## Advanced Mode (CaseLinker)

FastAPI + React prototype based on the instructions. Streams a large CSV, encodes features, computes cosine similarities in blocks, builds a graph of matches, returns clusters, and includes a minimal UI (heatmap/graph) with centralized configuration and a procedure test.

### Run (Backend)

uvicorn advancedmode.backend.app:app --reload

Start a scan:

curl -X POST http://127.0.0.1:8000/scan/start \

-H "Content-Type: application/json" \

-d '{"duration\_hours": 0.02, "similarity\_threshold": 0.7, "year\_window": 5}'

Check status:

curl http://127.0.0.1:8000/scan/status

Get results:

curl http://127.0.0.1:8000/scan/results

Test procedure mappability (per bpcd/procedure.md):

curl "http://127.0.0.1:8000/test/procedure?sample\_limit=10000"

Get/update centralized config:

curl http://127.0.0.1:8000/config

curl -X POST http://127.0.0.1:8000/config -H "Content-Type: application/json" -d '{

"csv\_path": "fusion\_deduction/data/SHR65\_23.csv",

"similarity\_threshold": 0.7,

"year\_window": 5,

"geo\_window": null,

"sample\_limit": 48

}'

### Run (Frontend)

cd advancedmode/frontend

npm install

npm run start

# open http://localhost:5173

The dev server proxies API calls to http://127.0.0.1:8000.

### Structure

advancedmode/

backend/

app.py - FastAPI app with endpoints

scan\_controller.py- Orchestrates streaming -> features -> matcher -> graph -> clusters; returns matrix and edges

data\_loader.py - CSV streamer (batch iterator)

features.py - Simple numeric+categorical encoder with L2-normalization

matcher.py - Block-wise cosine similarity (pure Python demo); samples matrix for heatmap

graph\_engine.py - Build adjacency and connected components clustering; edge filtering (year/geography)

config.py - Centralized config (CSV path, thresholds, windows)

test\_mode.py - Procedure test (coverage and feasibility)

tests/

test\_matcher.py - Sanity test for matching

frontend/

index.html - Vite entry

vite.config.js - Proxy /scan, /config, /test to backend

package.json - React + Vite

src/

main.jsx - App bootstrap

App.jsx - UI: config panel, scan controls, heatmap/graph views, procedure test button

README.md - This file

### Notes

* Default CSV path is fusion\_deduction/data/SHR65\_23.csv (configurable via /config).
* Year/geography windows:
  + year\_window: only link cases with |YearA − YearB| ≤ window
  + geo\_window: if both have numeric CNTYFIPS, require |diff| ≤ window; otherwise require same MSA or State
* Results include:
  + matrix/labels: sampled similarity matrix for heatmap
  + top\_clusters: connected components sorted by size
  + edges: edges for the largest cluster (subset), for graph visualization
* For scale/performance, replace pure-Python loops with numpy/scipy; persist sessions and add WebSockets for live updates.