLI Smoke Test

5) CLI smoke test (CPU, 1 epoch)

```
$ foqus-train --epochs 1 --batch-size 8 --num-workers 0  
--device cpu --out-dir tmp_run  
$ ls -lah tmp_run
```

Expected artifacts in tmp\_run/:

- task3\_curves.png — loss plot
- task3\_curves.csv — per-epoch loss CSV
- task3\_best.pt — best checkpoint

## Step 6 — Run Tests

6) Run unit tests (fast)

```
$ pytest -q
```

All tests should pass. You may see a harmless warning from PyTorch Lazy modules.



## Step 7 — Full Training Example

7) Example full training run (CPU example; use `--device cuda` if available)

```
$ foqus-train \  
  --epochs 20 \  
  --train-length 800 --val-length 200 \  
  --batch-size 16 --num-workers 0 \  
  --image-size 128 --n-coils 8 \  
  --lr 3e-4 --margin 0.2 \  
  --out-dir report/figs/exp1
```

Outputs will be saved under `report/figs/exp1/`.

## Step 8 — CI on GitHub

8) CI (GitHub Actions) – when you push your branch

- Workflow file: `.github/workflows/ci.yml`
- Runs: Ruff, Black (check), and pytest on Python 3.11
- Badge is included in README.

# Appendix A — Full Reset

## Appendix A – Full reset checklist

If you want to reset the environment completely and verify reproducibility:

1) Deactivate & remove venv:

```
$ deactivate # if currently active; ignore if not  
$ rm -rf .venv
```

2) Remove caches:

```
$ find . -name "__pycache__" -type d -prune -exec rm -rf {} +
```

3) Recreate venv & reinstall:

```
$ python3.11 -m venv .venv # or python3 -m venv .venv if  
>=3.10  
$ source .venv/bin/activate  
$ python -m pip install -U pip  
$ pip install -e ".[dev]"
```

4) Smoke test & run tests:

```
$ focus-train --epochs 1 --batch-size 8 --num-workers 0  
--device cpu --out-dir tmp_run  
$ pytest -q
```

# Appendix B — Troubleshooting

## Appendix B — Troubleshooting

- `ModuleNotFoundError` for `foqus_ml_task`:
  - Ensure you've run: `$ pip install -e ".[dev]"` inside the activated venv.
- Editable install fails with 'Multiple top-level packages discovered':
  - `pyproject.toml` must restrict package discovery (already configured):

```
[tool.setuptools.packages.find]
where = ["."]
include = ["foqus_ml_task*"]
exclude = ["report*", "tests*", "assets*", "configs*"]
```
- Lint warnings about Ruff config:
  - We use the new `[tool.ruff.lint]` section; run pre-commit to auto-fix formatting.
- CUDA not found / training slow:
  - Use `--device cpu` or install a CUDA-enabled PyTorch build if you have a GPU.