ReadMe File for Replication Package

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

Empirical Forecasting Application: U.S. Monthly Data

1 Purpose and Scope

This document provides detailed instructions to replicate the **U.S. monthly forecasting** results presented in the paper and its online supplement. The replication package generates:

- out-of-sample (OOS) forecasts for industrial production and inflation,
- forecast evaluation metrics (MAE, MSPE, MSFE, quantile scores),
- full recursive expanding-window estimation for $p \in \{1, 2, 3, 4\}$,
- figures and tables corresponding to the empirical results.

This ReadMe focuses exclusively on the **U.S. monthly application**. Separate ReadMe files are provided for the Euro Area, the FRED-QD application, and the Monte Carlo simulations.

2 Software and Requirements

2.1 MATLAB Environment

The code is written for MATLAB and has been tested on:

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

External helper functions are included within:

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

These folders are automatically added to the MATLAB path by the main script. Full replication may take several hours depending on hardware.

3 Directory Structure

The replication package for the U.S. monthly application contains:

```
.  (root folder)
|-- main_forecasting_US.m
|
|-- data/
|   |-- USData_Updated.xlsx
|   |-- Z.xlsx
|
|-- Forecasting_Results/
|   |-- Appendix_Forecasting_US.m
|   |-- Figures_oos.m
|   |-- Tables_oos.m
|   |-- Tables_oos_function.m
|   |-- Table_5.m
|   |-- oos_US_p1.mat
|   |-- oos_US_p2.mat
|   |-- oos_US_p3.mat
|   |-- oos_US_p4.mat
|   |
|   |-- Appendix_Figures/
|       |-- Appendix_Forecasting_US_p1.pdf
|       |-- Appendix_Forecasting_US_p2.pdf
|       |-- Appendix_Forecasting_US_p3.pdf
|       |-- Appendix_Forecasting_US_p4.pdf
|
|-- functions/
|   |-- APVAR.m
|   |-- TVP_VAR.m
|   |-- TVP_VAR_FB.m
|   |-- TVP_RW_EB.m
|   |-- BVAR_OLS_iter.m
|   |-- CCMM_SVO.m
|   |-- load_data_Instruments.m
|   |-- extract.m
|   |-- getOLS.m
|   |-- carter_kohn.m
|   |-- carter_kohn2.m
```

```
|   |-- draw_alpha.m
|   |-- draw_beta.m
|   |-- draw_sigma.m
|   |-- SVRW*.m
|   |-- UCSV*.m
|   |-- pctile.m
|   |-- quantile.m
|   |-- additional helper functions
|
|-- matlabtoolbox/
|-- emtools/
|-- emtexbox/
|-- emgibbsbox/
|-- emeconometrics/
|-- emstatespace/
```

4 Data and Pre-processing

4.1 Files

- `USData_Updated.xlsx`: monthly U.S. macroeconomic series, including:
  - INDPRO (industrial production),
  - PCEPI (price index),
  - FEDFUNDS (policy rate).
- `Z.xlsx`: monthly instruments used as economic drivers in the AVP–VAR model.

Data are loaded using:

```
[Y,Z,T,varnum,data,dataZ,dates,varnames] =
load_data_Instruments(select,selectZ,tcode,tcodeZ,standard,standardZ);
```

4.2 Transformations and Standardization

The following convention applies:

- **tcode = 1**: level
- **tcode = 2**: first difference
- **tcode = 5**: log-difference

In the main script, the default transformation is:

- INDPRO: log-difference
- PCEPI: log-difference
- FEDFUNDS: first difference

Drivers Z in this application are included in levels. If `standard = 1`, series are standardized to mean zero and variance one, improving numerical stability in TVPs.

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

6.1 Expanding-Window Scheme

The first 50% of the available sample (excluding the final h periods) is used as the initial estimation window.

For each forecast origin t:

1. Estimate the model using data 1:t.
2. Produce predictive distributions for different forecasting horizons $h = 1, \ldots, 24$.
3. Store quantiles, MAE, MSPE, MSFE, and quantile scores.

6.2 Quantile Evaluation

Quantiles evaluated:

$$\tau \in \{0.10,\ 0.25,\ 0.50,\ 0.75,\ 0.90\}.$$

Quantile score for horizon h and variable i:

$$\mathrm{QS}_\tau(t) = \left(y_{t+h,i} - \hat{q}_\tau\right)\left(\mathbf{1}\{y_{t+h,i} \leq \hat{q}_\tau\} - \tau\right).$$

7 Main Script: `main_forecasting_US.m`

7.1 Execution Flow

1. Add paths to toolboxes and functions.
2. Define endogenous variables, driver set, transformation codes, and standardization.
3. Loop over $p \in \{1, 2, 3, 4\}$.
4. Load and transform U.S. data via `load_data_Instruments`.
5. Run recursive expanding-window forecasting for all nine models.
6. Save results as:

   ```
   oos_US_p1.mat, ..., oos_US_p4.mat
   ```

7.2 MCMC Settings

Default:

- `nsave` $= 5000$ posterior draws,
- `nburn` $= 1000$ burn-in,
- `nthin` $= 5$ thinning interval.

8 Generating Figures and Tables

8.1 Step 1: Produce Forecasts

Run from the root folder:

```
>> main_forecasting_US
```

8.2 Step 2: Figures (MAE and Quantile Score Ratios)

In the folder `Forecasting_Results/`, run:

```
>> Figures_oos
```

This script:

- loads the `oos_US_pX.mat` files,
- computes performance ratios relative to OLS,
- saves PDF figures to `Appendix_Figures/`.

8.3 Step 3: Tables (MSPE, Computational Cost, Robustness)

Run:

```
>> Appendix_Forecasting_US
```

This script:

- loads results for each p,
- computes MSPE ratios for INDPRO and PCEPI,
- calls `Tables_oos` to generate all forecasting tables.

9 Mapping Between Scripts and Outputs

Output	Description	Script
OOS forecasts	Raw predictive distributions	`main_forecasting_US.m`
MAE, MSPE ratios	Main paper figures	`Figures_oos.m`
Table 2	MSPE for IP and PCEPI	`Table_2.m`
Tables of Section 4.2 (supplement)	Robustness by lag order	`Appendix_Forecasting_US.m`
Supplement Figures	Full set of figures	`Figures_oos.m`