also give me a guide for setting up git and git lfs after cloning, also as i am switching to ubuntu, do i need to mimic the symbolic links?

Looking at your screenshots, I can see your exact structure:

- Repo root: `civicpulse-ai_org/`
- `notebooks/data` is a **Windows junction** (already in `.gitignore` line 43)
- `.env` is gitignored (line 16)
- `data/processed/*.h5` and `*.npy` are gitignored
- `models/checkpoints/*.pt` tracked by Git LFS (only `.tmp` ignored)
- `civicpulse_ai.egg-info/` exists but is gitignored

Answering the Symlink Question First

No, you do not need to replicate the Windows junction on Ubuntu/WSL2. The junction `notebooks/data` was a Windows workaround because notebooks live in a subfolder but need to resolve `data/` paths. On Linux, the notebooks can reference `../data/` directly, or better — your `config.py` already uses absolute `Path(__file__).parent` resolution so paths work regardless. Just don't create any symlink — it will resolve naturally.

Git + Git LFS Setup After Cloning (WSL2/Ubuntu)

Step 1 — Install Git + LFS in Ubuntu

```
sudo apt install -y git git-lfs
git lfs install   # registers LFS hooks globally, one-time
```

Step 2 — Configure Git identity

```
git config --global user.name "Your Name"
git config --global user.email "your@email.com"

# Store credentials so you don't re-enter PAT every push
git config --global credential.helper store
```

Step 3 — Clone the repo

```
cd ~
git clone https://github.com/YOUR_USERNAME/civicpulse-ai_org.git
cd civicpulse-ai_org

# LFS pulls tracked files (best_model.pt etc) automatically after clone
git lfs pull
```

Step 4 — Verify LFS is tracking correctly

```
git lfs ls-files
# Should list: models/checkpoints/best_model.pt (once it exists)

cat .gitattributes
# Should show lines like:
# models/checkpoints/*.pt filter=lfs diff=lfs merge=lfs -text
```

Step 5 — Recreate `.env` (never in git)

```
# .env is gitignored — create it fresh on the GPU machine
cat > .env << EOF
CIVICPULSE_DEVICE=cuda
CIVICPULSE_BATCH_SIZE=128
CIVICPULSE_DATA_MODE=normal
CIVICPULSE_PATCH_SIZE=256
CIVICPULSE_HIDDEN_CHANNELS=64
CIVICPULSE_NUM_LAYERS=2
CIVICPULSE_LEARNING_RATE=0.001
EOF
```

Step 6 — Copy data files (gitignored, transfer manually)

```
# From Windows side into WSL2 (your Windows drive is at /mnt/c or /mnt/s)
# Based on your path: storage > ifhe > research > civic pulse...
cp "/mnt/s/ifhe/research/civic pulse/setup_CIVICPULSE_AI/civicpulse-ai_org/data/processed
   ~/civicpulse-ai_org/data/processed/

cp "/mnt/s/ifhe/research/civic pulse/setup_CIVICPULSE_AI/civicpulse-ai_org/data/processed
   ~/civicpulse-ai_org/data/processed/
```

> **Note:** Your `storage` drive will be `/mnt/s` in WSL2 — check with `ls /mnt/` after WSL boots to confirm the drive letter.

Step 7 — Install Python environment

```
python3.11 -m venv .venv
source .venv/bin/activate
pip install --upgrade pip

# PyTorch ROCm (for RX 7700 XT)
pip install torch torchvision torchaudio \
    --index-url https://download.pytorch.org/whl/rocm6.1

# Project packages
pip install -e .    # installs from setup.py — gets all deps including src/
pip install jupyter tqdm python-dotenv
```

Step 8 — No symlink needed, verify paths work

```
python -c "
from src.config import TrainingConfig
c = TrainingConfig()
c.print_summary()
"
# Should print config without any path errors
```

Pushing Checkpoint Back After Training

```
# After NB05 completes
cd ~/civicpulse-ai_org
git add models/checkpoints/best_model.pt
git commit -m "GPU training complete - RX7700XT 20/50/100 epochs"
git push origin main
# LFS will upload the .pt file to GitHub LFS storage automatically
```

Then on your local Windows machine:

```
git pull
git lfs pull   # downloads the new best_model.pt
```

What NOT to Do on Ubuntu

- ✗ Don't create `notebooks/data` symlink — not needed on Linux
- ✗ Don't commit `.env` — recreate it manually each machine
- ✗ Don't copy `venv/` or `civicpulse_ai.egg-info/` — both are gitignored, recreate with `pip install -e .`
- ✗ Don't run `git lfs install` inside the repo — run it globally before cloning

⁂

1. image.jpg
2. image.jpg
3. image.jpg
4. image.jpg
5. image.jpg