**Master in Data & Finance**

**RESEARCH PAPER**

*Academic Year 2024–2025*

# Time varying machine learning: Adaptive Regularization in Ridge Regression

**Lucien Kodeli and Evgenii Kostromin**

Under the supervision of Prof. Tjeerd de Vries

President of the Jury: Prof. Ahmed Kebaier

Keywords: Adaptive Regularization; Ridge Regression; Time-Varying Hyperparameters; Financial Machine Learning; Asset Pricing; Returns Prediction

⊠ PUBLIC REPORT □ CONFIDENTIAL REPORT

# Time Varying Machine Learning: Adaptive Regularization in Ridge Regression

**Lucien Kodeli and Evgenii Kostromin**

## Abstract

This thesis investigates the performance of different methods of dynamic selection of Ridge regularization strength in cross-sectional return prediction. We evaluate several adaptive strategies: Nested Walk-Forward Cross-Validation, Follow-the-Leader Online Expert Learner, Gaussian Process Thompson Sampling Bandit, and a Meta-Learner. Results show that optimal Ridge penalty hyperparameter changes over time, particularly post-2000 when return predictability declines as noise increases proving the necessity of adaptive methods in return prediction. We show that the Gaussian Process bandit approach offers the best trade-off between predictability, speed, and profitability, while the meta-learner delivers the highest predictive $R^2$ albeit not materializing into solid economic performance.

# Time Varying Machine Learning: Adaptive Regularization in Ridge Regression

## 1 Introduction

Predicting cross-sectional stock returns is a central challenge in empirical finance, complicated by the inherently low signal-to-noise ratio and the dynamic nature of financial markets. This thesis focuses on the critical role of adapting model regularization over time, particularly the Ridge regression penalty hyperparameter, with the goal of improving predictive accuracy and economic relevance in the face of evolving market regimes. Rather than assuming a static model, we examine whether allowing the regularization strength to vary, through strategies like online expert learning, Gaussian Process (GP) bandits, and meta-learners, leads to more resilient and effective predictions.

Through these, this thesis introduces several methodological innovations to the problem of dynamic regularization in cross-sectional return prediction. First, we develop a fully rolling and expanding-window validation framework that mimics the real-time information flow set of investors, avoiding lookahead bias and better reflecting out-of-sample uncertainty. Secondly, for the purpose of a model called meta-learner, we build a rich meta-dataset by augmenting the number of training tasks through varying window lengths, enabling the model to generalize across diverse market regimes. Third, we compare a range of adaptive strategies, including online expert learners, GP bandits, and meta-learners, under a unified evaluation pipeline using both statistical and economic criteria. Lastly, we contribute a rigorous assessment of how predictive performance translates (or fails to translate) into portfolio outcomes, particularly in environments where return predictability is structurally declining.

We evaluate these methods using a rich panel of monthly U.S. stock returns and firm characteristics from January 1925 to June 2019. Cross-sectional $R^2$ serves as the main metric of predictive accuracy, complemented by economic measures such as portfolio returns and Sharpe ratios. Our modeling framework emphasizes the importance of dynamic tuning, computational efficiency, and adaptability to structural breaks; especially around the year 2000, when return predictability deteriorates.

This deterioration is not purely statistical but likely driven by economic forces: as simple return-predictive strategies became widely adopted by hedge funds and systematic investors, the exploitable alpha embedded in commonly used firm characteristics eroded. As more capital chased these signals, the return patterns they once captured were arbitraged away, leading to noisier and less stable relationships. This highlights the importance of models that can adapt in real-time to changing market dynamics and remain effective even as the underlying data environment becomes more competitive and less informative. This work contributes to the growing literature on dynamic

modeling in empirical asset pricing, particularly in contexts where alpha decays over time and the forecasting environment evolves rapidly.

Our results show that several adaptive strategies outperform static Ridge regression, particularly in adapting to shifting market regimes. The GP bandit achieves the highest Sharpe ratio (0.852), robust returns (3.30%), and a predictive out-of-sample $R^2$ of 0.61%. The meta-learner, while delivering the highest statistical fit ($R^2 = 0.76\%$), generates more moderate economic performance (return of 2.63%, $Sharpe = 0.770$) compared with the expert learning and GP bandit models, reflecting the sometimes imperfect alignment between predictive accuracy and portfolio outcomes. In addition, it is much more computationally expensive than all other models. Simpler strategies like the expanding-window expert learner remain competitive, whereas the rolling-window expert underperforms in both dimensions. Overall, adaptivity improves model robustness, but the best-performing methods are those that balance responsiveness with stability.

To sum up, the expanding-window expert learner, GP bandit, and meta-learner offered the best combinations of statistical precision and economic performance, consistently adjusting to evolving market conditions. The GP bandit's strength lies in its principled balance between exploration (i.e. trying less-tested $\lambda$ values to gather information) and exploitation (i.e. using $\lambda$ values that have performed well so far), enabling it to quickly adjust to shifts in return dynamics without overfitting. The meta-learner, despite achieving the highest $R^2$, lagged slightly in returns due to its reliance on a longer learning phase, which delayed economic benefits. Methods like the rolling-window expert learner and nested cross-validation underperformed because they are very reactive to local performance information and suffer from high variance, making them less effective in the presence of structural breaks or weak signal environments.

We observe that adaptive models are able to efficiently detect regime shifts and adjust shrinkage intensity, accordingly, preserving economic value even when statistical fit declines. These findings underscore the importance of time-aware model tuning in financial machine learning and suggest that robust asset pricing models must evolve in tandem with the market conditions they aim to exploit.

## 2 Literature Review

Stock return predictability exhibits fundamental time variation, challenging traditional static modeling approaches. Early studies established theoretical foundations for time-varying risk premia (Merton, 1973) and documented predictive relationships using variables like dividend yields and interest rates (Fama and Schwert, 1977; Campbell and Shiller, 1988). However, subsequent research revealed these relationships are unstable across economic regimes (Goyal and Welch, 2008), with apparent predictability often disappearing in rigorous out-of-sample tests (Bossaerts and Hillion, 1999). This instability stems from structural breaks in financial markets (Stock and Watson, 1996; Paye and Timmermann, 2006), necessitating models that adapt to evolving market conditions.

Initial solutions to non-stationarity included rolling-window estimation and exponential forget-

ting factors (Pesaran and Timmermann, 2007; Hyndman et al., 2008). These methods continuously update coefficient estimates but maintain static regularization. Parallel statistical developments established shrinkage foundations: Lindley and Smith (1972) demonstrated Bayesian priors induce ridge-like shrinkage, presaging modern regularization techniques. The core insight that optimal shrinkage depends on prevailing market conditions remained, however, underdeveloped in financial applications.

As machine learning entered finance, researchers adapted regularization methods for non-stationary environments. Simple extensions include rolling-window LASSO (Koo et al., 2020) and kernel-weighted time-varying LASSO (Kapetanios and Zikes, 2018). More sophisticated approaches emerged as well: Monti et al. (2018) developed an adaptive regularization scheme which infers an adaptive Lasso penalty online. By monitoring recent performance, their algorithm adjusts the penalty parameter to track regime changes, allowing both coefficients and the degree of shrinkage to adapt over time. Yousuf and Ng (2019) extended these ideas to boosting, introducing a boosting-based method that focuses on recent observations at each iteration, effectively performing variable selection and coefficient estimation in one stage under non-stationarity. A key theoretical advance by Goulet Coulombe (2023) established duality between time-varying parameter (TVP) models and ridge regression with temporal smoothing penalties.

Surveys like Nagel (2021) document the recent explosion of ML applications in asset pricing, emphasizing that accounting for non-stationarity is essential when applying complex ML methods to financial data. In particular, Nagel (2021) focuses on using the past returns data for future returns prediction, emphasizing the importance of incorporating prior knowledge on the features by scaling the covariates matrix in Ridge regression. While Nagel (2021) shows that the task of choosing the optimization parameter for cross-validation is non-trivial for return prediction, he uses a leave-one-year-out method for hyper-parameter tuning, thus not incorporating the time-variation methods in his research.

A landmark study by Gu et al. (2020) conducted a horse-race of ML techniques—ridge, lasso, elastic net, trees, random forests, and neural networks—to predict U.S. stock returns, using an expanding-window re-training approach. They report large economic gains (e.g. an annualized out-of-sample Sharpe ratio of 0.77 for a neural network market-timing strategy vs. 0.51 for the historical mean) and show that nonlinear methods capture interactions and adapt through re-fitting, though all models agree on key predictors such as momentum and volatility.

Parallel advances occur in cross-sectional ML for factor construction (e.g. Kelly et al. 2019) and asset allocation (e.g. Routledge 2019), with frequent re-estimation to accommodate fading anomaly payoffs. Nagel and Xu (2019) propose an "asset pricing with fading memory" framework in which investors assign exponentially decaying weights to past data—behavior that corresponds to econometricians using weighted or rolling estimators.

Finally, econometric tests for episodic predictability (e.g. Giroud and Mueller 2019) complement adaptive ML methods by detecting when a time-varying model is needed versus when a simpler static model suffices. The state of the art combines real-time detection of regime changes with

adaptive modeling—whether via regime-switching, TVP, or dynamic penalties—to yield robust return forecasts.

## 3 Problem Statement

Our study investigates *time-dependent hyperparameter selection* for Ridge regression, building directly on the experimental design of Nagel (2021). Our study begins from the *return-prediction experiment* in Chapter 3 of Nagel (2021). The goal is to approximate conditional expected excess returns of an individual stock $i$,

$$\mathbb{E}[r_{i,t+1} \mid \boldsymbol{x}_{i,t}] \;=\; f(\boldsymbol{x}_{i,t}) \tag{1}$$

where the predictor vector $\boldsymbol{x}_{i,t}$ is built *solely* from the stock's own past monthly returns observed at time $t$, and used to predict the return at time $t+1$ (Nagel, 2021). The first lag is skipped to prevent micro-structure noise and short-horizon biases from contaminating the analysis. Specifically, the linear model

$$r_{i,t+1} = \sum_{k=1}^{119} b_k\, r_{i,t-k} + \sum_{k=1}^{119} c_k\, r_{i,t-k}^2 + \sum_{k=1}^{119} d_k\, r_{i,t-k}^3 + \varepsilon_{i,t+1} \tag{2}$$

uses *three sets of 119 lags*—levels, squares, and cubes—yielding 357 regressors.

Estimation uses the *ridge* estimator,

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \arg\min_{\boldsymbol{\beta}} \Big\{ \|\mathbf{r} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \;+\; \lambda\, \|\boldsymbol{\beta}\|_2^2 \Big\} \tag{3}$$

where

- $\mathbf{r}$ is the $(T * N) \times 1$ vector of demeaned monthly stock returns to be predicted, N being the number of firms included;
- $\mathbf{X}$ is the $T \times 357$ matrix whose columns are the lagged returns, their squares, and their cubes;
- $\boldsymbol{\beta}$ is the $357 \times 1$ vector of slope coefficients to be estimated;
- $\|\cdot\|_2$ denotes the Euclidean ($L_2$) norm; and
- $\lambda > 0$ is the Ridge penalty that controls the amount of shrinkage applied to the coefficients.

The penalty $\lambda$ chosen by **leave-one-year-out** (LOYO) cross-validation. Each fold removes an entire calendar year, fits the model on the remaining years, and records the validation $R^2$. Averaging the $R^2$ across all left-out years, the algorithm selects the $\lambda$ that maximizes the cross-validated $R^2$. A key insight from Nagel (2021) is that *prior knowledge about scale* matters: unequal rescaling of higher-order lags (e.g. dividing $r_{t-1}^2$ by 2 and $r_{t-1}^3$ by 4) reduces the dominance of extreme observations and improves predictive fit. The **Unequal-Scaling, $R^2$-optimized Ridge** configuration is therefore our baseline.

### 3.1 Limitations of LOYO Cross-Validation

Although LOYO CV is intuitive and yields stable estimates when the data are roughly stationary across calendar years, it may be *misleading* in settings with pronounced time variation. LOYO permits the model to be trained on future information relative to certain test observations, thereby overstating out-of-sample performance. In our replication we obtain an apparently impressive mean $R^2$ of 1.011% using LOYO on the full sample. Given the low signal-to-noise ratio in monthly equity returns, an out-of-sample $R^2$ of just over one percent is remarkable—historically, even a few basis points of predictive $R^2$ can translate into sizable economic gains and high Sharpe ratios. Yet a simple *nested walk-forward* (Section 5.1) procedure—with a five-year rolling training window and strictly forward validation—produces a mean $R^2$ of $-0.180\%$. The 1.2 percentage-point gap underscores how LOYO can ignore temporal dependence and model drift that are endemic to financial data.

### 3.2 Walk-Forward Validation and Optimal Regularization

Walk-forward validation respects chronology: models are fit on past data and evaluated on truly unseen future periods, mirroring live deployment. Because the "right" amount of shrinkage can drift with market regimes, we compare four complementary schedulers that all honor this forward flow of information but differ in how they search for (and adapt to) the optimal penalty:

1. **Nested Walk-Forward Cross-Validation.** A transparent benchmark: for each test year we re-estimate the model on the previous five years and pick $\lambda$ via an inner LOYO loop. It is simple, leakage-free, and delivers a clean estimate of true OOS $R^2$—but can be computationally heavy and myopic to within-window noise.

2. **Follow-the-Leader Online Expert Learner.** Treat each candidate $\lambda$ as an "expert" and keep a running score of its past forecast $R^2$. The next year uses the best-scoring expert. This online rule is fast, easy to tune, and can track gradual regime shifts without re-running a full CV at every step.

3. **Gaussian Process Thompson Sampling Bandit.** Model the reward function $f(\lambda)$ with a Gaussian Process Regression to balance exploitation (use the current best $\lambda$) with cheap, targeted exploration (test only a small batch of promising alternatives each year). Correlation across nearby penalties speeds learning and keeps computation low, making it attractive when the signal is weak and the grid is dense.

4. **Meta-Learner.** A model learns the mapping from "market state" features to the historically best $\lambda$. Once trained on many past tasks, it can jump directly to an appropriate penalty when conditions change, offering a flexible, data-driven alternative to manual tuning rules.

These methods were chosen because they span the main strategies for dealing with time variation in hyperparameters: exhaustive but honest evaluation (nested CV), light-weight online adaptation

(experts learner), principled exploration–exploitation (GP bandit), and context-aware policy learning (meta-learner). For each we report genuine OOS $R^2$, Nagel-style portfolio performance, and total run time, always using only information available at the decision date.

# 4 Data Preparation

This section details the preparation of monthly U.S. equity data used in the Ridge–regression experiment of Nagel (2021, p. 35). The raw sample—downloaded from the CRSP monthly files—covers `RET` (return), `PRC` (price), `SHROUT` (number of shares), and `HEXCD` (exchange index) of the US stocks from January 1925 to June 2019. All cleaning steps are implemented in Python and ensure that the final panel is free of inconsistencies, duplicates, and unsuitable securities before the regression analysis.

## 4.1 Data Processing

We process monthly U.S. equity data from CRSP (1925-2019) following Nagel (2021)'s methodology:

- **Data cleaning**: Remove records with missing TICKER identifiers and eliminate duplicate observations based on TICKER-DATE pairs.
- **Gap handling**: Ensure continuous monthly series for each stock by inserting missing months with NaN returns to maintain temporal alignment.
- **Feature engineering**: Generate 119 lagged returns ($r_{t-2}$ to $r_{t-120}$) along with their squared and cubed terms, yielding 357 predictors.
- **Microstructure filters**: Exclude small stocks (below NYSE 20th percentile market-cap) and penny stocks (lagged price $<$ \$1), with market-cap calculated as $CAP = PRC \times SHROUT \times 1000$.
- **Standardization**: Each month, cross-sectionally demean and scale all variables using $x^*_{i,t} = (x_{i,t} - \bar{x}_t)/\sigma_t$, discarding months with single observations.
- **Weighting scheme**: Assign equal weight to each calendar month using $w_{i,t} = 1/(n_t T)$ where $n_t$ is the number of stocks in month $t$ and $T$ is total months.

## 4.2 Final Panel

The cleaned panel spans July 1972–January 2019 and contains

- demeaned and standardized returns $r_{i,t+1}$,
- lags `RET_lag_`$k$, $k = 2, \ldots, 120$,
- quadratic and cubic lags `RET2_lag_`$k$, `RET3_lag_`$k$,

- equal-weight monthly observation weights.

Note that observations for January 1970–June 1972 are absent after gap-filling, so regression results may differ marginally from those in Nagel (2021, p. 39).

Table 1: Descriptive statistics for a representative subset of predictors[1]

| | RET | RET_lag_2 | RET_lag_3 | RET2_lag_2 | RET2_lag_3 | RET3_lag_2 | RET3_lag_3 | weight |
|---|---|---|---|---|---|---|---|---|
| Count | 738421 | 738421 | 738421 | 738421 | 738421 | 738421 | 738421 | 738421 |
| Mean | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.000001 |
| Std | 0.0933 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0 |
| Min | −1.0489 | −12.0166 | −12.0620 | −1.3798 | −1.3833 | −39.5163 | −39.6498 | 0.000001 |
| 25% | −0.0468 | −0.5351 | −0.5362 | −0.3270 | −0.3335 | −0.1096 | −0.1131 | 0.000001 |
| 50% | −0.0029 | −0.0497 | −0.0491 | −0.2087 | −0.2105 | −0.0522 | −0.0512 | 0.000001 |
| 75% | 0.0424 | 0.4703 | 0.4733 | −0.0253 | −0.0239 | 0.0070 | 0.0130 | 0.000002 |
| Max | 4.1597 | 37.8557 | 37.5795 | 46.9376 | 46.9620 | 46.9678 | 47.0098 | 0.000003 |

### 4.3 Regression Results

A simple OLS regression with equal scaling and no regularization ($\lambda = 0$) attains a respectable *in-sample* $R^2$ of 5.80 %, yet its leave-one-year-out cross-validated (out-of-sample) $R^2$ collapses to −1.76 %, a textbook case of over-fitting. Re-estimating the same model with the *unequal-scaling Ridge* specification of Nagel (2021) and tuning $\lambda$ by the same LOYO procedure boosts the out-of-sample $R^2$ to 1.01 %. This result is yet significantly lower than the in-sample $R^2$ of 3.13 %, emphasizing the importance of using the out-of-sample measures in return prediction.

Translating these forecasts into a trading strategy via Nagel's portfolio-construction rule—weights proportional to the predicted returns,

$$\hat{\boldsymbol{\omega}}_{t-1} = \frac{\hat{\boldsymbol{\mu}}_{t-1}}{\sum_{i=1}^{N} |\hat{\mu}_{i,t-1}|} \tag{4}$$

and evaluated with the same leave-one-year-out folds—yields[2]:

- cross-validated mean portfolio return: 4.11%,
- cross-validated standard deviation: 3.88%,
- cross-validated Sharpe ratio: 1.0607.

Thus, ridge regularization not only corrects the negative out-of-sample $R^2$ of the OLS benchmark but also delivers economically meaningful performance when implemented as a dollar-neutral long–short portfolio.

[1] The full data set contains 360 standardized predictor columns—lags 2–120 in levels, squares, and cubes. The table reports only eight variables plus the observation weight to illustrate typical scale and distribution.

[2] All portfolio statistics reported below are annualized.

# 5 Methodology

While financial return series are often modeled as weakly stationary, their conditional moments—such as risk premia—can evolve over time with business-cycle conditions, changes in liquidity, or arbitrage pressures. This time variation implies that a hyperparameter tuned on the distant past may become suboptimal in the present as market dynamics shift. Motivated by these regime shifts, this paper compares a suite of *time-dependent hyperparameter-tuning* schemes for return-prediction Ridge-based models under a consistent walk-forward evaluation.

We therefore assess each method along three dimensions: (i) predictive accuracy measured by genuine out-of-sample $R^2$, (ii) computational efficiency, and (iii) economic value captured through Nagel-style long–short portfolio metrics. The present section formalizes and contrasts the techniques—nested cross-validation with leave-one-year-out folds, two variants of an online expert learner with memory decay, a GP bandit, and a meta-learner providing the mathematical notation and intuition required for the empirical tests that follow.

## 5.1 Nested Walk-Forward Cross-Validation

First method we adopt is a *nested walk-forward* scheme whose main virtue is its **simplicity**: it uses a similar to Nagel's LOYO Cross-Validation approach seen before, yet it delivers an estimate of *true* out-of-sample $R^2$ that is directly comparable to the one we achieved in the previous section. The procedure has two concentric loops.

**Outer (walk-forward) loop.** At calendar year $t$ we train the model on all data up to $t-1$ and evaluate it on the next unseen year $t$. We then roll the estimation window forward by one year and repeat, mimicking the information set of a real-time investor.

**Inner (LOYO) loop.** Inside each training window we tune the Ridge penalty by omitting one year at a time:

$$\mathrm{CV}(h) = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \mathcal{L}\Big(f_{-y}^{(h)},\ \mathcal{D}_y\Big) \tag{5}$$

where $\mathcal{Y}$ is the set of years in the window, $f_{-y}^{(h)}$ is the model fitted with hyper-parameter $h$ on all years except $y$, $\mathcal{D}_y$ is the left-out year , and $\mathcal{L}$ is the negative out-of-sample $R^2$ loss. The $h$ that minimizes $\mathrm{CV}(h)$—equivalently, maximizes the mean $R^2$ across left-out years—is selected and the model is re-estimated on the full window before being tested on year $t$.

This design was selected for several reasons:

- **True out-of-sample check.** Every hyper-parameter choice is validated exclusively on data that precede the test year, so the resulting $R^2$ is a genuine forward-looking metric.

- **Block structure matches market regimes.** Year-long folds expose the model to distinct macro conditions—bull, bear, crisis—without the leakage that plagues ordinary $k$-fold CV on time-series data.
- **Ease of implementation.** The method is a direct adaptation of the approach reported in Nagel (2021).

Although computationally heavy, the method remains tractable for monthly stock data and gives a robust benchmark: if a model cannot achieve a positive walk-forward $R^2$ under this protocol, it is unlikely to do so in live trading. Hence we rely on the nested LOYO metric as the definitive starting point for assessing out of sample perfomance of the unequal-scaling Ridge regression reported in Nagel (2021).

### 5.2 Online Expert Learner: Follow-the-Leader approach

**Conceptual overview.** The online expert framework reframes *hyperparameter tuning* as a sequential learning problem: each candidate hyperparameter value (e.g. a Ridge penalty $\lambda$) is treated as an *expert* that produces a forecast $\hat{y}_{i,t}$ for period $t$. After the realized return $y_t$ is observed, the algorithm learns the loss of *every* expert, $\ell(\hat{y}_{i,t}, y_t)$, and updates a weight vector $\boldsymbol{w}_t$ so that better-performing experts receive larger weight in the next round. A classical update is the *Exponentially Weighted Average* (EWA) or *Hedge* rule (Littlestone and Warmuth, 1994; Cesa–Bianchi and Lugosi, 2006):

$$w_{i,t+1} = \frac{w_{i,t} \exp[-\eta\, \ell(\hat{y}_{i,t}, y_t)]}{\sum_{k=1}^{N} w_{k,t} \exp[-\eta\, \ell(\hat{y}_{k,t}, y_t)]} \tag{6}$$

where $\eta > 0$ is a learning rate.

**Variants in the literature.** A large literature extends Hedge to handle *changing* best experts. The *Fixed-Share* algorithm of Herbster and Warmuth (1998) periodically "bleeds" weight from leaders to laggards, allowing quick recovery when market regimes flip. Adaptive learning-rate schemes such as AdaHedge (de Rooij et al., 2014) or BOA (Wintenberger, 2017) further balance stability and reactivity. Empirical finance papers (e.g. Remlinger et al., 2021) have shown that online aggregation of factor models or ML predictors can out-perform any static specification while remaining robust to structural breaks.

**Implementation in our setting.** In our study we adopt the *limit case of EWA* in which the learning rate tends to infinity, $\eta \to \infty$, so the weight mass collapses onto the single best-performing expert. This simplification—often called *Follow-the-Leader*—removes the need to fine-tune $\eta$ and keeps the implementation transparent.

In our implementation we cast each candidate Ridge penalty $\lambda_i \in \Lambda$ as an expert. Performance in year $t$ is measured by the out-of-sample coefficient of determination $r_t(\lambda_i) \equiv R^2(\lambda_i, t)$. For a scoring mechanism, we also introduce a decaying memory parameter $\rho$:

$$Score_t(\lambda_i) = \rho\, Score_{t-1}(\lambda_i) + r_t(\lambda_i), \qquad 0 < \rho \leq 1 \tag{7}$$

and select $\lambda_t^\star = \arg\max_{\lambda_i \in \Lambda} Score_t(\lambda_i)$ for the next-period forecast. Setting $\rho = 1$ yields full memory; values $\rho < 1$ impose exponential forgetting so that recent performance carries more weight. We study two scheduling variants:

1. **Rolling window (5-year)** – experts are evaluated on a model trained with the *most recent* five years $(t-5, \ldots, t-1)$ and validated on year $t$. This emphasizes local market conditions but discards distant history.

2. **Expanding window** – training uses the entire history up to $t-1$, so information accumulates indefinitely. Memory decay ($\rho < 1$) prevents the score from being dominated by ancient observations, enabling adaptation without losing the benefits of a large sample.

In both cases the algorithm requires only the computation of $r_t(\lambda_i)$ for each $\lambda_i$ once per year, after which the best-scoring penalty is immediately deployed. The procedure therefore *eliminates upfront hyperparameter selection*: instead of seeking a single "optimal" $\lambda$, the learner continually re-optimizes in real time, mirroring an investor who tests and continuously balances models as new data arrive.

**Trade-offs and benefits.** Compared with nested walk-forward CV, the expert learner delivers finer granularity in adaptation, yet it demands parallel maintenance of all $N$ candidate models. Computation remains manageable for the modest grid of Ridge penalties used here ($N \ll 100$) and can be updated incrementally. If $\rho$ is set too low, the learner may overreact to noise; if too high, it may lag sudden regime shifts. With tuned decay, however, the method has a potentiality to offer a powerful balance of *simplicity*, *statistical robustness*, and *real-time adaptivity*—qualities valuable in non-stationary return-prediction tasks.

### 5.3 Bandit-Style Learner: Gaussian Process Thompson Sampling

**Conceptual overview.** A bandit algorithm treats hyperparameter choice as a *sequential exploration–exploitation* problem. At each period $t$ we must pick one "arm"—here, a Ridge penalty $\lambda$—observe its realized *reward* (the next-year out-of-sample $R^2$), and then decide which arm to pull next. Unlike the online–expert setting where we see the loss of *all* experts each round, the bandit learner only observes the arm(s) it actually evaluates. The core idea is to balance: (i) *exploitation* of penalties that have worked well so far, and (ii) *exploration* of penalties whose performance is uncertain but potentially better.

**Variants in the literature.** Bandit algorithms differ mainly in how they trade off exploration vs. exploitation and in the assumptions on the reward process. In stochastic settings, *upper–confidence* rules (UCB) choose the arm with the highest optimism-adjusted mean (Auer et al., 2002a), whereas

*Thompson sampling* draws an arm according to its posterior probability of being optimal (Thompson, 1933; Russo et al., 2018). For adversarial (non-stationary or even adversarially chosen) rewards, EXP3 randomizes with exponential weights over arms (Auer et al., 2002b). When the action space is continuous (e.g., $\lambda \in \mathbb{R}_+$) and rewards vary smoothly, *Gaussian Process bandits* such as GP-UCB (Srinivas et al., 2010) or GP–Thompson sampling (Russo and Van Roy, 2014) model $f(\lambda)$ directly and exploit correlation so that a single evaluation informs nearby $\lambda$'s.

**Implementation in our setting.** Our setting utilizes a modified Gaussian Process Thompson sampling approach to select the best preforming Ridge penalty $\lambda_t$. We model the mapping from $\lambda$ to expected one-year-ahead performance with a GP prior on $x = \log \lambda$:

$$f(x) \sim \mathcal{GP}(m, k),$$

with a squared-exponential kernel $k$ plus a white-noise term. Rewards are the annualized out-of-sample $R^2$ for year $t$, $r_t(\lambda) \equiv R^2(\lambda, t)$, linearly rescaled to $[0, 1]$ via $y_t(\lambda) = \max\{0, \min\{1, (r_t(\lambda) + 1)/2\}\}$ to stabilize GP fitting.

The procedure consists of 3 key ingredients:

1. **Pilot grid.** In the first validation year we evaluate the *entire* grid of 20 log-spaced penalties to initialize the GP with a dense, low-noise set of observations.

2. **Exploit/Explore split.** Each subsequent year: (a) we *deploy* the penalty with the highest GP posterior mean, $\lambda_t^\star = \arg\max_\lambda m_t(\log \lambda)$, for the actual forecast (exploitation), and (b) we *sample* from the GP posterior once (Thompson sampling) to pick a batch of $B$ additional penalties to test (exploration). The GP is then updated with the newly observed rewards, and the loop advances to year $t+1$. Only these $B$ penalties are re-fitted, keeping computation low.

3. **Memory decay.** To avoid stale regimes dominating the GP, observed rewards are exponentially "forgotten":

$$y_i = \rho\, y_i + (1 - \rho)\, y_t(\lambda_i), \qquad 0 < \rho \leq 1 \tag{8}$$

   with $\rho = 1$ in the baseline.

Algorithmically, this yields exactly one "production" $\lambda$ per year plus a small exploratory batch. The GP is re-fit after each round on the (decayed) reward vector, providing a smooth estimate of the reward surface over $\lambda$. Unlike pure Thompson sampling, however, the final forecast uses the posterior-mean maximiser, yielding a smoother $\lambda_t$ path and lower forecast variance.

**Trade-offs and benefits.** Compared with the Follow-the-Leader expert learner, the GP–bandit:

- **Saves computation**: after the pilot, it evaluates only a handful of penalties each year rather than the full grid.
- **Shares information**: the GP prior correlates nearby $\lambda$'s, accelerating learning relative to the other methods.
- **Produces smoother $\lambda_t$ paths**: using the posterior mean for deployment avoids the jitter that can arise from pure sampling policies.

The cost is additional modeling complexity (kernel choice, GP hyperparameters) and longer per-round updates due to GP fitting. But for our modest grid (20 penalties) and yearly frequency, these costs are minor and the approach offers a principled way to explore efficiently while staying close to the best-known penalty.

### 5.4 Meta-Learner

**Conceptual overview.** A meta-learner is a model designed to learn how to select among different configurations of another model, such as different hyperparameter values, based on contextual information. In our case, the meta-learner determines which Ridge regression penalty parameter $\lambda$ to apply in different market environments. Instead of performing a fresh grid search over $\lambda$ each time market conditions shift, we train a separate model that learns to associate market conditions with historically optimal $\lambda^*$ values.

The meta-learner operates by first dividing the historical data into a sequence of rolling tasks of fixed length (10 years in our baseline). For each task, Ridge regressions are trained using a set of candidate $\lambda$ values. However, instead of using leave-one-year-out cross-validation (LOYO), we adopt a more realistic time-based evaluation: each Ridge model is trained on the entire task window and then evaluated on its ability to predict returns in the year that immediately follows. For each $\lambda$, we compute the out-of-sample $R^2$ on this post-task year and record the result. This results in a dataset where each row corresponds to a specific task-lambda pair and contains: the task period (e.g., 1972–1982), the candidate $\lambda$, the resulting $R^2$ performance, and a set of market characteristics corresponding to the task's period (e.g., mean return, volatility, skewness, correlation, macroeconomic indicators).

Instead of selecting just the best $\lambda$ per task, we keep all candidate $\lambda$s and their performances. This richer training dataset enables the meta-learner to learn the functional relationship between the market regime and model performance across multiple $\lambda$s.

The meta-learner model itself is implemented using a Random Forest model. Random Forests are ensemble learning methods that construct a large number of decision trees during training and aggregate their predictions. They are well-suited for this task due to their robustness to overfitting, ability to capture nonlinear interactions, and resilience to feature noise. Given regime characteristics as inputs, the Random Forest learns to predict the $\lambda$ value that is likely to perform best under similar market conditions.

To ensure sufficient data for training, we use a relatively small grid of $\lambda$ values (typically 5–6), as supported by the literature. Around 30 to 40 historical tasks are generally needed for the meta-learner to generalize effectively.

**Implementation in our setting.** Rather than dividing the dataset once into a training and test phase, we extend the meta-learning framework into an expanding-window, walk-forward model. We begin by training on the first available task, evaluating $\lambda$s on the year immediately following it, and using that as the initial training point for the meta-learner. We then use the meta-learner to predict the best $\lambda$ for the next unseen task (ensuring no overlap in years), and use that $\lambda$ to train a Ridge model over the lagged returns of the preceding task and forecast the returns of the current year.

This process is repeated iteratively: at each step, we expand the meta-training dataset by one task, retrain the meta-learner, and make a prediction for the next year using the most recent non-overlapping task's features. This approach simulates the perspective of a real-time investor who updates their model annually using only past information.

**Feature Engineering and Extensions.** Once optimal $\lambda$ values have been predicted for each year, we train a Ridge model on the corresponding task (e.g., the prior 10 years of data) using the predicted $\lambda$ and use it to forecast returns for that year. This allows us to assemble a series of out-of-sample return predictions across time.

Prior to training, we enrich the dataset with both return-derived features (mean returns, volatility, dispersion, skewness, kurtosis, autocorrelation, percentage of positive returns, volatility dispersion) and macroeconomic indicators from FRED (Federal Funds Rate, 10Y Treasury yield, average investment-grade yield, inflation, and GDP growth) representing the market characteristics from which the meta-learner model will learn to pick the optimal $\lambda^*$.

To further augment the meta-training data, we construct additional tasks using shorter historical windows of 2, 3, 4, 5, 6, 7, 8, 9, and 10 years. For each task length, we recalculate all relevant features and model performance metrics. While we ultimately use a 10-year window for forecasting (as it provides a good balance between noise reduction and sufficient meta-training data), these additional tasks help increase the meta-learner's training volume and robustness. This choice reflects the trade-off between the stability gained from longer training windows and the higher number of training samples enabled by shorter ones.

# 6 Empirical Results

## 6.1 Nested Walk-Forward Cross-Validation

Figure 3 visualizes the annualized cross-validated $R^2$, the dynamically chosen Ridge penalties, and the distribution of yearly $R^2$ outcomes produced by the five-year rolling, LOYO-nested procedure. Two findings stand out.

**Predictive accuracy.** The walk-forward $R^2$ series is highly volatile and *negative on average*. The overall mean is $-0.18\%$, with virtually no difference before versus after 2000 ($-0.18\%$ vs. $-0.17\%$). Even the best individual years barely exceed 4%, while several episodes plunge below $-10\%$. Contrast this with the $\approx 1.0\%$ *positive* out-of-sample $R^2$ reported by Nagel (2021) under his leave-one-year-out *static* cross-validation. Once we enforce a strict walk-forward chronology the apparent predictive power of the unequal-scaling Ridge model disappears, confirming that the earlier result could be a form of a look-ahead optimism.

**Penalty instability.** The right-hand panel of the Figure 3 shows the selected penalty $\lambda_t$ for each validation year. Choices oscillate wildly: periods of extremely large shrinkage ($\lambda = 10$) alternate with repeated selections of very mild penalties ($\lambda \approx 0.6$) and occasional dips to the minimum grid value ($\lambda = 0.1$). Such instability suggests that the signal-to-noise ratio is too weak for the inner LOYO to pin down a persistent optimal penalty; small changes in the training window flip the preference from near-OLS to near-zero coefficients. The procedure thus trades one form of overfitting for another: it no longer peeks into the future, but it overreacts to idiosyncrasies of each five-year window.

**Computational burden.** The nested routine required approximately 1 100 seconds on a modern workstation—orders of magnitude heavier than a single LOYO run—because every outer fold triggers $|\Lambda| \times W = 20 \times 5$ inner fits. Given the modest predictive benefit (indeed, a detriment), such cost further undermines the method's appeal for this particular data set.

Table 2: Summary statistics—nested walk-forward CV

| Metric | Value |
|---|---|
| Mean CV $R^2$ (overall) | $-0.18\%$ |
| Mean CV $R^2$ (pre-2000) | $-0.18\%$ |
| Mean CV $R^2$ (post-2000) | $-0.17\%$ |
| Annualized mean portfolio return | 1.98% |
| Annualized standard deviation | 4.77% |
| Annualized Sharpe ratio | 0.416 |
| Execution time | 1 099.4 sec |

In sum, when the hyper-parameter is tuned strictly with information available *ex ante*, the unequal–scaling Ridge model fails to beat a zero-predictability benchmark and displays pronounced year-to-year instability — findings that contrast sharply with the more optimistic figures reported under Nagel (2021)'s non-rolling validation. Intriguingly, the strategy nevertheless earns a *positive* cumulative return prior to 2000, even though the mean OOS $R^2$ for that sub-sample is negative. The explanation lies in the *left-skewed* distribution of yearly $R^2$ values: a handful of extreme losses drag down the average, whereas the *median* $R^2$ is still positive. Hence a small number of bad years depress the mean statistic without eliminating the overall profitability of the trading rule. This observation reinforces Nagel's caution that the average cross-validated $R^2$ is not, by itself, a reliable

guide to the economic value of a forecasting model.

## 6.2 Follow-the-Leader Online Expert Learner

### 6.2.1 Rolling Window Modification

To gauge how the *memory–decay parameter* $\rho$ shapes performance, we run the rolling-window expert learner on a grid $\rho \in \{0.05, 0.10, \ldots, 1.00\}$ and record the annualized walk-forward $R^2$ for each value.[3] Figure 4 collects four diagnostics: overall $R^2$ as a function of $\rho$ *(top-left)*, the same metric split before/after 2000 *(top-right / bottom-left)*, and execution time *(bottom-right)*.

**Sensitivity to memory decay.** Performance is highly non-linear in $\rho$. Very short memory ($\rho < 0.4$) yields negative or near-zero $R^2$, mirroring the instability seen in nested CV. As $\rho$ passes 0.5, accuracy improves monotonically, peaking at $\rho^\star = 0.95$ with an overall walk-forward $R^2$ of 0.23%. Pushing memory to the limit ($\rho \to 1$) causes a mild drop, indicating that *some* forgetting is helpful but wholesale discarding of past information is harmful.

**Regime dependence.** A closer look at the split samples reveals why. Before 2000, the $R^2$ curve is *strictly increasing*: the more past information the learner remembers, the higher the forecast precision, peaking above 0.4% when memory is almost full. After 2000, however, the curve goes the other way: $R^2$ stays just below zero for moderate decay but collapses once $\rho > 0.8$, turning strongly negative as stale information overwhelms the weak contemporary signal. In other words, a long memory is beneficial in the earlier regime—when past-return patterns persist—but becomes a liability once the predictive content of returns fades. The optimal $\rho^\star = 0.95$ therefore represents a compromise: it keeps enough history to exploit pre-2000 predictability while retaining just enough forgetting to limit the damage in later years.

**Computational cost.** Execution time is modest—about 305 seconds at $\rho^\star$—roughly one-quarter of the nested-CV run. Because the expert learner avoids an inner cross-validation loop and updates weights online, its complexity is essentially linear in the number of candidate penalties and validation years.

Fixing the memory–decay parameter at its empirical optimum, $\rho^\star = 0.95$, yields a cleaner picture of the expert learner's behavior. The year–by–year $R^2$ in Figure 5 confirms a *volatile* pattern of predictability—sporadic spikes above 3–4% offset by frequent dips below −3%. Relative to the nested-CV benchmark, the expert learner now delivers a small but *positive* average $R^2$ (Table 3), although annualized Sharpe ratio remains around 0.40, the value that is even lower than before. Nearly all of the forecast efficacy is concentrated in the pre-2000 era; post-2000 performance remains negative, indicating that the information embedded in past returns continues to fade in more recent decades.

[3] Each model is trained on the most recent five calendar years and validated on the next year, with experts corresponding to the same grid of Ridge penalties as in the nested-CV experiment.

Table 3: Summary statistics—rolling-window expert learner ($\rho^\star = 0.95$)

| Metric | Value |
|---|---|
| Mean CV $R^2$ (overall) | 0.23% |
| Mean CV $R^2$ (pre-2000) | 0.45% |
| Mean CV $R^2$ (post-2000) | −0.20% |
| Annualized portfolio return | 1.90% |
| Annualized st. deviation | 4.77% |
| Annualized Sharpe ratio | 0.399 |
| Execution time | 304.7 sec |

The cumulative-return curve suggests that, once transaction costs are ignored, the strategy compounds capital almost exclusively during 1980–2000 and remains largely flat thereafter. Meanwhile the selected penalty $\lambda_t$ follows a stair-step pattern, rising gradually from $\approx 0.6$ to 2.7. This upward drift signals a *deterioration of the predictive signal*: as lagged returns lose informational content, the learner must apply progressively stronger Ridge shrinkage to avoid fitting noise.

**Statistical comparison with Nested CV.** To assess whether the rolling expert learner's year-by-year $R^2$ improvements over the nested–CV benchmark are statistically significant, we perform a Diebold–Mariano (DM) test on the fold-wise squared errors of the two strategies. Let $e_t^{\text{roll}}$ and $e_t^{\text{nest}}$ be the squared forecast errors in year $t$ for the rolling-window expert learner and nested-CV, respectively, and define the loss differential

$$d_t \;=\; e_t^{\text{roll}} \;-\; e_t^{\text{nest}}, \quad t = 1, \ldots, T \tag{9}$$

The null hypothesis,

$$H_0 : \; \mathbb{E}[\, d_t \,] = 0,$$

states that both methods have equal predictive accuracy, against the two-sided alternative $H_1$ : $\mathbb{E}[d_t] \neq 0$. The DM statistic is

$$\text{DM} = \frac{\bar{d}}{\sqrt{\widehat{\text{Var}}(\bar{d})}} \tag{10}$$

where $\bar{d} = (1/T)\sum_t d_t$ and $\widehat{\text{Var}}(\bar{d})$ is the sample variance of $d_t$ divided by $T$. In our sample ($T = 42$ folds) we find

$$\bar{d} \approx -2.70 \times 10^{-6}, \quad \text{DM} \approx -1.74, \quad p\text{-value} \approx 0.08.$$

Since $|\text{DM}| < 1.96$, we cannot reject $H_0$ at the 5% level (though it is marginal at 10%). Thus there is no strong evidence that the rolling expert learner's accuracy differs from the nested-CV benchmark.

Overall, although the rolling expert learner has a higher average $R^2$, we cannot confirm that it dominates the strictly nested walk-forward CV in accuracy. Nevertheless, it significantly outperforms the previous method in speed, while its investment performance (0.4 Sharpe ratio) remains

the same.

The post-2000 deterioration of both methods, however, raises doubts about their robustness. The evidence further reinforces the previous conclusions: once predictive models are evaluated in *genuinely* out-of-sample conditions, the incremental value of past-return information is positive but decidedly modest, and heavily dependent on the prevailing market regime.

### 6.2.2 Expanding Window Modification

As with the rolling variant, we first sweep the memory–decay parameter $\rho$ on the expanding-window learner.[4] Figure 6 shows that predictive accuracy again exhibits a hump-shaped response, but the peak is markedly higher: overall walk-forward $R^2$ attains 0.67% at $\rho^\star = 0.85$. Execution time rises to $\approx 2\,545$ seconds—an eight-fold increase over the rolling learner— because each iteration refits on an ever larger sample.

**Regime differences.** Using an expanding window with $\rho^\star = 0.85$ boosts the pre-2000 $R^2$ to 1.16%, roughly three times larger than the rolling version. After 2000 the accuracy figure turns negative ($-0.29\%$), yet the cumulative-return plot in Figure 7 shows that the strategy keeps rising and finishes above $\times 4.5$ the initial capital. In other words, the broader data set still earns money in recent years even though its statistical $R^2$ has faded. The rolling learner could not do this: once the signal weakened it mostly moved sideways.

**Penalty dynamics.** Selected penalties drift upwards more steeply than in the rolling case, reaching $\lambda \approx 3.7$ by the late 2010's (Figure 7, top-right). Because the estimation sample grows every year, the model needs stronger shrinkage to guard against fitting noise from the large, sometimes outdated, history.

**$R^2$ distribution.** Switching from a rolling to an expanding window changes the shape of the year-by-year $R^2$ histograms in a noticeable way. With the rolling learner the pre-2000 histogram is narrow, centered just above zero and punctuated by a few outliers above 3%; the post-2000 histogram shifts decisively to the left, clustering below the zero line. For the expanding learner the pre-2000 mass moves rightward and thickens between 1% and 4%, signaling that positive years occur more regularly. After 2000 the distribution still drifts left of zero, but it is broader and even retains a handful of small positive bars—features almost absent under the rolling rule. Taken together, the histograms show that the expanding window makes good years *more* reliably good while not eliminating the negative drag in the later period. This is could also be seen on the $R^2$ graph in Figure 7: $R^2$ becomes more volatile post-2000 comparing to the rolling-window case.

Overall, keeping all past data in the training set helps to improve the investment performance delivering an annualized return of 3.25% with a Sharpe of 0.84. The price is computing time

[4] Here the training sample grows cumulatively, so the expert scores reflect *all* history, discounted by $\rho^{t-s}$.

Table 4: Summary statistics—expanding-window expert learner ($\rho^\star = 0.85$)

| Metric | Value |
|---|---|
| Mean CV $R^2$ (overall) | 0.67% |
| Mean CV $R^2$ (pre-2000) | 1.16% |
| Mean CV $R^2$ (post-2000) | −0.29% |
| Annualized portfolio return | 3.25% |
| Annualized st. deviation | 3.88% |
| Annualized Sharpe ratio | 0.837 |
| Execution time | 2 544.8 sec |

(about eight times slower than the rolling setup). This trade-off suggests that expanding windows significantly out-perform the previous methods, especially if accompanied by a fine-tuned memory decay parameter.

### 6.3 Gaussian Process Thompson Sampling Bandit

We now turn to the Gaussian–Process Thompson bandit. Two tuning parameter matter: (i) the *exploration batch size* $B$ determining how many arms we try each year besides the current best (i.e., alternative $\lambda$ values to test), and (ii) the *memory decay factor* $\rho$ used when updating past rewards.

**Tuning $B$.** Figure 8 shows a clear hump–shape in accuracy as we vary the number of exploratory arms evaluated each year. Pure exploitation ($B = 0$) already reaches a respectable $R^2$ of about 0.52%, but performance peaks around $B = 5$ at 0.62%. Interestingly batches below 5 dilute the informational gain per extra fit and the curve drifts back down (e.g., $B = 3$ delivers only 0.28%). The time cost moves in the opposite direction: because only the $B$ sampled penalties are re–estimated each round, runtime grows roughly linearly in $B$—from roughly 130 seconds at $B = 0$ to about 450 seconds at $B = 6$.

**Sensitivity to forgetting.** Sweeping the decay parameter confirms the regime asymmetry seen before (Figure 9): before 2000, $R^2$ rises with weaker forgetting (peak $\approx 0.96\%$ when old data are kept); after 2000, performance hovers around zero and turns negative once past rewards dominate.

**Regime differences.** At $B^\star = 5$ the pre-2000 mean $R^2$ is 0.96%, the post-2000 mean is −0.09%. The histograms in Figure 10 echo the expert-learner pattern: a tight right shift before 2000 and a leftward drift afterwards, though the bandit retains a few positive years even in the later period. Notably, the algorithm chooses at the maximum grid value ($\lambda = 10$) during 2009–2019, signaling that the return signal is weak and heavy shrinkage is preferred.

**Penalty path.** The penalty path reinforces this regime split. In the early years, $\lambda_t$ drifts down towards moderate values ($\approx$ 0.6–2) as the GP learns that mild shrinkage pays off. Post-2008,

the bandit repeatedly jumps to the top of the grid ($\lambda = 10$), signaling that the reward surface has flattened and the model must heavily regularize to avoid chasing noise. Despite the weaker statistical fit, the cumulative-return curve still trends upward after 2000, indicating that modest, noisy signals can translate into economic gains once transformed into portfolio weights (Figure 10). In short, the bandit adapts fast enough to keep the strategy profitable, but the quality of the statistical signal deteriorates markedly in the later regime.

**Economic value and cost.** With an overall CV $R^2$ of 0.61%, the bandit attains essentially the same predictive precision as the expanding online expert learner (0.67%) while running in $\approx 726$ seconds—far faster than the nested CV benchmark ($\sim 1100$ s) and the expanding expert learner ($\sim 2545$ s), though slower than the rolling expert ($\sim 305$ s) (as bandit approach uses an expanding data setting). Economically, it delivers an annualized return of 3.30% with a Sharpe of 0.85, matching the expanding learner's payoff at a lower computational bill. In short, the GP–bandit strikes a favorable accuracy–speed trade-off: near-top $R^2$ and Sharpe, without the heavy inner-loop cost of full cross-validation or ever-growing refits.

Table 5: Summary statistics—GP Thompson bandit ($B^\star = 5$)

| Metric | Value |
|---|---|
| Mean CV $R^2$ (overall) | 0.61% |
| Mean CV $R^2$ (pre-2000) | 0.96% |
| Mean CV $R^2$ (post-2000) | −0.09% |
| Annualized portfolio return | 3.30% |
| Annualized st. deviation | 3.87% |
| Annualized Sharpe ratio | 0.852 |
| Execution time | 725.9 sec |

### 6.4 Meta-Learner

A key observation from the results is that the predictive effectiveness of the meta-learner improves significantly only after a sufficient number of historical tasks are accumulated (empirical evidence suggests this is around 30-40 tasks, which is consistent with the behavior of our model). This aligns with the general principle in meta-learning: the learner requires enough examples of past environments and their associated optimal hyperparameters to generalize reliably to new ones.

This translates into longer training window needed compared with other models. Indeed, in order to make the Meta-Learner work we needed to pre-train it using the data from 1972 to 1989, meanwhile the other models are flexible in the choice of the training window length.

**Penalty dynamics.** The penalty parameter $\lambda$ remains low from 1989 to around 2010, suggesting the model found strong signals in lagged returns. After 2010, $\lambda$ rises sharply and remains elevated—indicating reduced signal strength and a shift toward stronger regularization. This coincides with the model reaching maturity (i.e., having enough tasks for robust training), enabling

it to better adapt to evolving market conditions, such as increased noise and competition from systematic strategies. Furthermore, this increase in $\lambda$ is not only a function of model maturity, but also likely reflects structural changes in financial markets. As we approach the mid-2000s and beyond, the meta-learner detects a decline in exploitable signal strength, adapting its penalty upward in response. This aligns with a well-documented evolution in the microstructure of the market: the proliferation of systematic hedge funds, high-frequency trading, and increased competition in quantitative strategies has compressed the predictability of returns. The model effectively interprets this shift by selecting stronger regularization to dampen noise and prevent overreaction to increasingly random or transient patterns.

**$R^2$ distribution.** When analyzing the behavior of the $R^2$ we notice that it is relatively similar to that of previous models: pre-2000 values cluster on the positive side (mean $\approx 0.92\%$) with a few large spikes, whereas the post-2000 distribution widens and shifts left (mean $\approx -0.79\%$) and includes several strongly negative years (Figure 11).

**Economic value and cost.** The meta-learner delivers economic performance broadly in line with the other models. The price is *data hunger*: the model must be pre–trained on a sizable history (here 1972–1989) to accumulate enough "tasks" for the meta-level to generalize. With too few tasks the mapping from regimes to optimal $\lambda$ is poorly identified and the forecasts become erratic, making the approach unsuitable for short samples or newly listed assets. In short, once trained enough the meta-learner is competitive, but it cannot be deployed straight away—it requires extra pretraining time and a long initial window to stabilize its decisions.

However, once the number of tasks increases; typically after 30 to 40, the meta- learner becomes more robust and begins to generalize better. We observe a shift in its recommended $\lambda$ values: it selects higher regularization levels, which act to smooth the signal, reduce overfitting, and emphasize only the most persistent return patterns. Hence, although the $R^2$ metrics decrease slightly during this phase, the economic utility of the model improves, as evidenced by growth of cumulative returns.

Table 6: Rolling Meta-Learner Performance

| Metric | Value |
|---|---|
| Mean CV $R^2$ (overall) | 0.76% |
| Mean CV $R^2$ (pre-2000) | 0.92% |
| Mean CV $R^2$ (post-2000) | −0.79% |
| Annual return | 2.63% |
| Annual volatility | 3.44% |
| Sharpe ratio | 0.77 |
| Execution time | 4 164 .3 sec |

# 7 Conclusion

Table 7: Performance comparison across tuning schemes

| Method | CV $R^2$ (%) | Portfolio | | | Time (sec) |
|---|---|---|---|---|---|
| | | Return | St. dev. | Sharpe | |
| Nested walk–forward CV | −0.18 | 1.98 | 4.77 | 0.416 | 1 099 |
| Rolling expert ($\rho^\star = 0.95$) | 0.23 | 1.90 | 4.77 | 0.399 | 305 |
| Expanding expert ($\rho^\star = 0.85$) | 0.67 | 3.25 | 3.88 | 0.837 | 2 545 |
| GP–Thompson bandit ($B^\star = 5$, $\rho^\star = 1$) | 0.61 | 3.30 | 3.87 | 0.852 | 726 |
| Rolling meta-learner ($Y^\star = 10$) | 0.76 | 2.63 | 3.44 | 0.770 | 4 164 |

This thesis set out to examine whether allowing the degree of regularization in cross-sectional return prediction models to adapt over time leads to improved performance; both statistically and economically. We aimed to test not only whether adaptive tuning of the Ridge penalty yields better out-of-sample predictions, but also how different strategies for achieving this adaptation compare. To this end, we evaluated a progression of methods: Nested Walk-Forward Cross-Validation, Follow-the-Leader Online Expert Learner, Gaussian Process Thompson Sampling Bandit, and a Meta-Learner.

Among methods evaluated, the nested walk-forward CV baseline exhibits the lowest predictive performance, with a negative out-of-sample $R^2$ of $-0.18\%$. Despite this, its portfolio delivers an annualized return of 1.98%, slightly outperforming the rolling-window expert, which achieves only 1.90% but comming far short of other methods. This underperformance is likely driven by the limited number of observations available at each step, which makes the $\lambda$-selection process noisy and prone to overfitting. The nested CV struggles to extract stable signals from these small samples, resulting in weak generalization and inconsistent portfolio returns.

The rolling-window expert strategy improves stability by averaging $\lambda$-scores over time, but remains limited by its myopic focus on recent performance, yielding modest gains in both predictability ($R^2 = 0.23\%$) and economic performance ($Sharpe = 0.399$).

The expanding-window expert further improves by integrating longer-term performance memory, achieving higher predictability ($R^2 = 0.67\%$) and a strong Sharpe ratio (0.837), though at the cost of much higher computational cost as the model took 2545 seconds compared to the 1099 seconds and 305 seconds for the nested CV and rolling-window expert respectively.

The GP bandit exhibits a slightly higher Sharpe (0.852) with slightly lower $R^2$ ($R^2 = 0.61$), balancing exploration (i.e. trying less-tested $\lambda$ values to gather information) and exploitation (using $\lambda$ values that have performed well so far) via a Bayesian updating scheme that delivers robust and consistent returns (3.30%) at moderate computational cost (726 seconds).

Finally, the meta-learner achieves the highest overall predictability ($R^2 = 0.76\%$) and robust economic performance ($Sharpe = 0.770$), despite lagging slightly in raw returns behind the GP bandit (3.30%) and expanding-window expert (3.25%). This highlights the value of learning $\lambda$ values directly from historical regimes, though the method is computationally very expensive (4164 seconds) and requires more training data compared to others.

Overall, although such strategies as expanding-window experts and meta-learners are competi-

tive, the GP bandit approach offers the best tradeoff between predictability, speed, and profitability. This is confirming the benefit of balancing exploration and exploitation in adaptive regularization of Ridge regression.

Moreover the key findings suggest that dynamic tuning of regularization parameters is essential in the presence of structural changes, such as the weakening signal in commonly used predictors like past returns, particularly after the year 2000. Adaptive approaches, were able to adjust regularization strength in response to shifting conditions, delivering significant economic outcomes despite sometimes low statistical $R^2$ values. However, these gains often came at the cost of increased computational complexity or the need for large training windows. These results carry important practical implications. In modern financial environments characterized by low signal-to-noise ratios and competitive erosion of alpha, robust model selection and continuous tuning are essential.

Several limitations should be acknowledged. First, our evaluation hinges only on $R^2$, a statistical measure that may not fully capture economic relevance. Second, the modeling scope is limited to linear and moderately adaptive models, omitting more complex approaches such as deep learning. Third, the empirical setting is confined to a single asset class and region, which may somewhat limit the generalizability of our results.

Future research could extend this work along multiple dimensions. Incorporating recurrent neural networks such as LSTMs may allow for richer modeling of temporal dependencies. Optimizing models directly for financial performance metrics, such as the Sharpe, could offer a more economically meaningful or tangible objective. Exploring dynamic ensembling strategies may further enhance adaptability, while live testing in paper trading environments could help validate practical relevance. Furthermore, one could apply the same principle of time varying hyperparameters to other machine learning methods on different hyperparameters than the regularization parameter in the Ridge regression. It would be undoubtedly interesting to study the behavior of more than one hyperparameter, but likely much more computationally intensive. In sum, this thesis highlights the promise and challenges of adaptive regularization in asset pricing, and points toward fruitful directions for extending its insights.

## References


Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3), 235–256.

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1), 48–77.

Bossaerts, P., & Hillion, P. (1999). Implementing mean–variance portfolio selection with estimation risk: A comparison of various methods. *Journal of Finance*, 54(3), 953–978.

Campbell, J. Y., & Shiller, R. J. (1988). Stock prices, earnings, and expected dividends. *Journal of Finance*, 43(3), 661–676.

Cesa–Bianchi, N., & Lugosi, G. (2006). *Prediction, Learning, and Games*. Cambridge University Press, Cambridge.

Chowdhury, S. R., & Gopalan, A. (2017). On kernelized multi-armed bandits. In *Proceedings of AISTATS* (pp. 901–909).

Center for Research in Security Prices (CRSP). (2023). *CRSP Monthly Stock Database*. Wharton Research Data Services, University of Chicago Booth School of Business.

de Rooij, S., van Erven, T., Grünwald, P. D., & Koolen, W. M. (2014). Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research*, 15, 1281–1316.

Fama, E. F., & Schwert, G. W. (1977). Asset returns and inflation. *Journal of Financial Economics*, 5(2), 115–146.

Giroud, X., & Mueller, H. M. (2019). Testing for episodic predictability in asset returns. *Journal of Econometrics*, 208(1), 75–88.

Goulet Coulombe, P. (2023). A ridge perspective on time-varying parameter models. *Journal of Econometrics*, 244(1), 53–80.

Goyal, A., & Welch, I. (2008). A comprehensive look at the empirical performance of equity premium predictions. *Review of Financial Studies*, 21(4), 1455–1508.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies*, 33(5), 2223–2273.

Herbster, M., & Warmuth, M. K. (1998). Tracking the best expert. *Machine Learning*, 32(2), 151–178.

Hyndman, R. J., Athanasopoulos, G., et al. (2008). Exponential smoothing: State space models under $L_1$ and $L_2$ loss. *International Journal of Forecasting*, 24(1), 2–10.

Kapetanios, G., & Zikes, F. (2018). Time-varying Lasso for large linear models. *Journal of Econometrics*, 206(2), 395–417.

Kelly, B., Pruitt, S., & Su, Y. (2019). Characteristics are predictors: A deep learning perspective. *Journal of Financial Economics*, 135(3), 820–838.

Koo, B., Park, J., & Yoon, S. (2020). Rolling Lasso for time-varying parameter selection. *Econometric Reviews*, 39(5), 523–545.

Lindley, D. V., & Smith, A. F. M. (1972). Bayes estimates for the linear model. *Journal of the Royal Statistical Society: Series B*, 34(1), 1–42.

Littlestone, N., & Warmuth, M. K. (1994). The weighted majority algorithm. *Information and Computation*, 108(2), 212–261.

Merton, R. C. (1973). An intertemporal capital asset pricing model. *Econometrica*, 41(5), 867–887.

Monti, M., Anagnostopoulos, C., & Montana, G. (2018). Adaptive regularization for Lasso in nonstationary environments. *arXiv preprint arXiv:1803.01234*.

Nagel, S. (2021). *Machine Learning in Asset Pricing*. Book manuscript.

Nagel, S., & Xu, Z. (2019). Asset pricing with fading memory. *Journal of Financial Economics*, 132(2), 271–290.

Paye, B. S., & Timmermann, A. (2006). Instabilities in the predictive relationships between stock returns and financial variables. *Journal of Empirical Finance*, 13(3), 274–315.

Pesaran, M. H., & Timmermann, A. (2007). Selection of estimation window in forecasting regressions. *Journal of Econometrics*, 137(2), 587–632.

Remlinger, C., Alasseur, C., Brière, M., & Mikael, J. (2021). Expert aggregation for financial forecasting. *arXiv preprint arXiv:2111.15365*.

Routledge, B. R. (2019). Machine learning for asset allocation. *Financial Analysts Journal*, 75(3), 22–31.

Russo, D., & Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4), 1221–1243.

Russo, D., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2018). A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 11(1), 1–96.

Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret algorithms and experimental design. In *Proceedings of ICML* (pp. 1015–1022).

Stock, J. H., & Watson, M. W. (1996). Evidence on structural instability in macroeconomic time series relations. *Journal of Business & Economic Statistics*, 14(1), 11–30.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4), 285–294.

Wintenberger, O. (2017). Optimal weights for sequential prediction of a bounded sequence. *Bernoulli*, 23(2), 1202–1229.

Yousuf, S., & Ng, W. (2019). Boosting with locally time-varying parameters. *Journal of Financial Econometrics*, 17(1), 158–166.

# Appendix

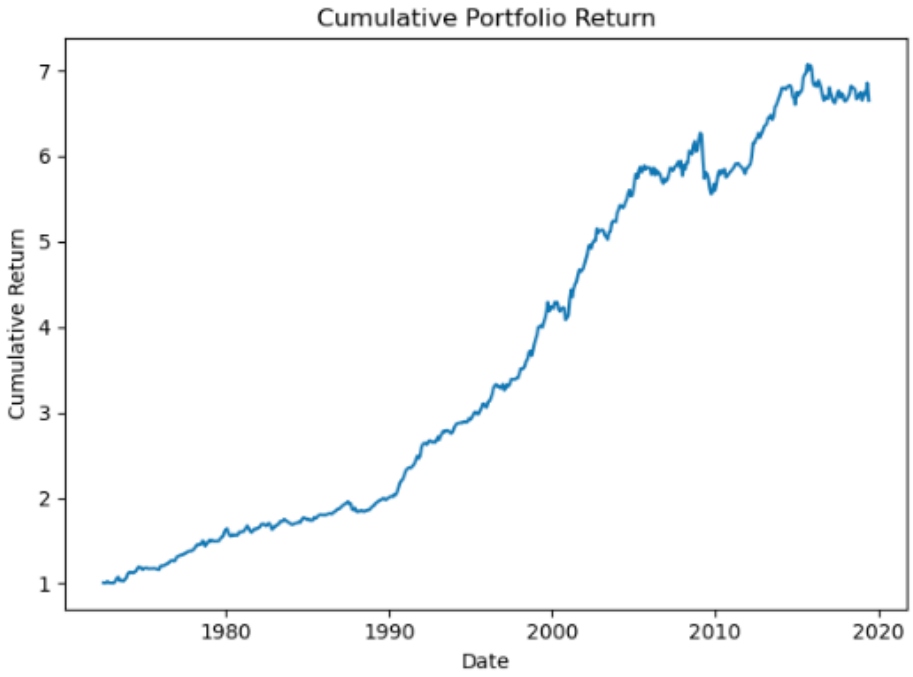


Figure 1: Cumulative Performance of Ridge Regression strategy with LOYO Cross-Validation.

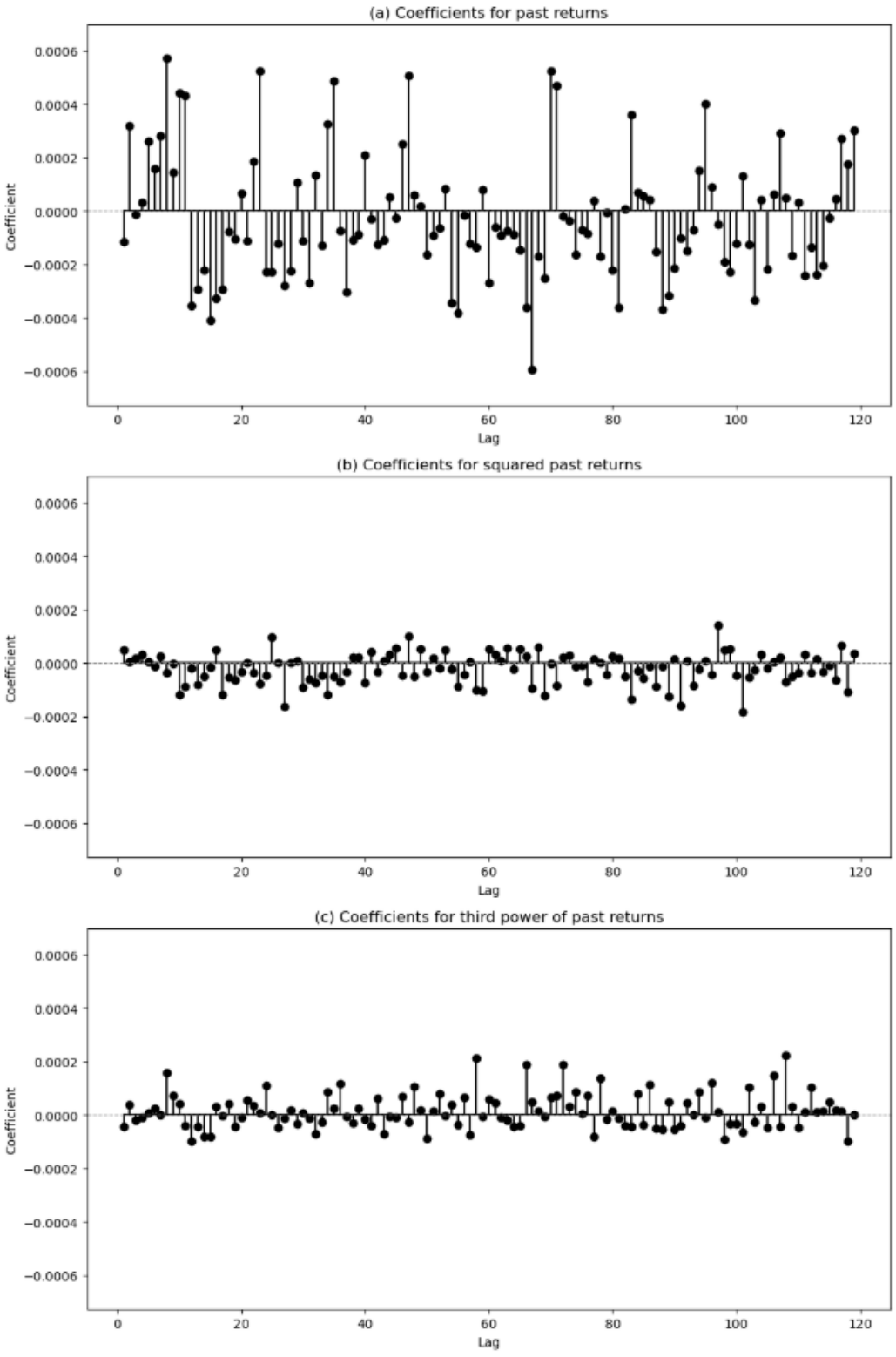


Figure 2: Coefficients for past returns. Each dot corresponds to one lag ($k = 2, \ldots, 120$); stems indicate the sign and magnitude of the estimated Ridge coefficients.

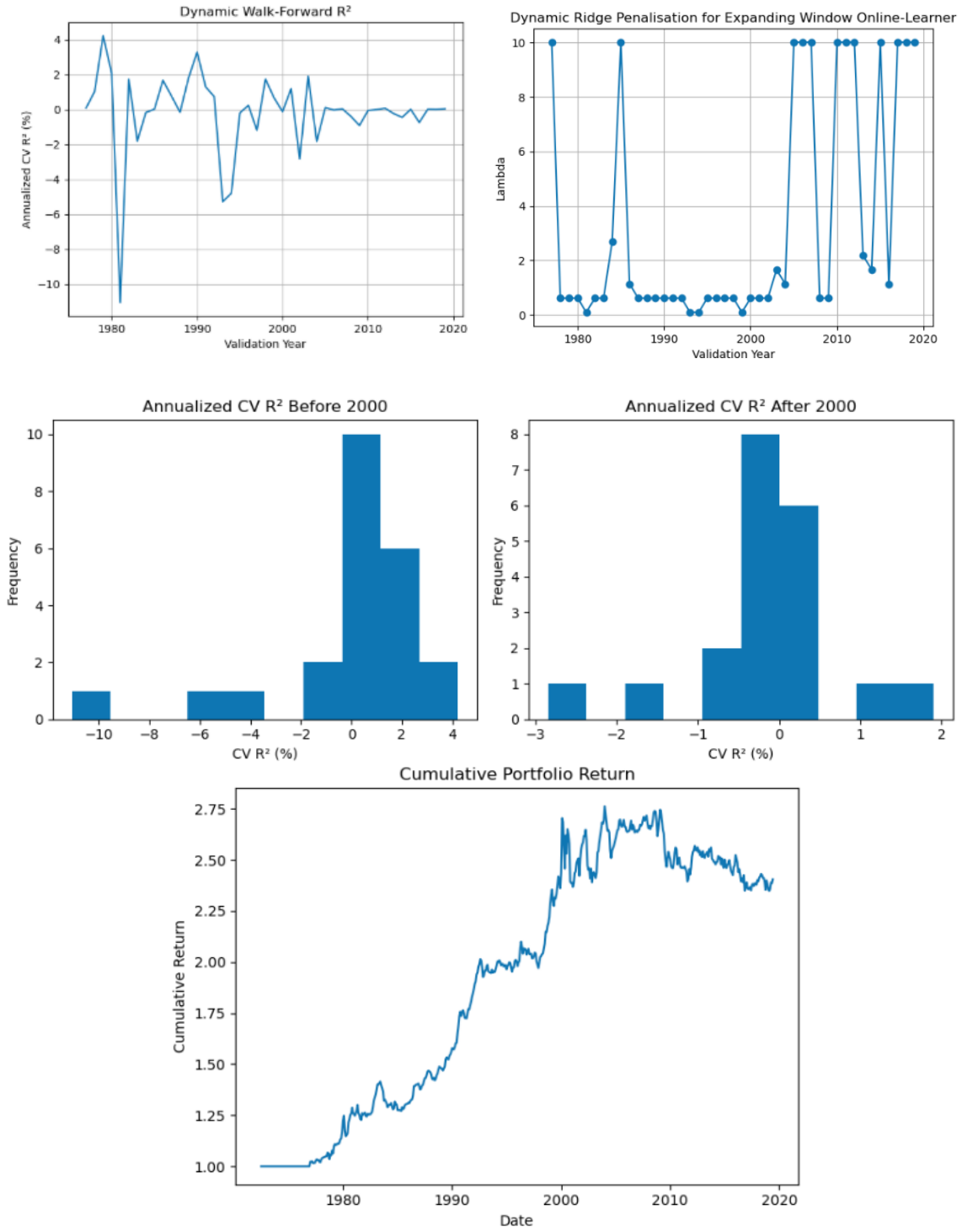


Figure 3: Diagnostics for the nested walk-forward CV experiment. **Top left**: annualized out-of-sample $R^2$ by validation year. **Top right**: Ridge penalty $\lambda$ selected each year. **Middle**: distribution of yearly $R^2$ before (left panel) and after 2000 (right panel). **Bottom**: Cumulative Performance of Nested Walk-Forward strategy with LOYO Cross-Validation.

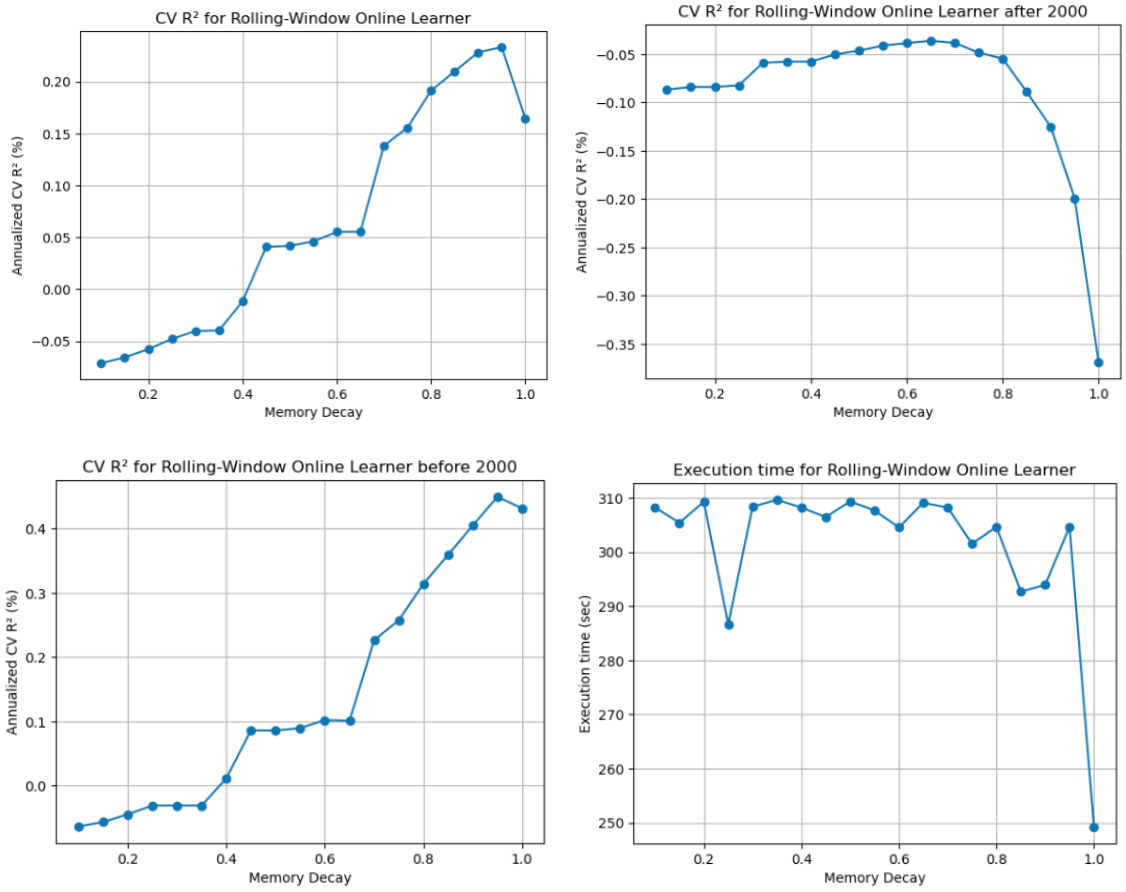


Figure 4: Rolling-window expert learner: effect of memory decay $\rho$. **Top-left**: Overall annualized CV $R^2$. **Top-right**: Post-2000 $R^2$. **Bottom-left**: Pre-2000 $R^2$. **Bottom-right**: Execution time. The optimal value $\rho^\star = 0.95$ maximizes overall $R^2$ at 0.23%.

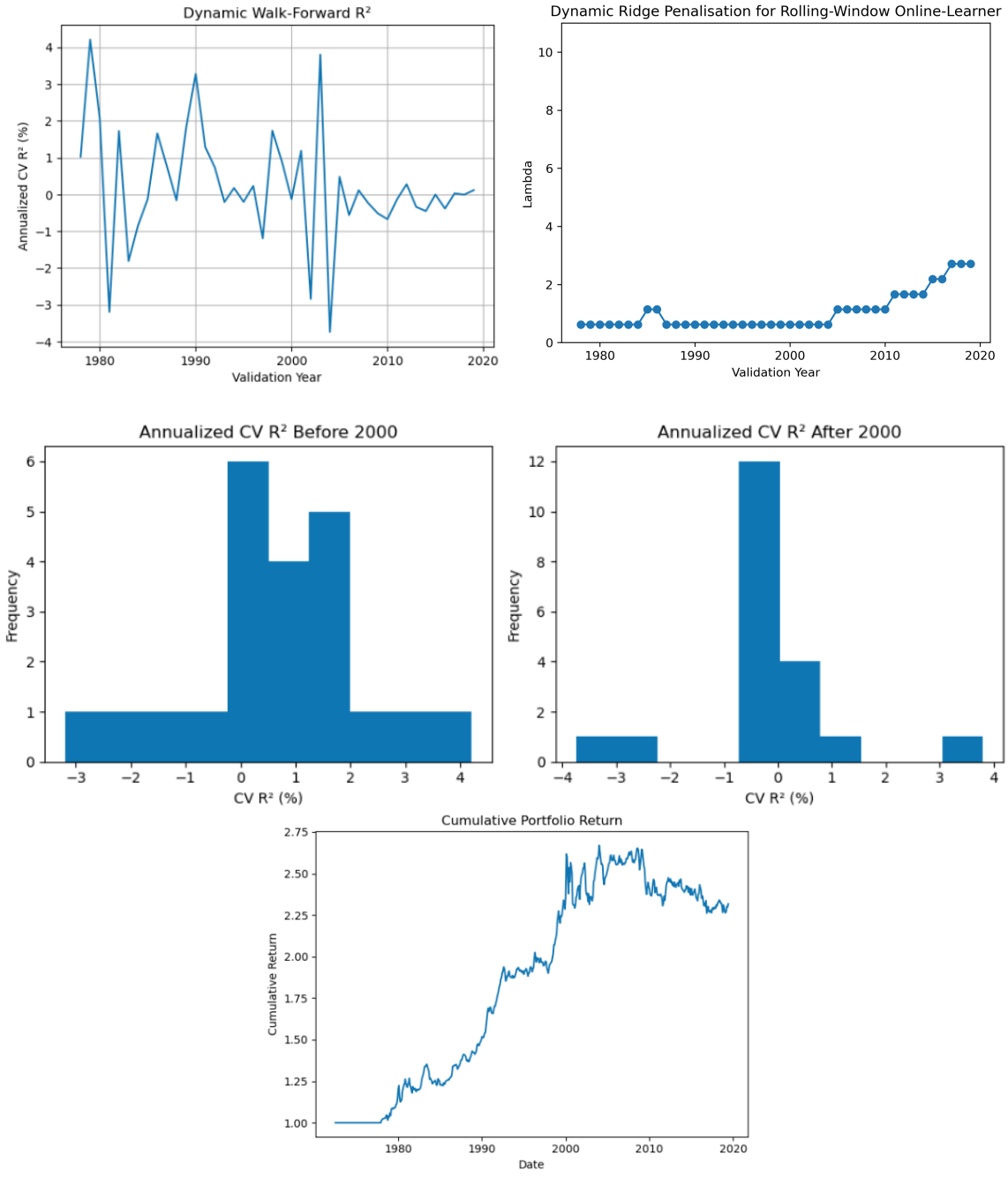


Figure 5: Rolling–window expert learner at the optimal memory decay $\rho^\star = 0.95$. **Top-left**: annualized out-of-sample $R^2$ by validation year. **Top-right**: Ridge penalty $\lambda$ selected each year. **Middle**: Distribution of yearly $R^2$ before (left panel) and after 2000 (right panel). **Bottom**: Cumulative Performance of Rolling Window Online Expert Learner strategy.

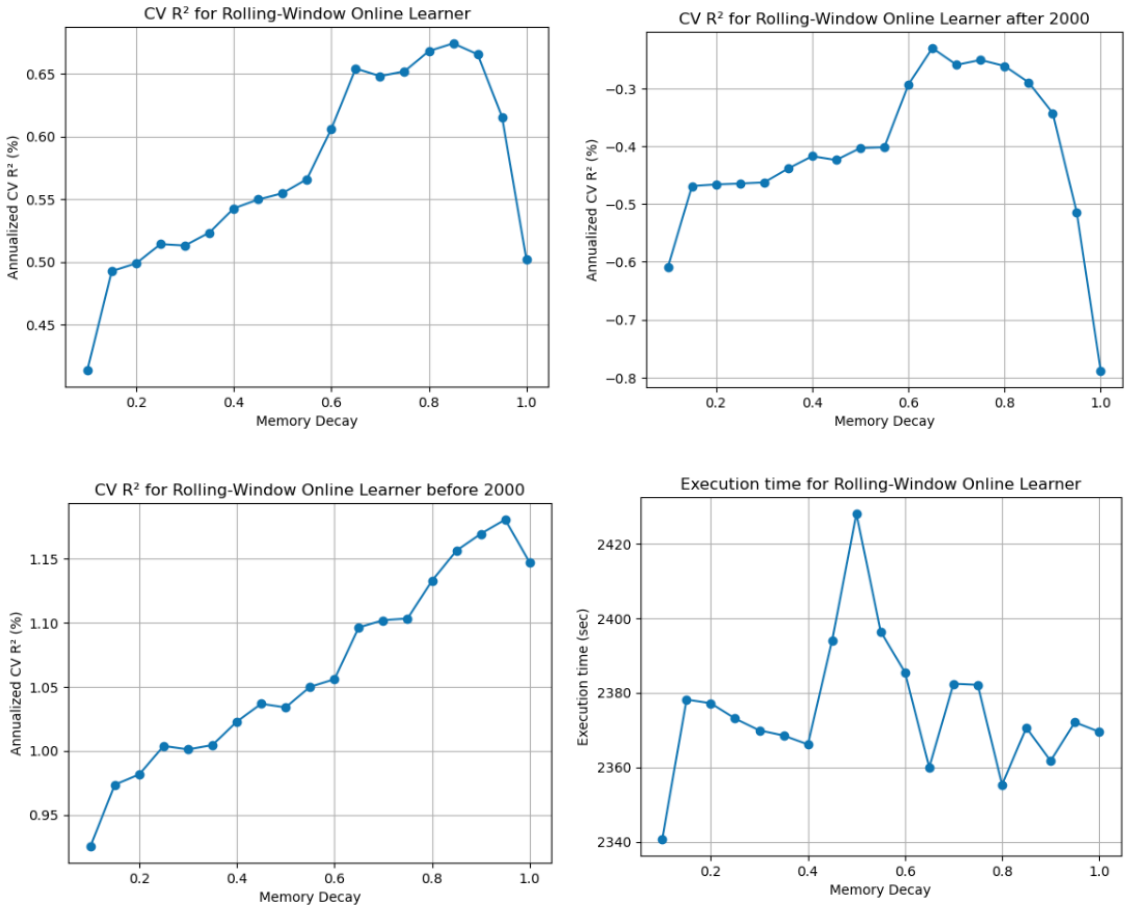


Figure 6: Expanding-window expert learner: effect of memory decay $\rho$. **Top-left**: overall annualized CV $R^2$. **Top-right**: Post-2000 $R^2$. **Bottom-left**: Pre-2000 $R^2$. **Bottom-right**: Execution time. The optimal value $\rho^\star = 0.85$ maximizes overall $R^2$ at 0.67%.

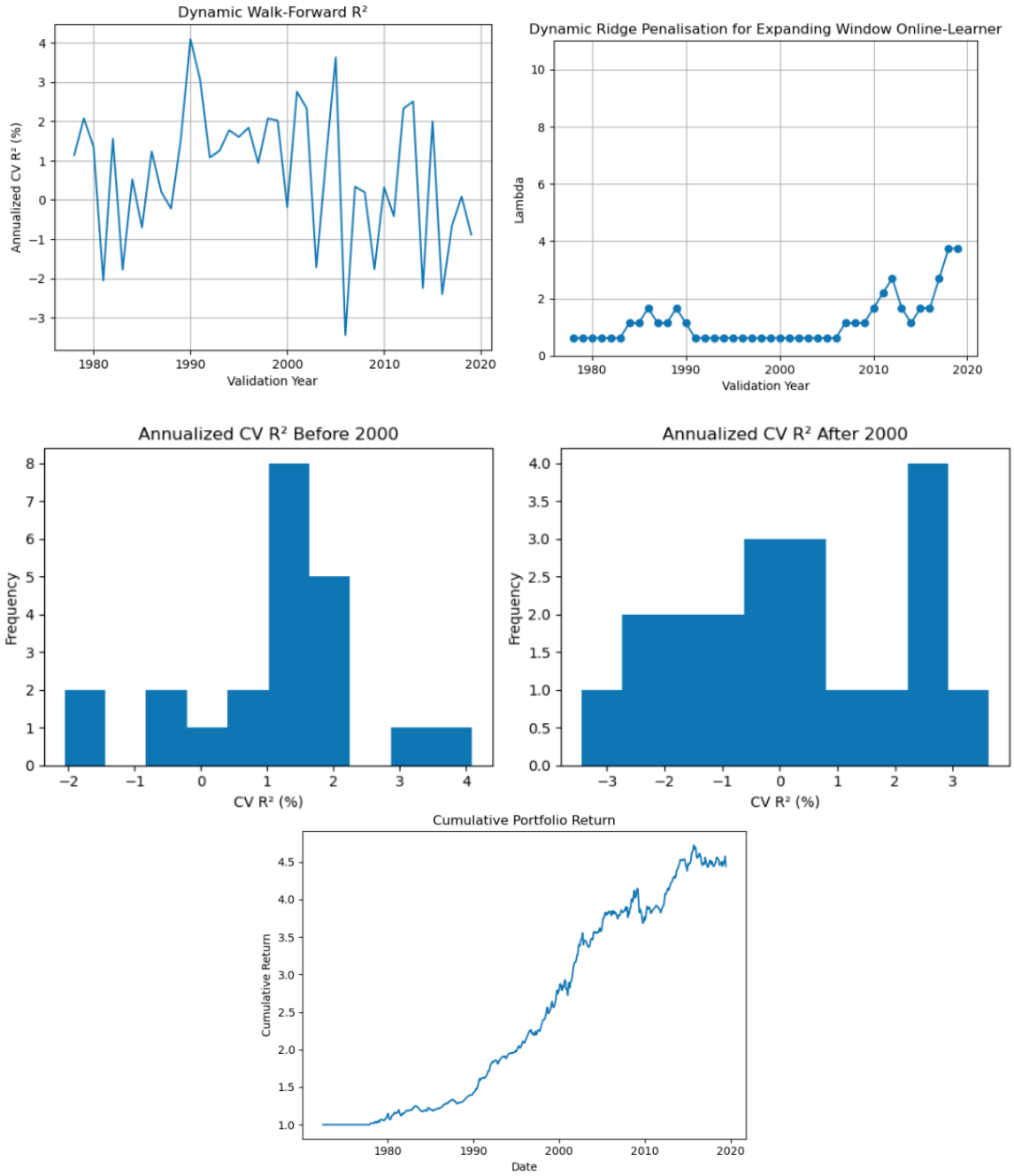


Figure 7: Expanding-window learner at $\rho^\star = 0.85$. **Top-left**: annualized out-of-sample $R^2$ by validation year. **Top-right**: Ridge penalty $\lambda$ selected each year. **Middle**: Distribution of yearly $R^2$ before (left panel) and after 2000 (right panel). **Bottom**: Cumulative Performance of Expanding Window Online Expert Learner strategy.

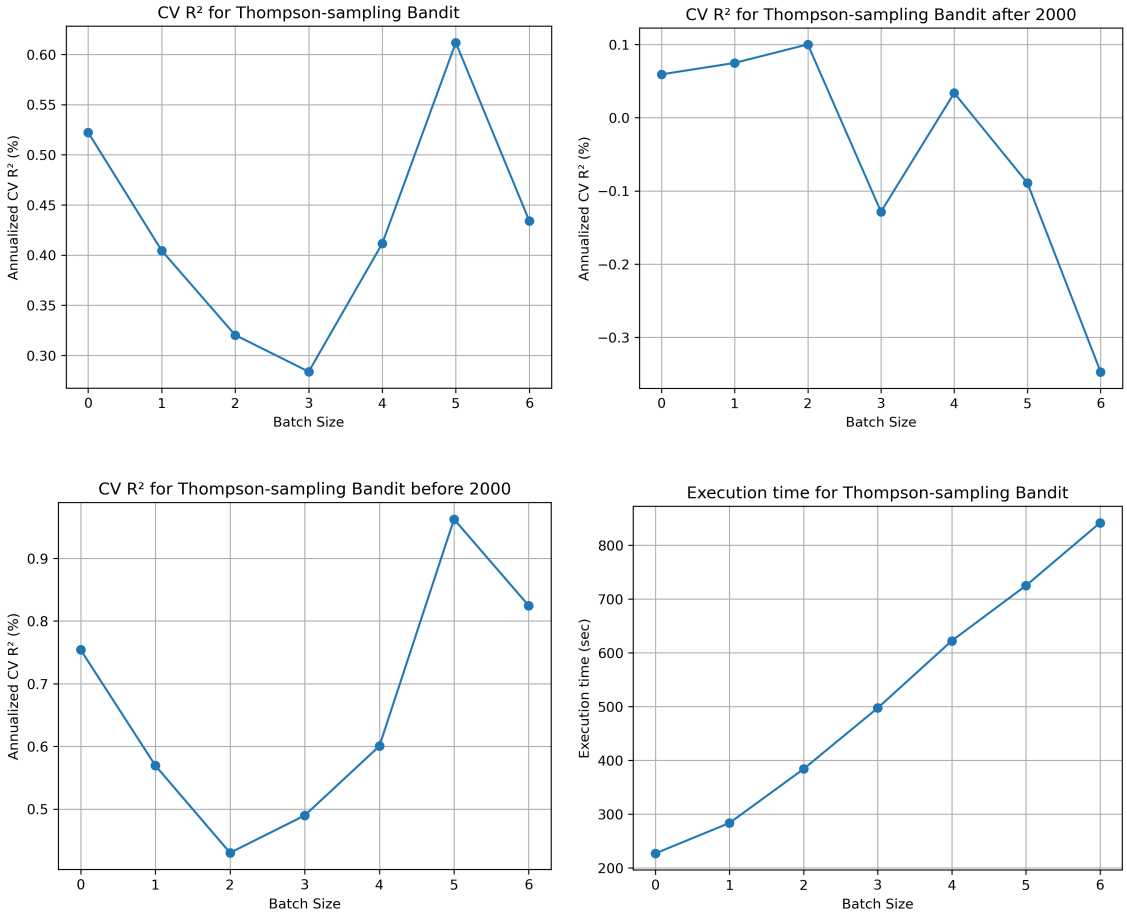


Figure 8: Gaussian Process Bandit Learner: effect of batch size $B$. **Top-left**: overall annualized CV $R^2$. **Top-right**: Post-2000 $R^2$. **Bottom-left**: Pre-2000 $R^2$. **Bottom-right**: Execution time. CV $R^2$ peaks at $B^\star = 5$.

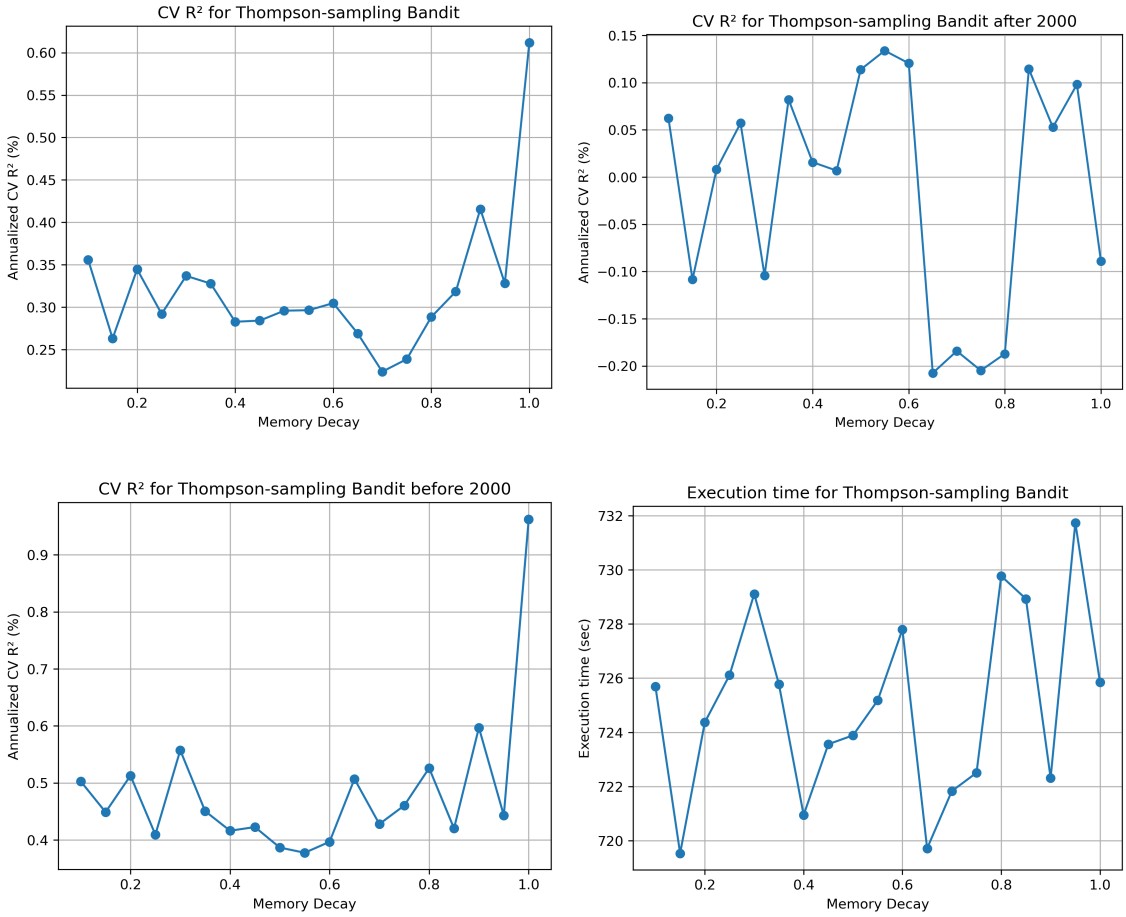


Figure 9: Gaussian Process Bandit Learner: effect of memory decay $\rho$. **Top-left**: overall annualized CV $R^2$. **Top-right**: Post-2000 $R^2$. **Bottom-left**: Pre-2000 $R^2$. **Bottom-right**: Execution time. CV $R^2$ peaks at $\rho^\star = 1$.

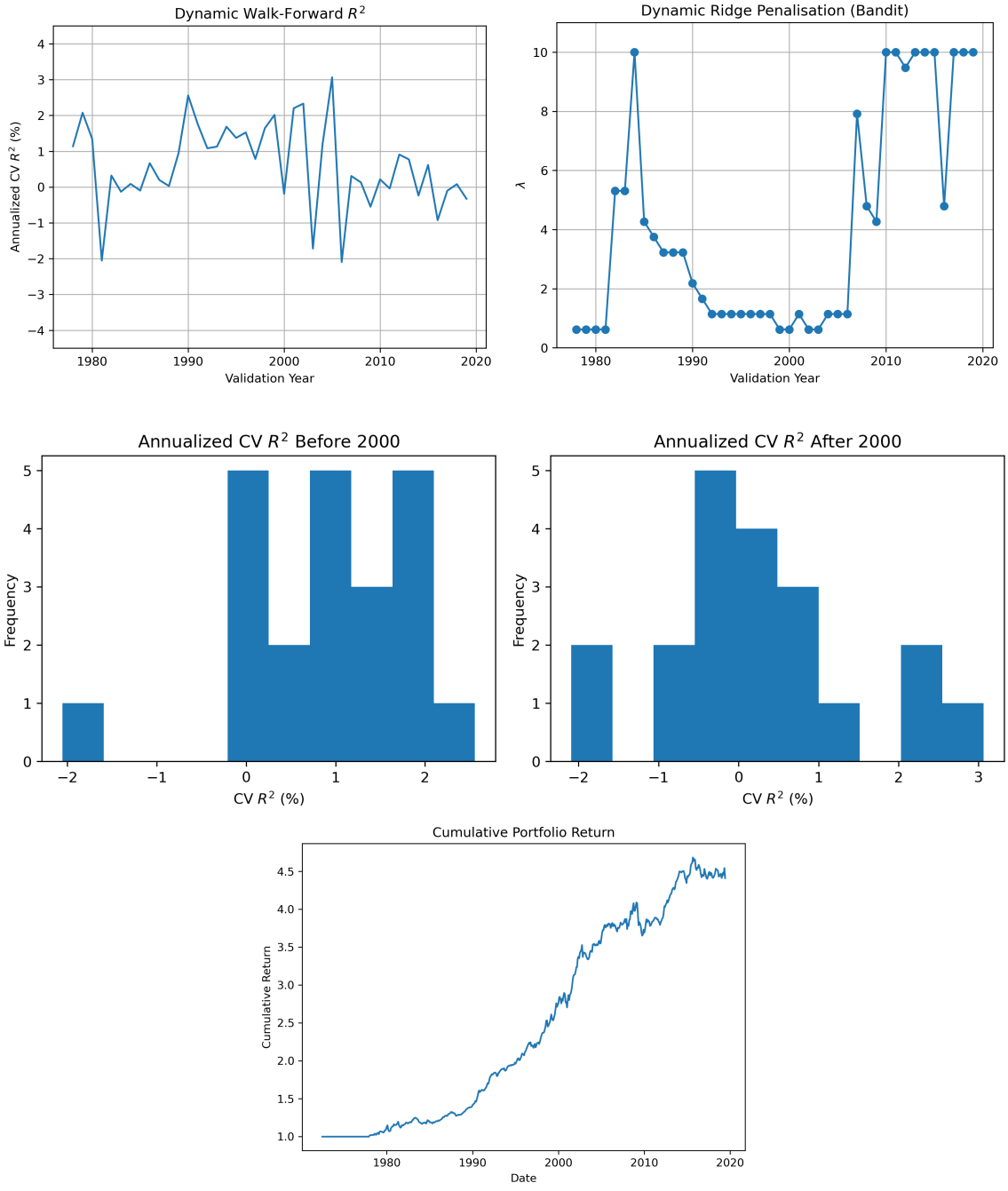


Figure 10: GP–Thompson bandit at $B^\star = 5$ and $\rho^\star = 1$. **Top-left**: annualized out-of-sample $R^2$ by year. **Top-right**: selected Ridge penalty $\lambda_t$. **Middle**: yearly $R^2$ histograms before (left) and after 2000 (right). **Bottom**: cumulative portfolio return.

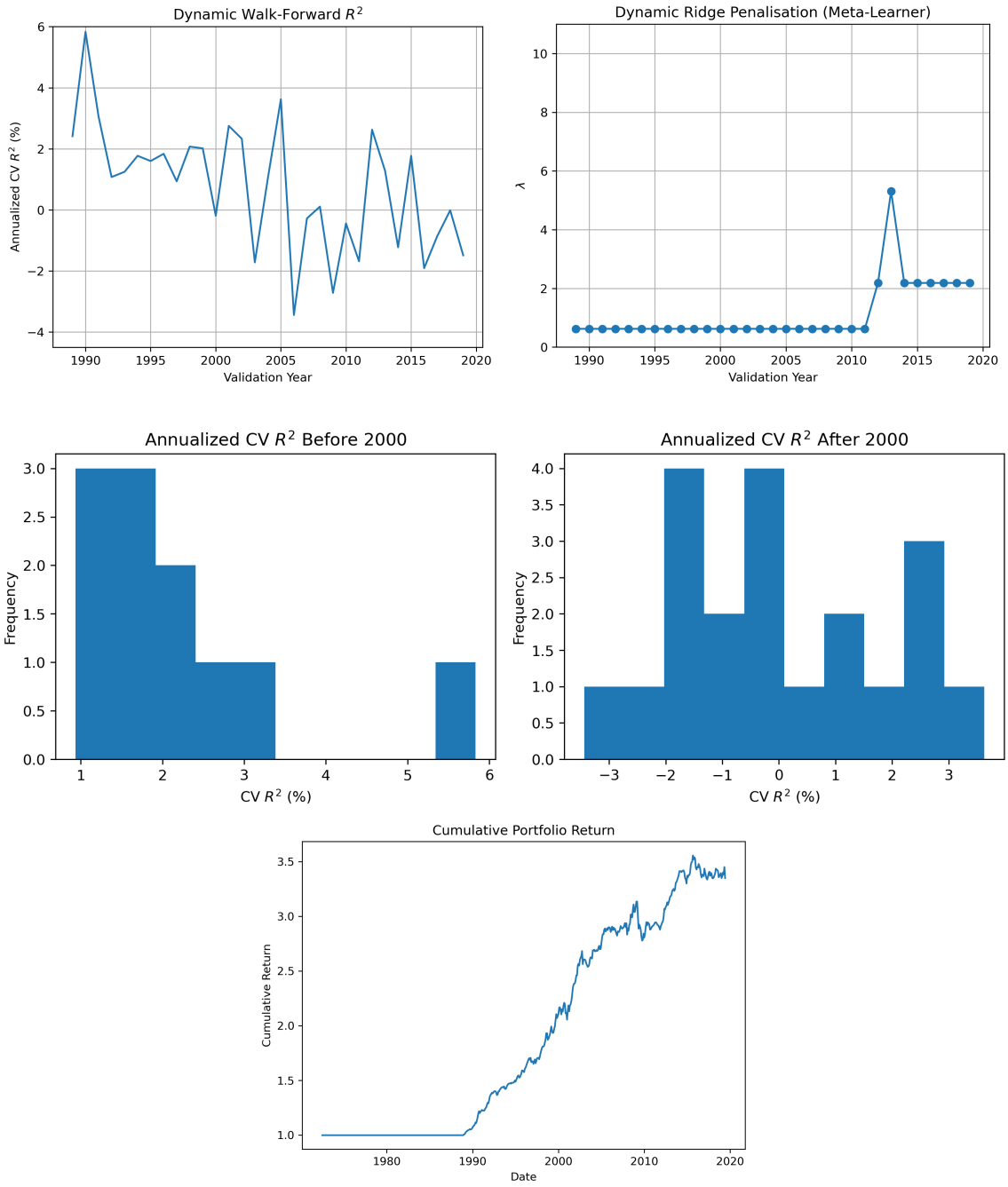


Figure 11: Diagnostics for the rolling Meta-Learner experiment. **Top left**: annualized out-of-sample $R^2$ by validation year. **Top right**: Ridge penalty $\lambda$ selected each year. **Middle**: distribution of yearly $R^2$ before (left panel) and after 2000 (right panel). **Bottom**: Cumulative Performance of rolling Meta-Learner strategy.