ReadMe File for Replication Package

”Learning from crises: A new class of time-varying parameter VARs with observable adaptation”

Empirical Forecasting Application: FRED-QD Quarterly Data

1 Purpose and Scope

This document provides detailed instructions for replicating the **FRED-QD forecasting results** included in the empirical section of the paper and online appendix.

The replication routines produce:

- out-of-sample forecasts for U.S. GDP growth, inflation (PCE), and the Federal Funds Rate,
- full expanding-window estimation for $p \in \{1, 2, 3, 4\}$,
- predictive distributions for horizons $h = 1, \ldots, 8$,
- quantile evaluation ($\tau = 0.10, 0.25, 0.50, 0.75, 0.90$),
- MAE, MSPE, MSFE, quantile scores, PITs, predictive likelihood.

This ReadMe covers only the **FRED-QD application**. Separate ReadMe files accompany the Euro Area application, U.S. monthly dataset, and Monte Carlo experiments.

2 Software and Requirements

2.1 MATLAB Environment

The code is written in MATLAB and has been tested under:

- MATLAB R2024b and R2025a,
- no proprietary MATLAB toolboxes are required.

External utility functions are included in the following folders:

- `matlabtoolbox/emtools/`
- `matlabtoolbox/emtexbox/`
- `matlabtoolbox/emgibbsbox/`
- `matlabtoolbox/emeconometrics/`
- `matlabtoolbox/emstatespace/`

These folders are automatically added to the MATLAB path by the main script.

3 Directory Structure

The FRED-QD replication package includes:

```
.  (root folder)
|-- main_forecasting_FREDQD.m
|-- structure.txt
|
|-- data/
|     |-- current.csv
|
|-- Forecasting_Results/
|     |-- Appendix_Forecast_FREDQD.m
|     |-- Figures_oos.m
|     |-- Tables_oos.m
|     |-- Tables_oos_function.m
|     |-- oos_FREDQD_p1.mat
|     |-- oos_FREDQD_p2.mat
|     |-- oos_FREDQD_p3.mat
|     |-- oos_FREDQD_p4.mat
|     |
|     +-- Appendix_Figures/
|           |-- OOS_FRED_p1.pdf
|           |-- OOS_FRED_p2.pdf
|           |-- OOS_FRED_p3.pdf
|           |-- OOS_FRED_p4.pdf
|
|-- functions/
|     |-- APVAR.m
|     |-- TVP_VAR.m
|     |-- TVP_VAR_FB.m
|     |-- TVP_RW_EB.m
|     |-- BVAR_OLS_iter.m
|     |-- CCMM_SVO.m
|     |-- extract.m
|     |-- fred_qd2.m
|     |-- getOLS.m
|     |-- pctile.m
|     |-- quantile.m
|     |-- (additional helper functions)
```

```
|
|-- fred-databases_code/
|     +-- fred-database_code/
|            |-- fredfactors.m
|            |-- factors_em.m
|            |-- prepare_missing.m
|            |-- remove_outliers.m
|
|-- matlabtoolbox/
(emtools, emgibbsbox, emstatespace, etc.)
```

4 Data and Pre-processing

4.1 Source Data

The forecasting application uses the **FRED-QD** database of McCracken & Ng.

The file:

- `data/current.csv` contains the full quarterly macro dataset.

Data loading and transformation are handled by:

```
[data,dates,series,tcode] = fred_qd2(’current.csv’, select, station, stationY);
```

4.2 Variables Used

Endogenous variables:

- `GDPC1`
- `PCECTPI`
- `FEDFUNDS`

Transformations:

- GDP: log-difference (tcode = 5)
- PCE: log-difference (tcode = 5)
- FFR: first difference (tcode = 2)

These are supplied in the main script as:

```
stationY = [5,5,2];
station  = 0;
standarize = 1;
```

4.3 Instruments (Drivers) Z

Selected instruments:

- S&P 500,
- S&P PE ratio,
- PERMITS,
- PPIIDC,
- BAA10YM,
- TOTALSLx,
- EXJPUSx,
- VIXCLSx,
- UMCSENTx.

Matched via:

```
[~, idx_selected] = ismember(selectz, series);
Z = data(:, idx_selected);
```

5 Models Implemented

This section describes the nine forecasting models used in the empirical application. For each model we list the MATLAB function implementing it, the precise input flags (e.g., `'SV'`, `'CV'`, `'CL'`), and a description of all scalar inputs (`h`, `p`, `r`, `nsave`, etc.) used by the functions.

Notation for Inputs Used in All Models

- *Y_sample*: matrix of endogenous variables up to the current forecast origin.
- *Z_sample*: matrix of instruments (economic drivers) up to the forecast origin.
- h: forecasting horizon, i.e., maximum number of steps ahead to forecast (U.S.: $h = 24$, Euro Area: $h = 8$).

- p: VAR lag length.
- r: number of latent factors in the factor-stochastic-volatility block ($r = 1$ in our illustration).
- $nsave$: number of posterior draws saved.
- $nburn$: burn-in draws discarded.
- $nthin$: thinning interval used to reduce autocorrelation of the MCMC chain.
- yf_true: ex-post realizations y_{t+h} used to compute forecast errors.
- $YFact$: augmented dataset $[Y, F_Y]$ used in FAVAR-type models.

All models return a $n \times h \times$ `nsave` array of predictive simulations `yfore_save` from which quantiles, means, and scores are computed.

Model 1: AVP–VAR (function: `APVAR.m`)

- Adaptive-parameter VAR where all innovations to β_t are driven by observed instruments Z.
- Stochastic or constant volatility selected by the final flag.

```
[yfore_save] = ...
APVAR(Y_sample, Z_sample, h, p, r, nsave, nburn, nthin, 'SV');
```

Interpretation of the string argument:

- `'SV'` : stochastic volatility via Gaussian-mixture SV.
- `'CV'` : constant homoskedastic disturbances.

Model 2: CP–VAR (function: `TVP_VAR.m` with `'CP','CL','CV'`)

- Standard constant-parameter VAR implemented within the TVP framework.
- All three blocks (coefficients, loadings, volatility) are held constant.

```
yfore_save = TVP_VAR(Y_sample, h, p, r, nsave, nburn, nthin, ...
'CP', 'CL', 'CV');
```

Flags:

- `'CP'` : constant parameters.
- `'CL'` : constant factor loadings.
- `'CV'` : constant volatility.

Model 3: TVP–VAR–EB (function: `TVP_RW_EB.m`)

- A Primiceri-style TVP-VAR estimated using empirical Bayes hyperparameters.
- Always includes stochastic volatility and random-walk states by construction.

```
yfore_save = TVP_RW_EB(Y_sample, p, nsave, nburn, h);
```

Model 4: CP–VAR–SV (function: `TVP_VAR.m` with `'CP','TVL-RW','SV'`)

- Regression coefficients are constant.
- Loadings follow a random walk.
- Stochastic volatility.

```
yfore_save = TVP_VAR(Y_sample, h, p, r, nsave, nburn, nthin, ...
'CP', 'TVL-RW', 'SV');
```

Flags:

- `'CP'` : constant regression coefficients.
- `'TVL-RW'` : time-varying loadings (random walk).
- `'SV'` : stochastic volatility.

Model 5: OLS VAR (function: `BVAR_OLS_iter.m`)

- Homoskedastic constant-parameter VAR estimated by OLS.
- Used as benchmark for MAE, MSPE, MSFE, and quantile-score ratios.

```
yfore_save = BVAR_OLS_iter(Y_sample, p, h, nsave);
```

Model 6: VAR–SVO-t (function: `CCMM_SVO.m`)

- Implements Carriero–Clark–Marcellino–Mertens (2023):
  - heavy-tailed measurement errors,
  - discrete outlier states,
  - stochastic volatility.

```
yfore_save = CCMM_SVO(Y_sample, sample_index, dates_sample, ...
p, yf_true', h);
```

Inputs:

- `sample_index`: current forecast origin.
- `dates_sample`: MATLAB datenums for time stamps.

Model 7: FAVAR (functions: `extract.m` + `BVAR_OLS_iter.m`)

- One PCA factor extracted from standardized drivers Z.

```
[FY] = extract(zscore(Z_sample), 1);
FY   = FY / chol(cov(FY)) - mean(FY / chol(cov(FY)));
YFact = [Y_sample, FY];

yfore_save = BVAR_OLS_iter(YFact, p, h, nsave);
```

Inputs:

- `r=1` factor.
- `YFact` contains endogenous variables plus extracted factor.

Model 8: FAVAR–SV (function: `TVP_VAR.m`)

- Same factor as Model 7, but estimated with SV and TV loadings.

```
yfore_save = TVP_VAR(YFact, h, p, r, nsave, nburn, nthin, ...
'CP', 'TVL-RW', 'SV');
```

Inputs identical to Model 4, but applied to the augmented dataset.

Model 9: TVP–VAR–FB (function: `TVP_VAR_FB.m`)

- Full Bayesian TVP–VAR with:
  - time-varying coefficients,
  - time-varying loadings,
  - stochastic volatility,
  - Horseshoe shrinkage priors.

```
yfore_save = TVP_VAR_FB(Y_sample, h, p, r, nsave, nburn, nthin);
```

6 Forecasting Design

The first 50% of the usable observations form the initial window.

At each forecast origin:

1. estimate the model,
2. generate draws for all $h = 1, \dots, 8$,
3. compute MAE, MSPE, MSFE, quantile scores, PITs, and predictive likelihood.

7 Main Script: `main_forecasting_FREDQD.m`

7.1 Execution Flow

1. add paths;
2. load FRED-QD data;
3. transform and standardize variables;
4. loop over $p = 1, 2, 3, 4$;
5. estimate Models 1–9 at each forecast origin;
6. save:

```
oos_FREDQD_p1.mat, ..., oos_FREDQD_p4.mat
```

7.2 MCMC Settings

- `nsave` = 5000,
- `nburn` = 1000,
- `nthin` = 5.

8 Generating Figures and Tables

8.1 Step 1: Produce Forecasts

```
>> main_forecasting_FREDQD
```

8.2 Step 2: Figures

```
>> Figures_oos
```

Saves PDF figures in `Appendix_Figures/`.

8.3 Step 3: Tables

```
>> Appendix_Forecast_FREDQD
```

Creates MSPE tables and robustness tables for all p.

9 Mapping Between Scripts and Outputs

Output	Description	Script
OOS forecasts	Predictive distributions	`main_forecasting_FREDQD.m`
MAE, MSPE ratios	Main figures	`Figures_oos.m`
FRED-QD tables	VAR(p) for $p = 1$–4	`Appendix_Forecast_FREDQD.m`
Supplement Figures	Full forecasting figures	`Figures_oos.m`