Does Dealer Gamma Move Volatility? Causal Evidence from the Option-Expiration Roll-off

Code Documentation: scripts, data flow, and pipeline

Duc V. Le

Georgetown University

June 2026

Abstract

This document describes the analysis code for an *empirical* instrumental-variables study of how option-dealer gamma hedging affects equity-market volatility — whether dealer gamma causally moves realized volatility, estimated on WRDS / OptionMetrics data. The dollar gamma scheduled to expire at each monthly option expiration serves as a calendar-determined instrument for dealer gamma. Each script is documented below by purpose, inputs, and outputs, and the overall pipeline is shown as a diagram. The licensed source data are not included in the repository; the acquisition scripts rebuild every panel from WRDS for users with their own access.

Code availability. `github.com/duc-v-le/dealer-gamma-volatility`

Contents

1 Overview 2

2 Pipeline diagram 3

3 Data-acquisition layer 4

4 Empirical-analysis layer 5

5 Utilities and manuscript figures 7

6 Output artifacts 8

7 Reproduction order 9

1 Overview

The repository is organized as a single pipeline: WRDS / OptionMetrics panels are acquired, processed, and analyzed, producing the result tables in `data/` and figures in `figures/`.

Empirical workflow (real data)

- Data Acquisition:
  - Scripts named `wrds_download_*.py` connect via `wrds_lib.py`.
  - They pull *dealer-gamma* and *price panels* from WRDS / OptionMetrics.
  - Raw files are saved directly to the `data/real/` directory.
- Data Analysis:
  - Scripts named `e4_*.py` read the downloaded panels.
  - They estimate *instrumental-variables* and *panel specifications*.
  - Final outputs are written as small *result tables* and *figures*.

Data and credentials

- Authentication:
  - The WRDS username and password are read at run time from `~/.pgpass` or the `WRDS_USER` environment variable.
  - No secret keys or credentials are stored directly in the code source files.
- Data Privacy:
  - The licensed data panels saved in `data/real/` are strictly non-redistributable.
  - Only the aggregated final result tables are preserved inside `data/`.

2 Pipeline diagram

Figure 1 shows the data flow. Boxes are colour-coded:

- **External source** (red): licensed WRDS / OptionMetrics panels.
- **Processing scripts** (blue): acquisition and analysis scripts.
- **Intermediate data** (gray): panels and other temporary files.
- **Final outputs** (green): result tables and figures.

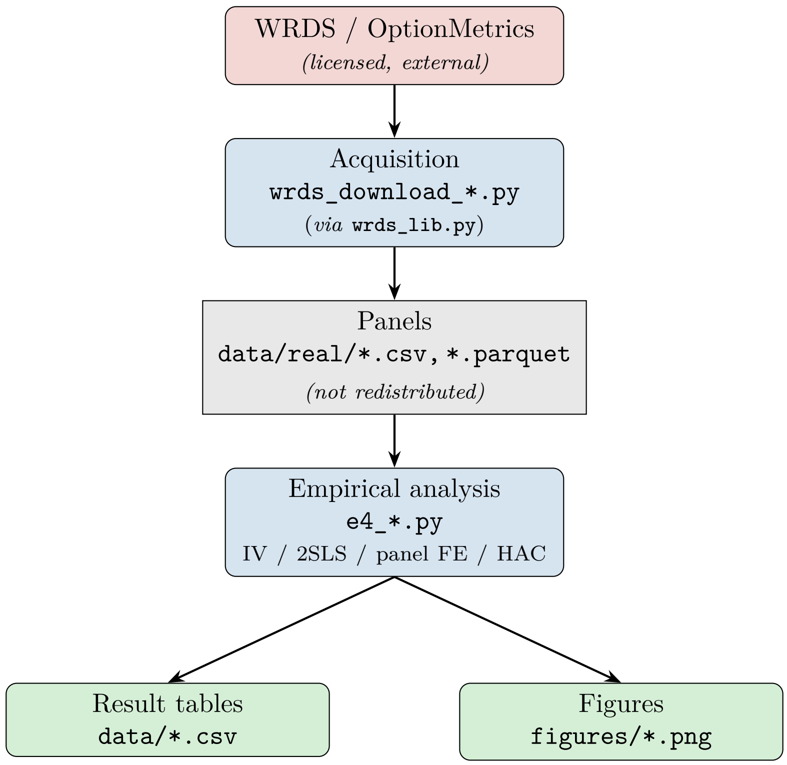

Figure 1: End-to-end pipeline. WRDS / OptionMetrics data is acquired, processed, and analyzed top-to-bottom, producing the result tables and figures.

Module dependencies.

- `wrds_lib.py` is imported by every acquisition and WRDS-backed analysis script.
- `paperA_figures.py` reads several result tables to assemble the manuscript figures.
- All other scripts are standalone entry points.

3 Data-acquisition layer

These scripts connect to WRDS and write panels to `data/real/`. They require a WRDS account; the panels they produce are the inputs to the empirical layer.

Script	What it does	Writes
`wrds_lib.py`	Minimal WRDS PostgreSQL connection helper; reads credentials from `~/.pgpass`. Imported by every WRDS script.	— (library)
`wrds_schema_probe.py`	Documents OptionMetrics schema discovery (tables, columns) for provenance.	— (console)
`wrds_rank_universe.py`	Ranks the optionable universe by liquidity to define the sample.	`universe_ranked.csv`
`wrds_download_spx_gamma.py`	Daily SPX dealer-gamma and price panel (OptionMetrics SPX).	`spx_gamma_panel.csv`
`wrds_download_gamma.py`	Generalized single-name gamma panel for any ticker.	`⟨name⟩_gamma_panel.csv`
`wrds_download_panel.py`	Top-50 liquid-options panel of net dealer gamma.	`panel_gamma.csv`
`wrds_download_panel_sp500.py`	Full S&P 500 panel; parallel-by-year, resumable.	`panel_sp500.parquet`, `sp500_secids.csv`
`wrds_download_panel_sp500_cp.py`	Full S&P 500 panel with call/put-split gamma.	`panel_sp500_cp.parquet`
`wrds_download_spx_expiry.py`	SPX gamma split by days-to-expiry bucket.	`spx_gamma_expiry.csv`
`wrds_download_spx_rolloff.py`	SPX dollar-gamma scheduled to expire by the next monthly expiration (the instrument).	`spx_rolloff.csv`
`wrds_download_rolloff.py`	Generalized scheduled-roll-off panel for any ticker.	`⟨name⟩_rolloff.csv`, `spx_rolloff.csv`

4 Empirical-analysis layer

These scripts read the panels above and estimate the specifications. Each writes a small result table (regression coefficients, t-statistics, first-stage diagnostics); some also write a figure. The common dependent variable is forward realized volatility; the key regressor is dealer gamma, instrumented by the gamma scheduled to roll off at the next monthly expiration.

Script	**What it does**	**Reads → writes**
`e4_leadlag_real.py`	Lead–lag of dealer short-gamma concentration vs. future realized volatility on the real SPX panel.	`spx_gamma_panel.csv` → `e4_real_leadlag.csv`, `e4_real_leadlag.png`
`e4_real_robust.py`	Newey–West (HAC) inference plus a stress-episode overlay.	`spx_gamma_panel.csv` → `e4_real_hac.csv`, `e4_real_episodes.png`
`e4_crossname.py`	Single-name generalization (e.g. NVDA/TSLA/AAPL) with a VIX-level control.	`⟨name⟩_gamma_panel.csv` → `e4_crossname.csv`, `e4_crossname.png`
`e4_causal_opex.py`	Expiration-calendar (OPEX) instrument for dealer gamma; hand-rolled 2SLS plus an event study.	`spx_gamma_panel.csv` → `e4_causal_opex.csv`, `e4_opex_eventstudy.png`
`e4_iv2sls.py`	Proper 2SLS (`linearmodels`) on gamma-by-expiry; diagnoses instrument strength.	`spx_gamma_expiry.csv` → `e4_iv2sls.csv`
`e4_iv_rolloff.py`	Scheduled-roll-off IV (the strong instrument).	`spx_rolloff.csv` → `e4_iv_rolloff.csv`
`e4_iv_rolloff_fwdvol.py`	Roll-off IV under the unified forward-volatility outcome.	`spx_rolloff.csv` → `e4_iv_rolloff_fwdvol.csv`
`e4_iv_robust_excl.py`	Exclusion-restriction hardening: volume and ΔVIX controls, over-identification J, falsification placebos.	`spx_rolloff.csv` → `e4_iv_robust_excl.csv`
`e4_paperA_outcomes.py`	Roll-off IV across single names and outcomes (volatility, tail, variance, kurtosis).	`⟨name⟩_rolloff.csv` → `e4_paperA_outcomes.csv`
`e4_panel_iv.py`	Pooled panel 2SLS with time fixed effects and liquidity heterogeneity (top-50).	`panel_gamma.csv` → `e4_panel_iv.csv`
`e4_panel_mechanism.py`	Convexity-near-stress test across VIX regimes (full S&P 500).	`panel_sp500.parquet` → `e4_panel_mechanism.csv`
`e4_paperA_robustness.py`	Specification robustness: peak-0DTE subsample and OPEX-week exclusion.	`panel_sp500.parquet` → `e4_paperA_robustness.csv`

Script	What it does	Reads → writes
e4_horizon.py	Forward-volatility horizon robustness (1/5/10/21-day).	panel_sp500.parquet → e4_horizon.csv
e4_systemic_aggregate.py	Aggregate market-gamma → index-volatility IV and a systemic-regime decomposition.	panel_sp500.parquet, spx_gamma_panel.csv → e4_systemic_aggregate.csv
e4_sign_robust.py	Call/put gamma decomposition; validates the dealer-sign convention.	panel_sp500_cp.parquet → e4_sign_robust.csv
e4_twoway_fullpanel.py	Two-way (firm + date) clustered panel outcomes.	panel_sp500_cp.parquet → e4_twoway_fullpanel.csv
e4_earnings_exclusion.py	Re-runs the panel IV excluding earnings windows (exclusion-restriction defense).	panel_sp500_cp.parquet → e4_earnings_exclusion.csv

5 Utilities and manuscript figures

Script	What it does	Reads → writes
`fetch_papers.py`	Downloads and parses the source-literature PDFs; records download status.	→ `manifest.csv`
`paperA_figures.py`	Assembles the eight manuscript figures (F1–F8) from the result tables and panels.	result tables, `panel_sp500_cp.parquet` → `paperA_F1..F8.png`
`md_to_latex.py`	Targeted Markdown-to-LaTeX converter used to typeset the manuscript.	— (LaTeX)

6 Output artifacts

`data/` (in the repository).

Aggregated result tables — small CSVs of coefficients, *t*-statistics, and summary statistics. No licensed records.

`figures/` (in the repository).

Generated PNG figures: the real-data plots (`e4_real_*`, `e4_opex_eventstudy`, `e4_crossname`) and the manuscript figures (`paperA_F1..F8`).

`data/real/` (not in the repository).

The licensed WRDS / OptionMetrics panels. Rebuilt from source by the acquisition scripts for users with their own WRDS access.

7 Reproduction order

All analysis requires a WRDS account. Run the acquisition script that produces a panel before the analysis script that consumes it. The main dependencies are:

Acquisition script	**Analysis scripts (run after)**
`wrds_download_spx_gamma.py`	`e4_leadlag_real.py`, `e4_real_robust.py`, `e4_causal_opex.py`
`wrds_download_spx_rolloff.py`	`e4_iv_rolloff.py`, `e4_iv_rolloff_fwdvol.py`, `e4_iv_robust_excl.py`
`wrds_download_spx_expiry.py`	`e4_iv2sls.py`
`wrds_download_gamma.py`	`e4_crossname.py`
`wrds_download_rolloff.py`	`e4_paperA_outcomes.py`
`wrds_download_panel.py`	`e4_panel_iv.py`
`wrds_download_panel_sp500.py`	`e4_panel_mechanism.py`, `e4_paperA_robustness.py`, `e4_horizon.py`, `e4_systemic_aggregate.py`
`wrds_download_panel_sp500_cp.py`	`e4_sign_robust.py`, `e4_twoway_fullpanel.py`, `e4_earnings_exclusion.py`

The sample-selection helper `wrds_rank_universe.py` and the schema probe `wrds_schema_probe.py` are run once to document universe construction and data provenance.