Analysis Script Execution and Result Directory Mapping

Overview of the Python-Based Correlation Analysis Pipeline

---

1. Purpose

This document describes how each Python analysis script is executed and how its outputs are organized into corresponding result directories.

The purpose of this guide is to provide a **clear one-to-one mapping** between:

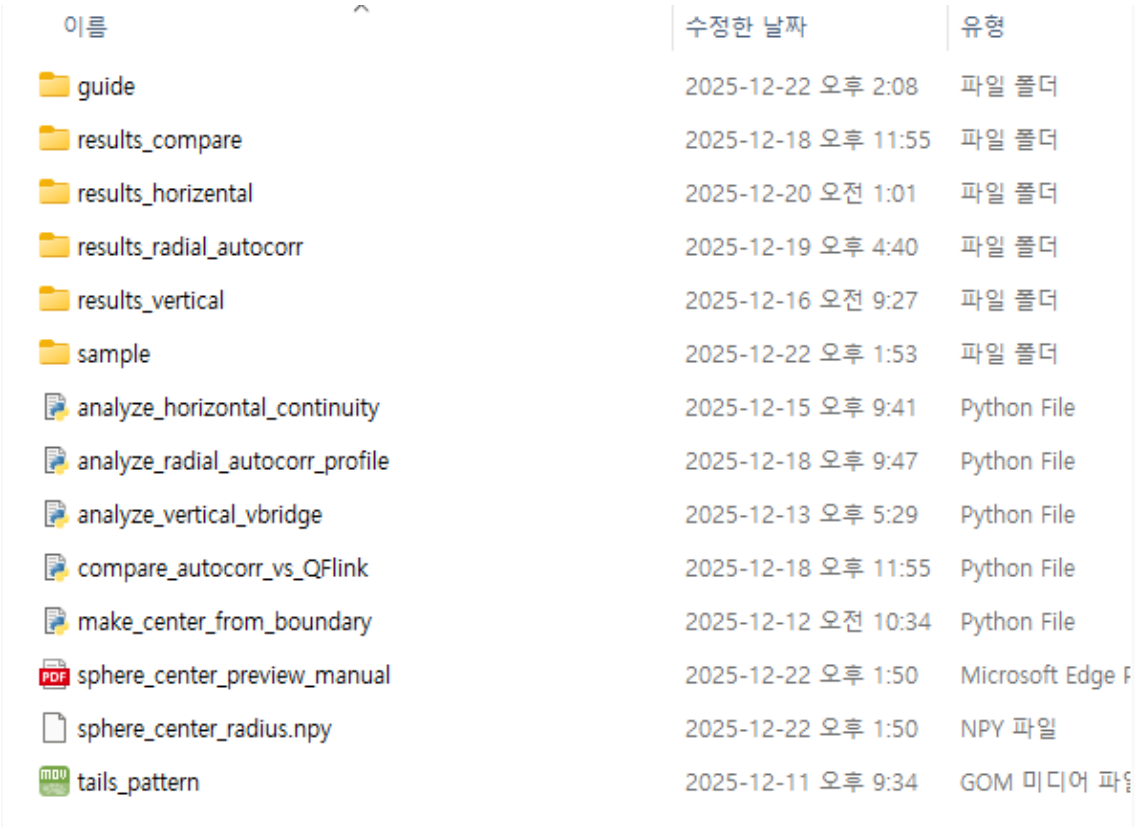
- analysis scripts,
- console execution steps, and
- generated result folders and files.

---

2. Analysis Directory Overview

All analysis scripts and input data are located in a single working directory.

Figure 1. Overall directory structure containing Python analysis scripts, input video, and result folders.



Key elements:

- Python scripts (`*.py`)
- Input video file (`tails_pattern.mov`)
- Result directories:
  - `results_horizontal/`
  - `results_vertical/`
  - `results_radial_autocorr/`
  - `results_compare/`

---

3. Horizontal Spatial Continuity Analysis

3.1 Script Execution

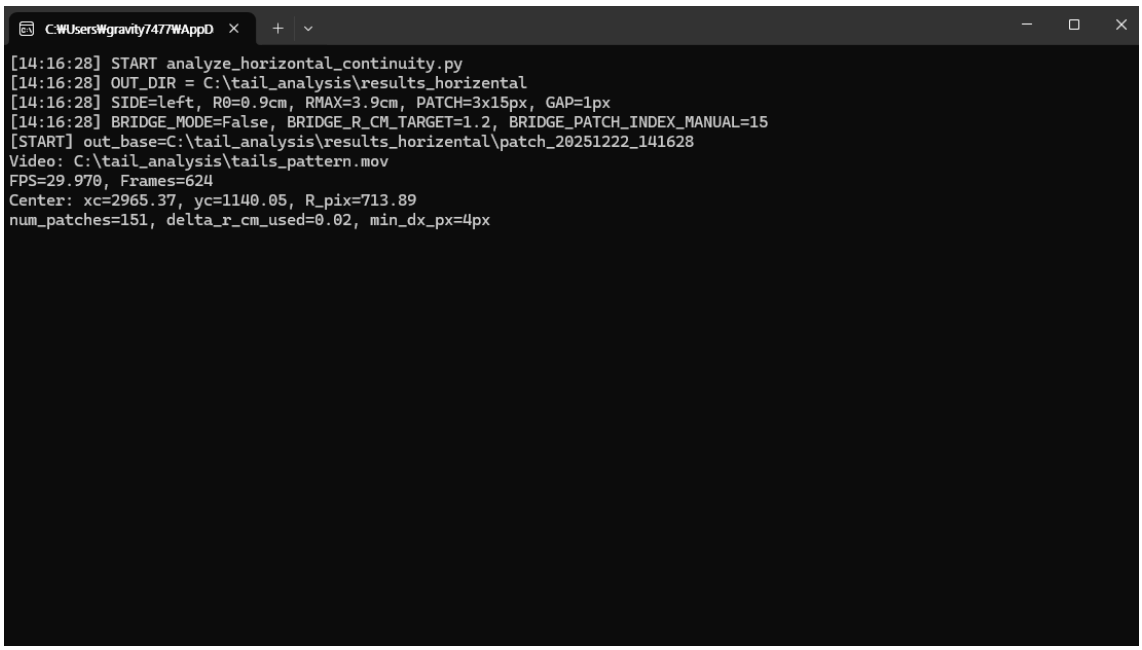
The horizontal spatial continuity analysis is performed by executing:

analyze_horizontal_continuity.py

When executed, the script:

- loads the input video tails_pattern.mov,
- reads the sphere center and radius from sphere_center_radius.npy,
- initializes patch-based sampling along the horizontal (radial) direction.

Figure 2. Console output during execution of analyze_horizontal_continuity.py.



3.2 Generated Results

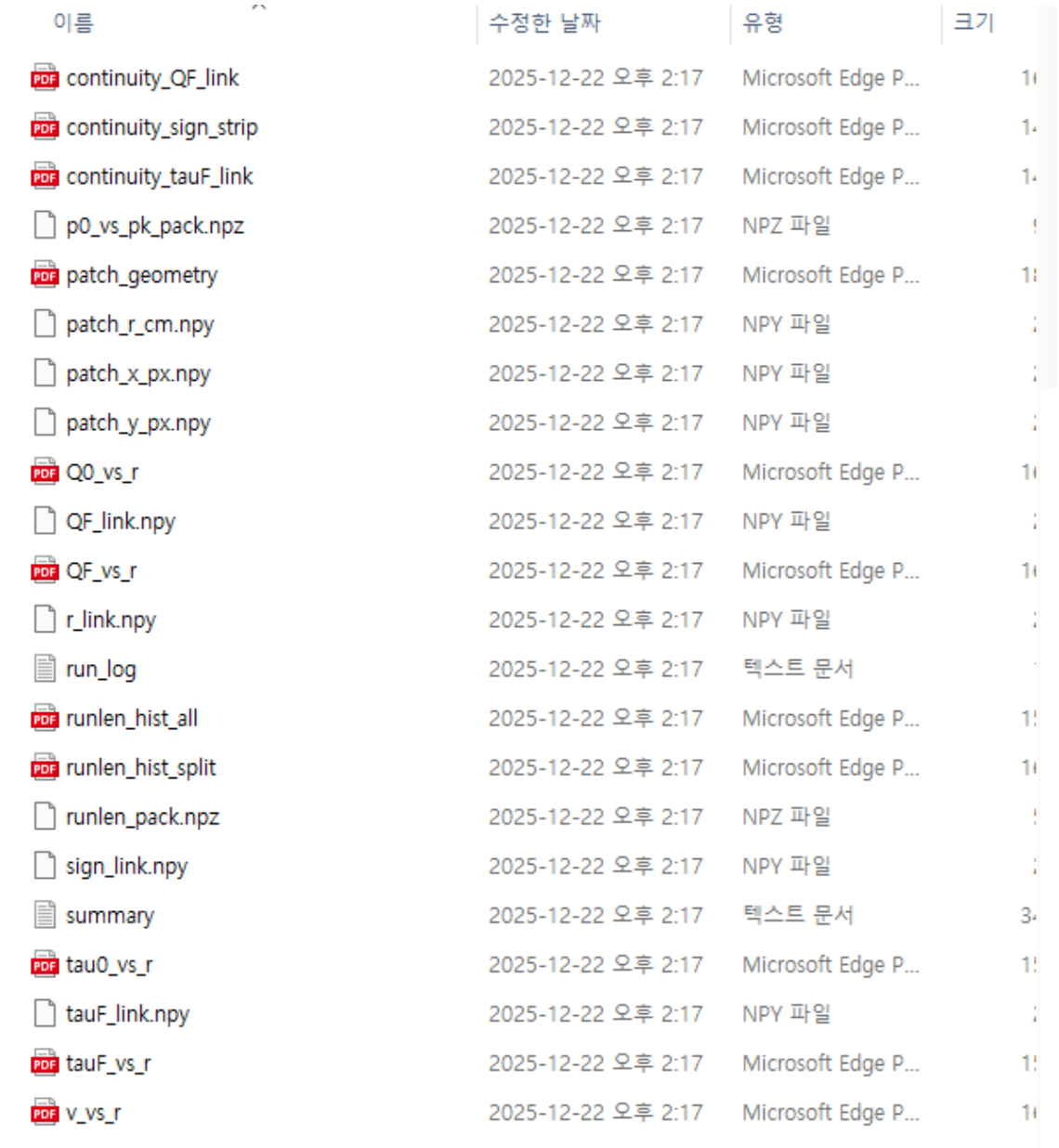
Upon completion, a new timestamped subdirectory is created inside:

results_horizontal/

This directory contains:

- horizontal continuity maps,
- run-length histograms,
- radial profiles of correlation strength and time lag,
- summary text files and NumPy data files.

Figure 3. Files generated in the `results_horizontal` directory after horizontal continuity analysis.



이름	수정한 날짜	유형	크기
continuity_QF_link	2025-12-22 오후 2:17	Microsoft Edge P...	1
continuity_sign_strip	2025-12-22 오후 2:17	Microsoft Edge P...	1
continuity_tauF_link	2025-12-22 오후 2:17	Microsoft Edge P...	1
p0_vs_pk_pack.npz	2025-12-22 오후 2:17	NPZ 파일	
patch_geometry	2025-12-22 오후 2:17	Microsoft Edge P...	1
patch_r_cm.npy	2025-12-22 오후 2:17	NPY 파일	
patch_x_px.npy	2025-12-22 오후 2:17	NPY 파일	
patch_y_px.npy	2025-12-22 오후 2:17	NPY 파일	
Q0_vs_r	2025-12-22 오후 2:17	Microsoft Edge P...	1
QF_link.npy	2025-12-22 오후 2:17	NPY 파일	
QF_vs_r	2025-12-22 오후 2:17	Microsoft Edge P...	1
r_link.npy	2025-12-22 오후 2:17	NPY 파일	
run_log	2025-12-22 오후 2:17	텍스트 문서	
runlen_hist_all	2025-12-22 오후 2:17	Microsoft Edge P...	1
runlen_hist_split	2025-12-22 오후 2:17	Microsoft Edge P...	1
runlen_pack.npz	2025-12-22 오후 2:17	NPZ 파일	
sign_link.npy	2025-12-22 오후 2:17	NPY 파일	
summary	2025-12-22 오후 2:17	텍스트 문서	3
tau0_vs_r	2025-12-22 오후 2:17	Microsoft Edge P...	1
tauF_link.npy	2025-12-22 오후 2:17	NPY 파일	
tauF_vs_r	2025-12-22 오후 2:17	Microsoft Edge P...	1
v_vs_r	2025-12-22 오후 2:17	Microsoft Edge P...	1

4. Vertical Synchronous (Bridge) Analysis

4.1 Script Execution

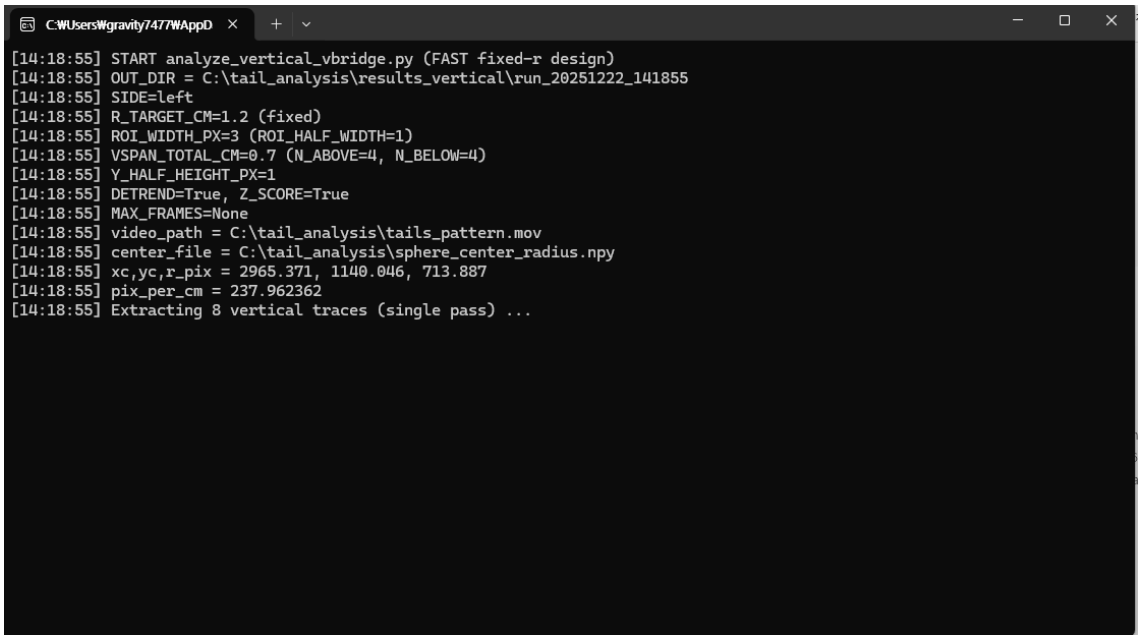
Vertical synchronous analysis at a fixed radial anchor is performed by executing:

analyze_vertical_vbridge.py

This script:

- uses a fixed radial position,
- extracts vertically separated intensity traces,
- computes zero-lag correlation matrices to assess temporal synchrony.

Figure 4. Console output during execution of analyze_vertical_vbridge.py.



4.2 Generated Results

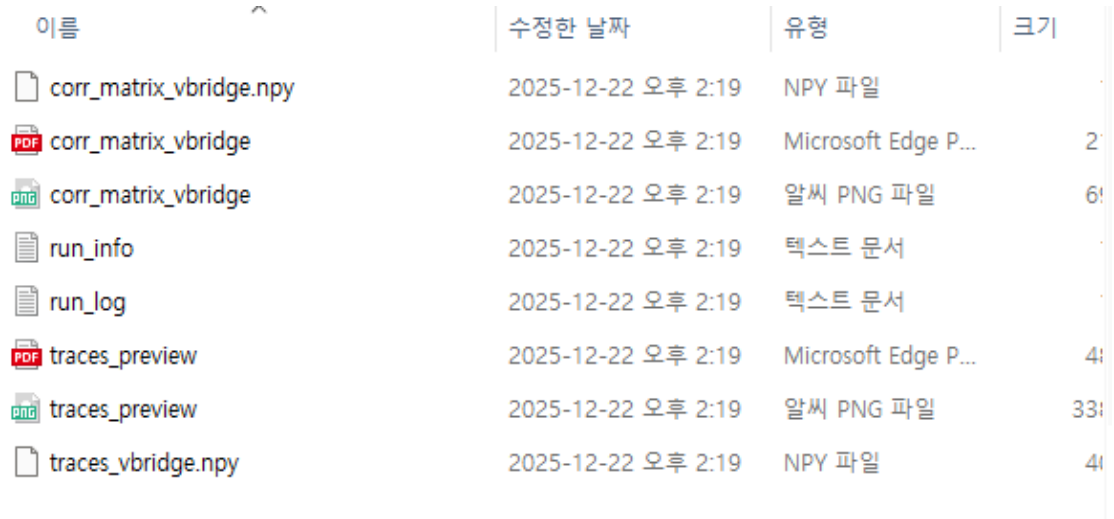
The script generates a timestamped result directory inside:

`results_vertical/`

This directory includes:

- vertical correlation matrices,
- trace previews,
- synchrony visualization plots,
- associated NumPy and text files.

Figure 5. Files generated in the `results_vertical` directory after vertical synchrony analysis.



---

5. Radial Temporal Persistence Analysis

5.1 Script Execution

Radial temporal persistence is analyzed by executing:

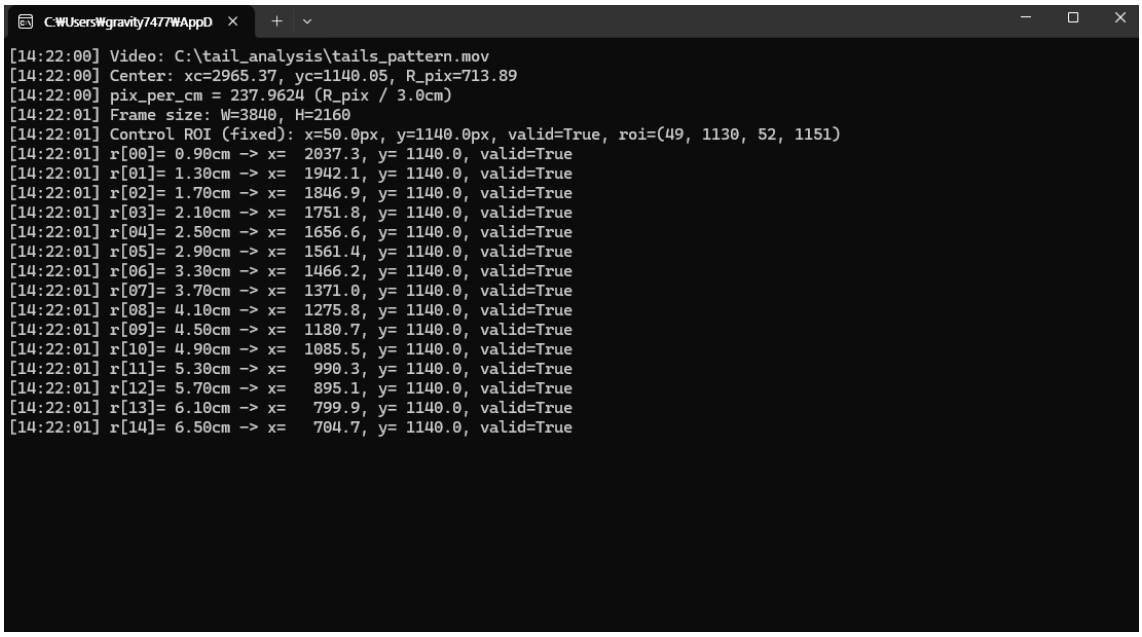
`analyze_radial_autocorr_profile.py`

This script:

- samples intensity time series along the radial direction,
- computes temporal autocorrelation functions,

- evaluates time persistence as a function of radial distance.

Figure 6. Console output during execution of `analyze_radial_autocorr_profile.py`.



5.2 Generated Results

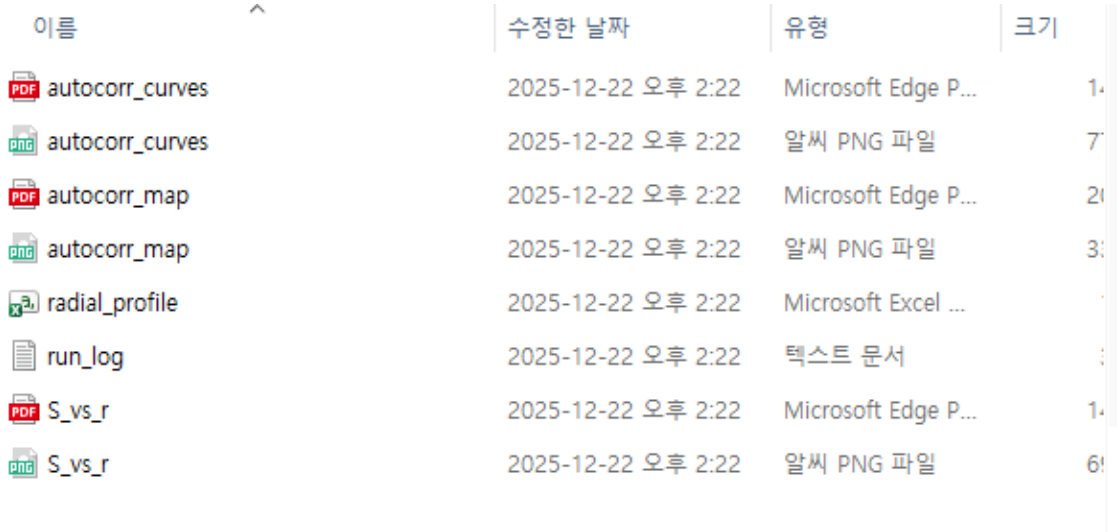
The outputs are written to a timestamped directory inside:

`results_radial_autocorr/`

Generated files include:

- autocorrelation curves,
- radial autocorrelation maps,
- persistence metrics,
- summary logs and data tables.

Figure 7. Files generated in the `results_radial_autocorr` directory after radial autocorrelation analysis.



---

6. Comparative Analysis: Continuity vs Temporal Persistence

6.1 Script Execution

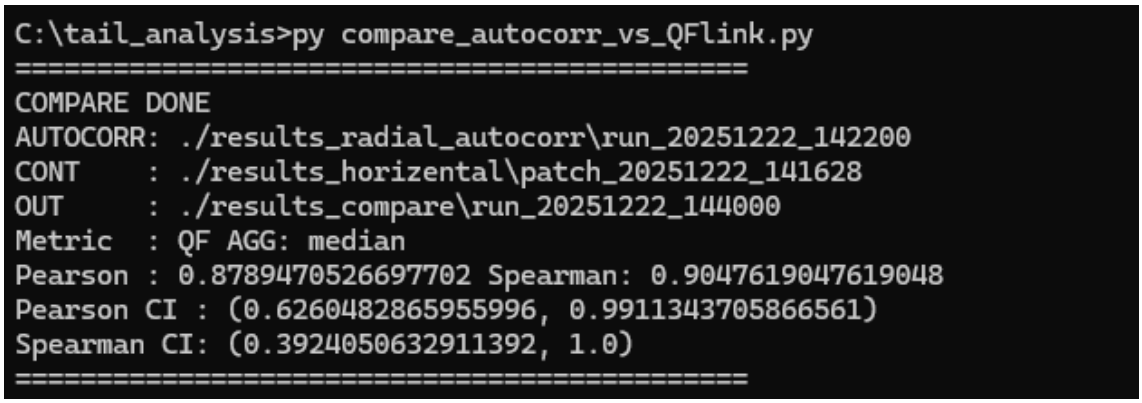
A final comparative analysis is performed using:

`compare_autocorr_vs_QFlink.py`

This script:

- automatically selects the **most recent** result directory from `results_horizontal/`,
- automatically selects the **most recent** result directory from `results_radial_autocorr/`,
- compares spatial continuity metrics with temporal persistence metrics.

Figure 8. Console output during execution of `compare_autocorr_vs_QFlink.py`.



6.2 Generated Results

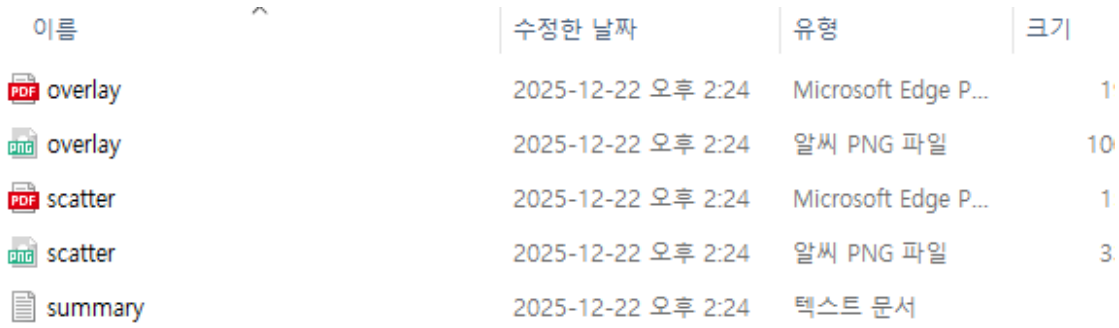
Comparison results are stored in:

`results_compare/`

These files quantify:

- the correspondence between spatial continuity and temporal persistence,
- agreement between horizontal correlation structure and radial memory effects,
- summary plots and comparison statistics.

Figure 9. Files generated in the results_compare directory after comparative analysis.



---

7. Summary of the Analysis Flow

The complete analysis pipeline proceeds as follows:

1. Execute `make_center_from_boundary.py` → define sphere geometry
2. Execute `analyze_horizontal_continuity.py` → spatial continuity
3. Execute `analyze_vertical_vbridge.py` → vertical temporal synchrony
4. Execute `analyze_radial_autocorr_profile.py` → temporal persistence
5. Execute `compare_autocorr_vs_QFlink.py` → continuity–persistence comparison

Each script produces outputs in a **dedicated result directory**, ensuring traceability and reproducibility.