Chaire en macroéconomie et prévisions

Maximally Machine-Learnable Portfolios

Philippe Goulet Coulombe[1] & Maximilian Göbel[2]

Cahier de recherche
Working paper
2023-01

Avril / April 2023

ESG UQÀM
École des sciences de la gestion
Université du Québec à Montréal

[1] Université du Québec à Montréal
[2] Bocconi University

Chaire en macroéconomie et prévisions

La Chaire en macroéconomie et prévisions est fière de s’appuyer sur un partenariat avec les organisations suivantes:

Finances Québec   Hydro Québec   Emploi et Solidarité sociale Québec   CDPQ

Les opinions et analyses contenues dans les cahiers de recherche de la Chaire ne peuvent en aucun cas être attribuées aux partenaires ni à la Chaire elle-même et elles n’engagent que leurs auteurs.

Opinions and analyses contained in the Chair’s working papers cannot be attributed to the Chair or its partners and are the sole responsibility of the authors.

Maximally Machine-Learnable Portfolios

Philippe Goulet Coulombe[*]
Université du Québec à Montréal

Maximilian Göbel
Bocconi University

First Draft: December 5, 2022
This Draft: April 24, 2023

Abstract

When it comes to stock returns, *any form* of predictability can bolster risk-adjusted profitability. We develop a collaborative machine learning algorithm that optimizes portfolio weights so that the resulting synthetic security is maximally predictable. Precisely, we introduce MACE, a multivariate extension of Alternating Conditional Expectations that achieves the aforementioned goal by wielding a Random Forest on one side of the equation, and a constrained Ridge Regression on the other. There are two key improvements with respect to Lo and MacKinlay's original maximally predictable portfolio approach. First, it accommodates for any (nonlinear) forecasting algorithm and predictor set. Second, it handles large portfolios. We conduct exercises at the daily and monthly frequency and report significant increases in predictability and profitability using very little conditioning information. Interestingly, predictability is found in bad as well as good times, and MACE successfully navigates the debacle of 2022.

[*] Departement des Sciences Économiques, goulet_coulombe.philippe@uqam.ca. For helpful discussions and comments, we would like to thank Frank Diebold, Dave Rapach, Erik Christian Montes Schütte, Hugo Subtil, Dalibor Stevanovic, and Boyuan Zhang as well as participants at the UQAM seminar and CFE London 2022. For research assistance, we are grateful to Mikael Frenette and Félix-Antoine Gaudreault.

1 Introduction

A natural trading strategy is to buy (sell) assets whose price one expects to appreciate (depreciate) with *higher certainty*. That is, out of a basket of securities, it may be preferable to focus active trading efforts on the most predictable assets given one's information set. It is well known that marginal predictive accuracy improvements can translate into substantial profits without equally substantial risks for investors. But such desirable assets are in very short supply, if they can be identified at all. Consequently, a natural question is whether we can reach to the ideal by constructing more predictable synthetic securities as linear combinations of (mostly) unpredictable existing ones. This paper devises a data mining technique that drills out those – Maximally Machine-Learnable Portfolios (MMLP) – by directly optimizing portfolio weights so to maximize forecasting accuracy, and thereby risk-adjusted returns.

The origins of such ideas lie within Lo and MacKinlay (1997)'s maximally predictable portfolios (MPP), where a set of weights $\boldsymbol{w}$ are chosen so to maximize the R^2 of $\boldsymbol{w}'\boldsymbol{r}_t$ with a tightly specified linear regression based on a few factors. In this convenient sparse (both in returns and predictors) linear framework, obtaining the MPP reduces to solving an eigenvalue problem (akin to canonical correlation analysis) subject to a constraint.[1] Fundamental limitations are apparent. First, the predictive function is of rather limited sophistication. Namely, it is linear, low-dimensional, and most often unregularized. This limits the obtained MPPs to lie within a narrow space of maximally *linearly*-predictable portfolios. Clearly, additional patterns of predictability are likely to be found when allowing for complex nonlinear relationships while keeping an eye on the characteristically hostile signal-to-noise ratio (SNR) of asset pricing applications. Second, the portfolio side of the equation is similarly unregularized, opening two evenly unpleasant routes. One can either limit the number of assets to be included and severely bound the space of MPP candidates, or run into the well-documented estimation problems of (large) covariance matrices (Ledoit and Wolf, 2004). MMLPs are designed to avoid all of the above by including a powerful nonlinear nonparametric function approximator on the right hand side, and regularization schemes for both the learnable portfolio weights and the corresponding predictive function. Those qualities, flexible nonlinearities and thoughtful regularization, have both been instrumental to the ML renaissance in empirical asset pricing (Gu et al., 2020; Chen et al., 2021a; Nagel, 2021), and will be so again crossing the bridge from MPP to MMLP.

MACE. We introduce MACE, which stands for Multivariate Alternating Conditional Expectations, and is a multifaceted generalization of Breiman and Friedman (1985)'s ACE algorithm. The latter was originally designed for nonlinearly transforming a univariate regression target to maximize association. As the name suggests, ACE achieves its aim by alternating the estimation of two functions (one for the right-

[1] In practice, solving the non-convex fractional programming optimization problem this quickly becomes a daunting task – particularly when considering many assets, factors, and constraints on the portfolio's composition (Gotoh and Fujisawa, 2014). For this reason, many alternative numerical methods have been proposed to solve more successfully the MPP problem (Yamamoto et al., 2007; Konno et al., 2010a,b; Takaya and Konno, 2010; Gotoh and Fujisawa, 2014). Recent applications include Harris et al. (2022) and Ta et al. (2022).

hand side (RHS), and another for the left-hand side (LHS)) taking the other as fixed at each iteration, very much in the spirit of EM algorithms. Adapting ACE to the MMLP problem, MACE modifies it in two key aspects. First, the LHS is multivariate (an extension) and linear (a restriction). Second, it replaces ACE's rudimentary additive polynomial models by a Ridge Regression (RR) on the portfolio side and a Random Forest (RF) on the prediction side. RR provides a linear and regularized fit for the LHS, avoiding overfitting and non-plausible allocations. RF is a powerful off-the-shelf predictive algorithm that (i) handles high-dimensional data, (ii) can approximate a wide range of unspecified nonlinearities, (iii) requires little tuning, (iv) very rarely overfits. Then comes a panoply of algorithmic details that make the cohabitation of aforementioned elements possible: a learning rate, block out-of-bag sampling, decreasingly random optimization, and bagging strategies rather than predictions. Those are all extensively discussed in the paper.

We also examine the link between MACE and traditional mean-variance portfolio optimization. In short, it minimizes the portfolio variance that is orthogonal to the information set. Given the low level of predictability that is characteristic for this application, the unconditional and conditional solutions are not miles apart in terms of resulting variance. This simple observation provides an explanation for the good variance properties of the algorithm as well as previous observations for MPPs (Ta et al., 2022; Harris et al., 2022). It also explains why, as we will see, that taking the MACE portfolio as fixed and not trading it according to RF also delivers competitive results.

Regarding portfolio construction, the majority of the literature relies on a two-step procedure, prediction coming first and portfolio construction, following various fixed rules, coming next. MACE, in contrast, fits into a more recent stream of studies that optimizes portfolio weights directly – explicitly or implicitly encompassing the prediction step. For instance, Cong et al. (2021) do so with a reinforcement learning algorithm targeting Sharpe Ratios fueled with a large database à la Gu et al. (2020), and Firoozye et al. (2022) deploy linearity to rewrite their simultaneous mean-variance/prediction problem with vector autoregressive predictions as canonical regression analysis. MACE's advantages with respect to those alternatives are hereby visible. It is nonlinear and nonparametric, yet remains simple, transparent and rather traditional in its trading decisions, and works with or without terabytes of data. Statistically, it is a kind of semi-parametric canonical correlation analysis (Michaeli et al., 2016) supplied with various desirable features for financial forecasting. Economically, it is a conceptually simple extension of the mean-variance principle.

Statistical Arbitrage at the Daily Frequency. We consider two applications. The first is creating portfolios of the 20, 50, and 100 most capitalized firms on the NASDAQ for trading at the daily frequency. We evaluate those from January 2017 to December 2022, an era for which gains of ML-based statistical arbitrage are expected to be low, if they exist at all (Krauss et al., 2017). The information set is lagged returns of the portfolio itself. Thus, in this setup MACE is looking for maximally *nonlinearly* mean-reverting portfolios. And indeed it does find some, scoring enviable returns and risk-reward ratios. Nonlinearities prove instrumental to such results as the degree of mean-reversion (when approximated linearly) is shown to be highly state-dependent. Out-of-sample R^2 testifies to that, ranging from a moderate 0.5-0.9% in calmer periods, to 12% during the first wave of Covid-19, and a staggering 20% when zooming in on March 2020. MACE is shown to heavily rely on day-to-day oscillations to achieve swift returns in tumultuous months – a behavior learned in part from the financial crisis. Most importantly, it also outperforms benchmarks outside of high volatility episodes, both in bull and bear markets. In particular, all MACEs deliver positive returns in 2022, ranging from 5% to 23%. This and other features lead MACE, sometimes with only 20 highly liquid stocks, to nearly double the market's Sharpe Ratio.

This application expands on various strands of the statistical arbitrage literature, where many tactics (ranging from heuristics to cointegration tests) have been proposed to identify mean-reverting portfolios or pairs with predictable spreads (see Krauss (2017) for an extensive survey). Typically, the candidate securities are fixed ex-ante rather than "discovered", and mean reversion is linear. Nonetheless, when a good discovery is made – like that of Medhat and Schmeling (2022) exploiting very different time series behaviors for low- vs. high-turnover stocks – gains can be huge. Our approach mines for such discoveries. Hence, more closely related are the works of d'Aspremont (2011), Cuturi and d'Aspremont (2013), and Fogarasi and Levendovszky (2013) who also focus on constructing maximally mean reverting portfolios. Linearity is inherent, and allows for such problems to be reformulated as extensions of canonical correlation analysis with a varying degree of elaboration. An important focus of the literature, as it is the case for MPPs, has been on improving computations and placing reasonable constraints (like sparsity) on the allocation (Zhao and Palomar, 2016, 2018; Griveau-Billion and Calderhead, 2021). Nonetheless, linearity remains pervasive, with some directly targeting linear autocorrelation statistics (Zhao and Palomar, 2016), and others embedding directly linear forecasting models, like Vector Autoregressions, within the optimization framework (Griveau-Billion and Calderhead, 2021; Firoozye et al., 2022). Therefore, MACE, through its use of RF, evidently widens the space of exploitable time series dependence for statistical arbitrage. Additionally, the portfolio side of the equation is not constrained computationally nor statistically from including a myriad of stocks and a long daily sample, a case of interest from a conceptual standpoint (testing market efficiency) and eventually practical in an era of shrinking transaction costs.

Monthly Trading Based on Macroeconomic Indicators. The second application is at the monthly frequency and utilizes the canonical Welch and Goyal (2007) data set to construct an MMLP with a subset of large-cap individual stock returns from CRSP. Thus, it uses no firm characteristics and is rather looking for aggregate predictability based on trivially available macroeconomic indicators. All RFs (MACE or not) struggle to deliver positive R^2's from the late 1980s up to the mid-2000s. However, MACE hits an R^2 of above 4% in the last 15 years of our sample ending in 2019 – an era for which predictability and associated economic gains (ML-based or not) are often reported to have waned (Chordia et al., 2014; Han et al., 2018; Gu et al., 2020; Cong et al., 2021). This is achieved in part during the financial crisis and the years thereafter, where MACE limits losses considerably, and catches up with the pre-crisis trend as early as mid-2009. Other non-MACE portfolios using RF as the predictive function also manage to somehow mitigate losses during the meltdown, but then fail to find and leverage predictability when exiting the Great Recession. Moreover, during the slowdown of 2018, only MACE continues with mostly unabated upward-trending returns. We find, using interpretable ML tools, that this success is attributable to MACE uncovering a portfolio with a subtle response to elevated volatility, in the form of a nonlinearly time-varying risk premium.

Our approach thus differs from Gu et al. (2020) and the vast body of (monthly) studies who use a pooled panel approach with nearly 2 million observations of stock returns from roughly 30,000 U.S.-listed companies, with a corresponding feature set of over 100 company characteristics and macro-predictors. This is also the backbone of Cong et al. (2021)'s reinforcement learning approach. A similar pooled panel with many cross-sectional and time series characteristics is also found for exchange rates in Filippou et al. (2022) and for cryptocurrencies in Filippou et al. (2021). Given that MACE does not predict each stock separately, and rather focuses on forecasting a single synthetic index with easily available time series, it can be described, at least in relative terms, as a very low-maintenance strategy. Moreover, by construction, it cannot rely on cross-sectional anomalies that have already dissipated, or focus on illiquid, once non-adequately priced stocks. Obviously, MACE is not exempt either from risking an eventual depletion of its sources of predictability. Nonetheless, those concerns are alleviated by the multi-solution nature of the algorithm and the opaque prediction function. Indeed, MACE's market timing comes from a (mostly) black box prediction function that cannot be easily deduced by other market participants, and most importantly, if a certain linear combination has been overharvested, MACE can dig out others.

Outline. This paper goes as follows. Section 2 introduces MACE, motivates its structure, and discusses practical aspects. Section 3 conducts the daily trading empirical analysis and section 4 conducts a monthly frequency exercise. Section 5 concludes.

2 MACE

The sophistication of MMLPs, particularly the use of nonlinear tree ensembles-based predictions, necessitates the design of a vastly different framework for optimization than what prevailed for MPPs.

2.1 The Algorithm

MACE is, for the most part, a conceptually trivial extension of ACE. Its successful empirical development, however, requires a fair amount of subtle machine learning craftsmanship. Breiman and Friedman

(1985)'s ACE applied to a generic h-step ahead forecasting problem of a single target Y_{t+h} reads as

$$g(\underbrace{Y_{t+h}}_{1\times 1}) = f\,(\underbrace{\mathbf{X}_t}_{1\times K}) \ + \varepsilon_{t+h} \tag{1}$$

where g and f are unknown functions, ε_{t+h} is the prediction error, $\mathbf{X}_t$ is the matrix of K available predictors at time t (which may include lags or various indicators). Thus, the sole deviation from the textbook predictive regression setup is the introduction of g. ACE's goal is to find the optimal transformation of g, in the sense that it is maximally predictable by the output of f. Breiman and Friedman (1985) show that $\hat{g}$ and $\hat{f}$ can be obtained from running an iterative algorithm that alternates between obtaining the conditional expectation of $g(Y_{t+h})$ given $\mathbf{X}_t$ for a fixed g and the conditional expectation of $f(\mathbf{X}_t)$ given Y_{t+h} for a fixed f. Following the original incarnation, g and f typically consist of backfitted polynomial functions, which used to be a popular nonparametric ML approach – before being outshadowed by the advent of tree ensembles in the 1990s and the resurrection of neural networks in the mid-2000s. Nonetheless, the polynomial approach still remains the predictive function in recent ACE applications (Gopinathan and Durai, 2019; Rao et al., 2022).

This paper extends ACE in three ways so that it can uncover a modern brand of maximally predictable portfolios: **(i)** Y_{t+h} is replaced by $\mathbf{Y}_{t+h} \in \mathbb{R}^N$ and $g : \mathbb{R} \to \mathbb{R}$ by $g : \mathbb{R}^N \to \mathbb{R}$, **(ii)** we impose a series of constraints on g so that its output is a portfolio, and **(iii)** f is a high-performing off-the-shelf modern ML tool. All three are vital to the current application, to a varying degree of obviousness. **(i)** puts the M in MACE by making it a multivariate problem and therefore allowing for g's input to be, for instance, a panel of stock returns. From this, (1) becomes the general MACE problem

$$\underbrace{g\,(\underbrace{\mathbf{Y}_{t+h}}_{1\times N})}_{1\times 1} = f\,(\underbrace{\mathbf{X}_t}_{1\times K}) \ + \varepsilon_{t+h} \tag{2}$$

which lies within the broad class of nonparametric canonical correlation problems (Michaeli et al., 2016). This is also contained within the class of models for which Makur et al. (2015) develop theoretical guarantees for generic ACE-type algorithms. Now, **(ii)** restricts g's original nonparametric ambitions to that of learning a linear combination of $\mathbf{Y}_{t+h}$'s (with positive weights summing to 1) so $\hat{g}(\mathbf{Y}_{t+h})$ is a portfolio return series — as opposed to being literally anything, which could nevertheless be of interest in other financial applications. The minimization problem that ensues is

$$\min_{w,f} \sum_{t=1}^{T} (\boldsymbol{w}'\boldsymbol{r}_{t+h} - f(\mathbf{X}_t))^2 + \lambda||\boldsymbol{w}||^2 \quad \text{such that } \ \boldsymbol{w} \geq 0 \ \text{ and } \ \boldsymbol{w}'\boldsymbol{\iota} = 1 \tag{3}$$

where $\mathbf{Y}_{t+h}$ is hereafter also assumed to be a panel of stock returns $\boldsymbol{r}_{t+h}$. The addition of $\lambda||\boldsymbol{w}||^2$ provides l_2 regularization with intensity λ (an hyperparameter) that will guard against overfitting *and* non-

Algorithm 1 *MACE*

```
1: Initiate $\hat{z}_{0,t+h}$ as the scaled equally weighted portfolio, learning rate is $\eta$,
2: for $s = 1$ to $s_{\max}$ do
3:     The Random Forest Step
           $f_s^* = \arg\min_{f \in \mathcal{F}_{\text{RF}}} \sum_{t=1}^{T} (\hat{z}_{s-1,t+h} - f(\mathbf{X}_t))^2$
       where we keep $f_s^*(\mathbf{X}_t)$, a time series of block out-of-bag predictions.
4:     Update the RHS: $\hat{f}_s(\mathbf{X}_t) = \eta \times f_s^*(\mathbf{X}_t) + (1-\eta) \times \hat{f}_{s-1}(\mathbf{X}_t)$
5:     The Ridge Regression Step
           $\boldsymbol{w}_s^* = \arg\min_{\boldsymbol{w}} \sum_{t=1}^{T} (\hat{f}_s(\mathbf{X}_t) - \boldsymbol{w}'\boldsymbol{r}_{t+h})^2 + \lambda||\boldsymbol{w}||^2 \quad \text{st} \;\; \boldsymbol{w} \geq 0$
       where we keep in-sample predictions $z_{s,t+h}^* = \boldsymbol{w}_s^{*\prime}\boldsymbol{r}_{t+h}$.
6:     Update the LHS: $\hat{z}_{s,t+h} = \texttt{scale}(\eta \times z_{s,t+h}^* + (1-\eta) \times \hat{z}_{s-1t+h})$
7: end for   if $\boldsymbol{w}_s \approx \boldsymbol{w}_{s-1}$, or if some early stopping criterion has been met.
```

realistic allocations (Carrasco and Noumon, 2011). The non-negativity constraint may or may not be activated. For instance, it will be turned off in our daily application. Its activation implies additional shrinkage beyond that of the l_2 norm by inducing some sparsity (some weights will be constrained to 0). Lastly, **(iii)** is what will provide MACE with forecasting power. f is chosen to be a Random Forest (RF) for various reasons, some more subtle than others (see section 2.4). Surely, what we want first and foremost, is f to be a solid off-the-shelf predictive model handling nonlinearities and high-dimensional data while keeping overfitting in check without extensive hyperparameter tuning (Goulet Coulombe, 2020b). Clearly, RF checks the first two boxes, along with Boosted Trees and (Deep) Neural Networks (Friedman et al., 2001). The last requirement is met by RF, but not nearly as much by the other two well-known families of ML algorithms. As will become apparent in section 2.4, due to the iterative nature of the MACE (and, in general, the idea of having a function on each side of the equation), RF's easily obtainable out-of-bag predictions, that are resilient to overfitting, will be a key ingredient in our routine.

Initialization. Algorithm 1 is divided into two key steps, which are, intuitively, the updating of the right hand and left-hand side parameters, respectively. We initialize $\hat{z}_{0,t+h}$ as a plausible portfolio. When $\boldsymbol{w} \geq 0$ is activated, such a portfolio is the equally-weighted one (as used in section 4.1). When it is not (as in section 3.1), one can use the solution to the classic (and static) global minimum variance portfolio problem, which is an equally intuitive initialization point, especially given the forthcoming discussion in section 2.2. Regarding $\hat{f}_0(\mathbf{X}_t)$, it is set to 0 and $\eta = 1$ for line 4 in iteration $s = 1$. This simply means MACE is initiated at the equally-weighted portfolio (or else) and its corresponding RF conditional mean.

Given the inherent non-convexity of the objective and the plethora of possible solutions, initialization can matter. This is especially true in extremely low SNR environments and when regressors are generated endogenously – like in our daily returns prediction application. Our approach to the multiplicity-of-solutions problem is in the spirit of deep learning rather than classical econometrics. Indeed, a fair amount of ink has been spent on devising efficient algorithms to uncover the global optimum within a very restricted class of MPP problems, in part because the attained R^2 both in-sample and out-of-sample was regarded as a metric of market inefficiency. We deviate from the statistical philosophy of going at lengths to obtain "true parameters" and rather look for "useful parameters", that is, any solution that can generate value for wealth management strategists. Of course, the two objectives are surely not mutually exclusive, but they entail a different focus. Thus, in our applications, we do not especially care for f being unique nor the truest solution to anything, but rather aim at building a portfolio and a predictive function that *generalizes* well— that is, it maximizes R^2_{train} *and* R^2_{test}. In that spirit, devising mechanisms to maximize R^2_{train} remain essential, but they are coupled with equally relevant algorithmic elements to insure such feats can be reproduced out-of-sample. Limiting overfitting in both f and $\boldsymbol{w}$ *is* a necessary condition to successfully trade such a portfolio. Thus, in the coming paragraphs, we explain the nuts and bolts of MACE, and how it spreads the ML gospel of the bias and variance trade-off to the MMLP problem.

The Random Forest Step and Block Out-of-Bag Subsampling. In many ways, MACE is a traditional EM algorithm, where we optimize certain parameters while keeping others fixed, reverse roles in the following step, and alternate until some stopping criterion is met. Accordingly, the first step predicts a fixed portfolio using RF. Then, predictions are updated as a convex combination of $f^*_s(\mathbf{X}_t)$ (current predictions) and the previous iteration's predictions $\hat{f}_{s-1}(\mathbf{X}_t)$, where the speed of adjustment is determined by the learning rate η.

When constructing $f^*_s(\mathbf{X}_t)$, it is imperative that one does not use RF's *fitted values*, which are inevitably prone to immense overfitting. Indeed, RF's fit always delivers R^2_{train} close to 1 (for any standard tuning parameters combinations) even though the true R^2 is nowhere near that (see Goulet Coulombe (2020b) for an explanation and a barrage of examples with classic datasets). This does not prevent RF from delivering stellar R^2_{test}'s – the traditional object of interest – and it is why the $R^2_{\text{train}} > R^2_{\text{test}}$ differential has mostly stayed under the radar of the ML community. Whenever RF's in-sample predictions are required, one shall use the so-called out-of-bag (OOB) predictions, which are, by construction, immune to overfitting in a *cross-sectional* context – in the sense that their predictive accuracy will be exactly aligned with what one should expect out-of-sample (Breiman, 2001; Friedman et al., 2001). In other words, such predictions include the conditional mean, whatever its quality may be, and little to none true error term (Goulet Coulombe, 2020b). This is particularly crucial in MACE given that such predictions are to be fitted by another ML algorithm in a subsequent step. In a manner, this approximates the ideal "cross-fitting" solution where, in our context, the Ridge Regression step would be conducted on one half of the sample, and the RF step on the remaining half. The latter (rather demanding) scheme is the backbone of so-called honest causal forest (Athey et al., 2019) in heterogeneous treatment effect estimation. In fact, it can be shown that OOB sampling and variants provide a convenient approximation when sample sizes are limited or other practical aspects render plain splitting nonoperational (Chen et al., 2022).

The only thing standing in the way of such properties to be applied to our problem is the time series nature of our data. Indeed, time series dependence in the left-hand side (LHS) or right-hand side (RHS) variables, which creates major complications for classical bootstrap inference, generates similar hurdles for the validity of out-of-bag predictions. While that in $\boldsymbol{r}_{t+h}$ is negligible for $h = 1$, it is certainly not so when considering $\boldsymbol{r}_{t+12}$, the average return between t and $t+12$ (in effect, a sliding moving average). This is even more prevalent in the case of $\mathbf{X}_t$ where predictors, while being stationary, may be quite persistent. This persistence will break the non-overfitting properties of OOB predictions – with an immediate consequence that $f_s^*(\mathbf{X}_t)$ includes overfitted elements of $\hat{z}_{s-1,t+h}$, and thus failing to approximate the LHS and RHS being trained on "truly" separate data sets. All this is obviously related to how time series dependence biases downward bootstrapped standard errors used in small sample frequentist inference (Kreiss and Lahiri, 2012), and the solution to the aforementioned problem – block bootstrapping or subsampling – is backed out from the wide literature on the subject (Hyndman and Athanasopoulos, 2018). Thus, to obtain a $f_s^*(\mathbf{X}_t)$ which is plausibly exempt from overfitting, we will use *block* out-of-bag in-sample predictions. Such techniques have been used to reliably extract more "structural" quantities like various macroeconomic latent states in Goulet Coulombe (2020a) and Goulet Coulombe (2022).

The Ridge Step. The Ridge step takes RF predictions as given and optimizes $\boldsymbol{w}$ so that $\boldsymbol{w}'\boldsymbol{r}_{t+h}$ matches as closely as possible the predictions, in essence, collaborating with f so as to maximize association. The Ridge Regression apparatus comes with trivially implementable, yet healthy and necessary sources of regularization (Carrasco and Noumon, 2011). First, there is λ penalizing extreme allocations and shrinking $\boldsymbol{w}'\boldsymbol{r}_{t+h}$ to the equally-weighted portfolio – as opposed to 0 in a typical Ridge Regression. This is due to the unconditional variance of the portfolio being fixed to 1 (line 6) for identification purposes *during estimation*.[2] Thus, everything being shrunk to the same value is what remains of the original ridge "prior" that every coefficient is shrunk to (the same value of) 0. Given that the resulting portfolio will eventually be rescaled to satisfy the capital budget constraint ($\boldsymbol{w}'\boldsymbol{\iota} = 1$), $1/N$ is the value towards which the shrinkage is effectively pointing at.

Another source of regularization in the Ridge step is obviously the long-only constraint $\boldsymbol{w} \geq 0$. It embeds the prior knowledge that we "unconditionally" expect the market to follow an upward trajectory, and that MACE should preferably focus on portfolios of which it will most often hold a long position. Additionally, for our monthly rebalancing application, limiting the occurrences of overall short positions is desirable from a risk management perspective. This frequently imposed constraint in mean-variance optimization problems also plays here the additional role of limiting the Ridge's step expressivity by

[2] Indeed, it is easy to see in (2) why this is necessary : replacing f by $\zeta \times f$ and g by $\zeta \times g$ (where ζ is an arbitrary scalar) gives rise to the same likelihood. Naturally, this cannot occur when g is the identity function, as in typical regression problems, but is inevitable within ACE and its descendants.

chopping out a wide space of potential w's. As in anything, good regularization balances bias and variance wisely by imposing constraints that will contort our likelihood the least. Accordingly, the implicit prior motivating $w \geq 0$ for one-month ahead forecasts may not always be as well motivated for much shorter horizons – and indeed, we will relax that restriction in the daily application.

Learning Rate. Among the few more subtle technical extensions to ACE, we use out-of-bag block-subsampled predictions and introduce a learning rate η – whose combined action is mostly to curb overfitting and facilitate optimization. Their importance in practice is paramount since we are dealing with a high-dimensional Ridge Regression on one side and a RF on the other, with both having the ability of overfitting the *training* data, even if the other side of the aisle remains static. Directly inspired from Boosting and Neural Networks, the use of a learning rate curbs this problem and avoids zigzagging optimization paths. There is an obvious trade-off between η and s_{max}, with a lower η necessitating a larger s_{max}. In our experience, anything above 0.2 can quickly lead to unstable computations, learning rates above 0.1 will often lead to overfitted solutions (when the SNR is very low), and the optimization of larger portfolios may get suck with too small of a learning rate (like anything below 0.01). What lies within the 0.01-0.1 range usually provides interchangeable results and the symptoms of an impotent learning rate can easily be diagnosed from looking at the path of the in-sample loss.

2.2 Relationship to Mean-Variance Portfolio Optimization

MACE constructs a portfolio to be actively traded. Nonetheless, as we will see later, the raw MMLP portfolio often has nice properties when combined with much more passive trading (e.g., using a prevailing mean instead of RF). Notably, it has fine variance properties, even though variance is not explicitly minimized. Or is it? Furthermore, its predecessor, maximally predictable portfolios, have been noted to have good variance properties without necessarily aiming for it (Ta et al., 2022; Harris et al., 2022). In the brief discussion below, we show that this is no coincidence.

First, note that, in the absence of any predictability, i.e., in the true DGP, the conditional mean is the unconditional mean ($f(\mathbf{X}_t) = \mu \ \ \forall t$), then (3) becomes

$$\min_{w, \ \mu} \sum_{t=1}^{T} (\underbrace{w' r_{t+h}}_{z_{t+h}} - \mu)^2 + \lambda ||w||^2 \qquad \text{such that } \ w \geq 0 \ \text{ and } \ w'\iota = 1$$

where z_{t+h} is the portfolio return h-steps ahead. In population, this problem is

$$\min_{w} \operatorname{Var}[z_{t+h}(w)] + \lambda ||w||^2 \qquad \text{such that } \ w \geq 0 \ \text{ and } \ w'\iota = 1$$

which is a regularized mean-variance optimization problem (as in, e.g., Carrasco and Noumon (2011)) without the minimum return constraint. This constraint, $\mathbf{E}[z_{t+h}(w)] \geq \underline{\mu}$ where $\underline{\mu}$ is a minimal (unconditionally) expected return, bears a different meaning within the MMLP framework simply because the designed portfolio is meant for active trading, not to buy and hold. Nonetheless, as will be discussed in section 3.3, it is possible to bring some of it back in the form of healthy regularization for the MACE allocation by applying an analogous constraint on the mean of the conditional mean. This will in most circumstances, non-trivially improve out-of-sample economic performance.

According to previous observations, what MACE is doing in population, when f is a non-trivial function and the "true" R^2 is larger than 0, is solving

$$\min_{\boldsymbol{w},\ f} \mathrm{Var}[z_{t+h}(\boldsymbol{w}) \perp f(\mathbf{X}_t)] + \lambda||\boldsymbol{w}||^2 \quad \text{such that } \boldsymbol{w} \geq 0 \text{ and } \boldsymbol{w}'\boldsymbol{\iota} = 1.$$

Thus, minimizing the error term in (3) is equivalent to minimizing the residual variance of the portfolio, that is, the share of variance unexplained by the conditioning information. Given that true R^2 are never too far from 0 in predictive financial time series regressions, it is not surprising that MMLPs (or MPPs) have desirable unconditional variance properties. Equivalently, the above can be rewritten as

$$\min_{\boldsymbol{w},\ f} \mathrm{Var}[z_{t+h}(\boldsymbol{w})] - \mathrm{Var}[z_{t+h}(\boldsymbol{w})|\ f(\mathbf{X}_t)] + \lambda||\boldsymbol{w}||^2 \quad \text{such that } \boldsymbol{w} \geq 0 \text{ and } \boldsymbol{w}'\boldsymbol{\iota} = 1 \tag{4}$$

where the apparition of the $\mathrm{Var}[z_{t+h}(\boldsymbol{w})|\ f(\mathbf{X}_t)]$ term highlights an opportunity. From an economic utility maximization perspective, MACE's objective function postulates that it is not volatile returns (before active trading) *per se* that brings disutility, but prediction errors. In a world with small predictive margins, those two distinct objectives are in most contexts approximately equivalent. Nonetheless, this suggests that MACE is eager to handle more *unconditional* variance in the buy-and-hold return if some of it is predictable – and will be compensated by proactive trading based on informative signals. Hence, from the formulation in (4), there is an imminent tension for $\boldsymbol{w}$ in the MACE problem because of its dual mandate, i.e., minimizing $\mathrm{Var}[z_{t+h}(\boldsymbol{w})]$ and maximizing $\mathrm{Var}[z_{t+h}(\boldsymbol{w})|\ f(\mathbf{X}_t)]$. Those may push $\boldsymbol{w}$ in the same direction, or they may not – depending on what lies in $\mathbf{X}_t$ and the shape of f. The benefits of all forms of regularization on risk-reward ratios also become obvious from (4). An overconfident MACE allocation will inflate $\mathrm{Var}[z_{t+h}(\boldsymbol{w})|\ f(\mathbf{X}_t)]$ in-sample. If it fails to replicate predictive gains out-of-sample, it is likely left with a portfolio with higher unconditional variance that the global minimum variance solution, but no meaningful predictability to tame it.

2.3 MACE vs. Predicting Single Stocks Returns Separately

When it comes to time series predictions of stock returns, a popular ML approach is to conduct a pooled (nonlinear nonparametric) regression for a panel of stocks and their corresponding characteristics. Gu et al. (2020) is the prime example, and they translate their predictions into returns via a long-short portfolio strategy. Alternatively, one can model each stock return separately with its own time series regression, but this has important limitations. In contrast to the above, MACE forecasts a single series, the portfolio's return. From the linearity of the portfolio, we have that $\mathbf{E}[z_{t+h}(\boldsymbol{w})|\mathbf{X}_t] = \boldsymbol{w}'\mathbf{E}[\boldsymbol{r}_{t+h}|\mathbf{X}_t]$ where

$\mathbf{E}[\boldsymbol{r}_{t+h}|\mathbf{X}_t]$ is a vector of conditional expectations for each stock. Thus, one can legitimately wonder why not simplifying the algorithm considerably by (i) getting expectations from pooled or individual predictive regressions and then (ii) running the global minimum variance problem on residuals from such regressions. Precisely, solving

$$\min_{\boldsymbol{w}} \sum_{t=1}^{T} \left(\boldsymbol{w}'(\boldsymbol{r}_{t+h} - \mathbf{E}[\boldsymbol{r}_{t+h}|\mathbf{X}_t])\right)^2 + \lambda||\boldsymbol{w}||^2 \quad \text{such that } \boldsymbol{w} \geq 0 \text{ and } \boldsymbol{w}'\boldsymbol{\iota} = 1$$

after obtaining $\mathbf{E}[\boldsymbol{r}_{t+h}|\mathbf{X}_t]$ externally. There are quite a few reasons not to consider such a route, some conceptual and others, practical. All of them are worth mentioning here because they highlight some of MACE's advantage that may so far have gone unnoticed.

$\mathbf{E}[r_{i,t+h}|\mathbf{X}_t]$ is arguably much harder to learn than typical z_{t+h} candidates from aggregate data, simply by the virtue of the latter being a portfolio. Single stock returns contain a lot of variation that cannot be captured by macroeconomic predictors, and with a small fraction of it being explainable by micro-level firm characteristics, often for low-capitalization stocks. What remains is a large amount of noise weakening the potential f through an unappealing SNR and increased estimation error. This crucially matters because (i) the extremely low SNR for separate stock returns is a serious impediment to any algorithm attempting to learn $\mathbf{E}[r_{i,t+h}|\mathbf{X}_t]$ and (ii) the chosen *individual* model will often be one that puts a higher weight on minimizing estimation variance rather than entertaining ambitions to tackle bias^2. Thus, it can easily turn out that the selected/cross-validated f is the null function (or close to it) whereas the true DGP does, in fact, have an f yielding a positive R^2. In other words, choosing and optimizing ML (or any) models to predict $r_{i,t+h}$ separately might be at odds with the final objective of getting a fine estimate of $\mathbf{E}[z_{t+h}(\boldsymbol{w})]$. And because of that, a positive R^2 remains unattainable without exceedingly large samples.

One way out is the pooled (or global) regression approach with firm-level characteristics, where f's potency is revived through much more data and information on cross-sectional variation. Another route is to predict directly what one will end up trading, that is, the portfolio return. By the joint optimization of $\boldsymbol{w}$ and f, $\boldsymbol{w}$ provides f a more easily-forecastable target. Hence, the previously undetected $R^2 > 0$ becomes an attainable target because $\boldsymbol{w}$ collaborates in making f win at the bias-variance trade-off. This is convenient since getting a $\mathbf{E}[\boldsymbol{r}_{t+h}|\mathbf{X}_t]$ vector worthy of use is not necessarily an easy task, requiring large amounts of micro-level data not always easily accessible in real time, and a fair amount of computing resources. In comparison, MACE finds profitable predictability in a convenient low-maintenance setting.

Given the attention they get, predicting indexes such as the S&P 500 rarely deliver sizable R^2s at short horizons. But there are numerous ways into which stocks can be assembled, and some of those blends may be more promising than others from a predictive viewpoint. Linking it back to the econometric literature on the benefits and costs of aggregation (forecasting aggregates vs. aggregating components' forecasts), MACE can be seen as finding the optimal aggregation that keeps variance low (by aggregating) and yet keeps bias^2 similarly low by creating an aggregate with limited aggregation bias from neglecting heterogeneity (Lütkepohl, 2011).

2.4 Why Random Forest?

A natural question to ask is: why Random Forest? In principle, RF could be replaced by any ML algorithm. In practice, not quite so, and for many reasons. First, RF is the only algorithm which provides *internally* out-of-bag predictions. Obviously, nothing prevents a very patient researcher to bootstrap-aggregate Boosting and Neural Networks at every iteration s and incur a substantial computational burden. This is especially true of applications with many observations and regressors. Putting things in perspective, to obtain rightful $f_s^*(\mathbf{X}_t)$'s from Boosting and NN, it would take approximately 500 times (a reasonable number of bootstraps) longer than RF, assuming that the three algorithms have a roughly similar computational time (which is quite generous to Boosting and NN in this application). Alternatively, one can ditch any call on to OOB predictions, and extremely carefully tune hyperparameters. Given the impracticability of such an approach (for anything more complicated than Ridge, Lasso, and derivatives) and known results about the virtues of cross-fitting and analogous methods (Chernozhukov et al., 2018), it appears that justifying the costs of going for such an alternative route would require glowing expected benefits.

There aren't. Boosting, which, is often seen as marginally superior to RF in tabular data tasks, often does so by providing mostly small improvements in high SNR environments – a far cry from our financial application. In fact, with RF being less capricious tuning-wise, it has been reported in many low SNR applications to be equally if not more competitive than Boosting (see Gu et al. (2020) and Krauss et al. (2017) for returns, and Goulet Coulombe et al. (2022) and Goulet Coulombe et al. (2021) for macroeconomic forecasting). Additionally, extensive tuning is often required for Boosting to have an edge on RF, which is highly impractical within an iterative procedure.

Deep Neural Networks, which incredible merits in non-tabular data tasks are indisputable (Goodfellow et al., 2016), are known to still take the backseat to tree-based methods when it comes to tabular data (Grinsztajn et al., 2022). Moreover, it has been the subject of considerable discussion that basic feature-engineering (like creating lags) combined with tree-based methods may outperform NNs with architectures tailored for time series data (Elsayed et al., 2021). Of course, none of this rejects the possibility that letting f be constructed from some sophisticated deep recurrent network of any breed (like those in Babiak and Baruník (2020)) could further improve results. Rather, what it suggests is that this paper's results will not be severely handicapped by leaving the aforementioned extensions for future research.

A last deep learning-based alternative is to consider a neural network with two hemispheres as in Goulet Coulombe (2022) – one linear for the LHS and one for the RHS – with a loss function being the squared distance between their respective outputs, reminiscent of developments in Andrew et al. (2013) and Michaeli et al. (2016). This ditches the need for alternating anything and can be optimized directly through gradient descent. There are quite a few complications, however. The first, quite subtle in nature, is that modern neural networks, very large and deep, vastly overfit the data in-sample, yet produce stellar out-of-sample performance for many tasks. This phenomenon has many names – double descent and benign overfitting among them – and now has a theoretical literature of its own (Belkin et al., 2019; Hastie et al., 2019; Bartlett et al., 2020). The problem this poses for building a MMLP is that, if the most promising neural network attains a $R^2_{\text{train}} \approx 1$ fitting what is almost pure noise, there is neither room nor need *in-sample* to have the LHS collaborate in increasing the fit. Given that our application has a SNR which is a far cry from 1 and those of other typical successful deep learning applications, it is a severe complication. Nevertheless, using (often unstable) small networks, carefully crafting their design, and considering an extensive hyperparameters search – all this to avoid the slightest bit of overfitting in $f^*_s(\mathbf{X}_t)$ – could maybe make the derivation of MMLPs from such an approach less ill-fated. In the concluding remarks, we provide additional thoughts and suggestions on how this could perhaps be done in future research.

2.5 Setting Hyperparameters

Given that MACE incorporates two ML algorithms, it inevitably has quite a few hyperparameters (HP), which, for convenience, are summarized in Table 1. Fortunately, RF is well known to be very robust to tuning parameters choices (with default values often very hard to beat, see Goulet Coulombe (2020b) and references therein), and ridge is sparsely hyperparametrized. We opt for setting tuning parameters to fixed values, with their calibration being motivated by domain knowledge and observations of the (block) out-of-bag error metric ($RMSE_{\text{OOB}}$).

Given the nature of the MMLP problem – the prediction of a non-fixed target – considering a validation set as is often seen in ML forecasting studies in economics and finance (Gu et al., 2020; Goulet Coulombe et al., 2022) is an avenue with strong headwinds and likely limited benefits. This is due to the multiplicity of solutions where identical HPs, when re-estimating the model with more data (e.g., reincorporating the validation data), can lead to different solutions. This is not unique to MACE in the ML realm, as this is a commonly known feature of modern neural networks. Also, there is an imminent tension between maximizing the reliability of the validation set and minimizing the likelihood of moving to a different optimum than what we optimized the hyperparameters for. The former calls for a longer validation set and the latter for a shorter one. In the light of all that, it appears more reasonable to rely on common sense whenever possible, and on the blocked $RMSE_{\text{OOB}}$, which is a proper CV metric for time series (Bergmeir et al., 2018), whenever data-driven guidance is needed.

We first concentrate on those HPs pertaining to MACE's iterative optimization itself. The learning rate η is set at 0.1 for monthly data, which always delivers a stable solution and never gets stuck. For

Table 1: Summary of Tuning Parameters and Their Values in Applications

	Monthly Data	Daily Data ($N \in \{20, 50\}$)	Daily Data ($N = 100$)
η	0.1	0.01	0.05
$s_{\max}$	100	250	500
`stopping.rule`	$s = s_{\max}$	early stopping	early stopping
`mtry`	1/3	1/10	1/10
`minimal.node.size`	20	200	200
`block.size`	24 months	2 months	2 months
`subsampling.rate`	80%	80%	80%
`number.of.trees`	500	1500	1500
λ	$R^2_{s,\text{train}}(\lambda) = 0.05$	$R^2_{s,\text{train}}(\lambda) = 0.01$	$R^2_{s,\text{train}}(\lambda) = 0.01$

daily data, more care is needed given that $\boldsymbol{X}_t$ is not fixed. The smallest learning rate always appears desirable – as often observed for Boosting (Friedman et al., 2001) – but we have noticed that too small of a η coupled with large portfolios may lead to the algorithm not optimizing at all in-sample (analogous to exaggeratedly tiny learning rates for deep learning). Thus, it is set to 0.01 for $N \in \{20, 50\}$, but 0.01 being not large enough for $N = 100$ in-sample loss to start decreasing, we increase it to 0.05.

Closely related are the choice of $s_{\max}$, the maximal number of iterations, and the stopping method. For monthly frequency, we find that $s_{\max} = 100$ is well enough for MACE to converge and that performance never seems to deteriorate substantially with s, even after some plateau is achieved. Thus, the stopping criterion is simply $s^* = s_{\max}$. Things are not so easy with the daily application where regressors are created endogenously. First, we set $s_{\max} = 250$ since η is considerably smaller. Since optimization is much more demanding in this environment, it is not impossible for the OOB error to start increasing beyond a certain s – i.e., suggesting the LHS is starting to overfit. Hence, we set $s^* = \arg\min_s RMSE_{\text{OOB}}(s)$, which can be seen as some form of internal early stopping, a key player in the regularization arsenal of modern deep neural networks (Goodfellow et al., 2016). Early stopping is typically implemented using a validation set, but here, for aforementioned reasons, it is more preferable to use RF's internal error metric.

The next four hyperparameters in Table 1 are those of RF. For monthly data, `mtry` is set to the default value of 1/3, one that is typically hard to beat, except in extremely low SNR environments (Friedman et al., 2001; Olson and Wyner, 2018). At the daily frequency, we noticed that `mtry` $= 1/3$ could never deliver $RMSE_{\text{OOB}}(s) < 1$ for any s. Given that the daily application has a much lower SNR and a sparser $\boldsymbol{X}_t$ (hence, less potential for diversification in RF (Goulet Coulombe, 2020b)), it is not entirely surprising that `mtry` $= 1/3$ might be too large and lead to early overfitting. Thus, we set `mtry` $= 1/10$ which kills two birds with one stone by decreasing computing time sharply.

In a similar spirit, `minimal.node.size` is set to a very high value of 200 in the daily application (nonetheless $\approx$ 1/25 of the training sample size), which greatly helps in easing the daily application's computational burden all the while helping improving performance as measured by the OOB. Default values for `minimal.node.size` usually go up to 10. However, when faced with a low SNR, deep trees are either redundant or harmful, because additional splits allowed by `minimal.node.size` $= 10$ vs. `minimal.node.size` $= 200$ are typically fitting the noise and cancel out through bagging in the out-of-sample projection (Goulet Coulombe, 2020b). Limiting the expressivity of RF is not without precedent for predicting returns, as Gu et al. (2020) report using trees of very limited depth. For the monthly frequency application, we set it to `minimal.node.size` $= 20$, a moderately high value that eases computations without any apparent effect on $RMSE_{\text{OOB}}(s)$ (vs. default values).

The next two tuning parameters of RF are `subsampling.rate` and `block.size`. We set `subsampling.rate` $= 80\%$ for all applications, which is standard. `block.size`, on the other hand, needs to be chosen slightly more carefully to balance two goals. As already hinted at above, too low of a block size will lead to $f_s^*(\mathbf{X}_t)$ including fitted noise, to be subsequently fed into the Ridge Regression. An overkill block size will seriously handicap bagging by limiting the number of different block combinations used to construct the trees in the ensemble, ultimately weakening RF (on the variance side) by decreasing the diversity of trees. Thus, we set `block.size` to be a window size within which any form of meaningful dependence between the first and the last observations will have faded away, for both $\boldsymbol{X}_t$ and $\boldsymbol{r}_{t+h}$. For the monthly application, we set it to two years, which very well cover the dependence in $\boldsymbol{r}_{t+h}$ and the stationarized Welch and Goyal (2007) predictors in $\boldsymbol{X}_t$. In the daily application, since $\boldsymbol{X}_t$ and $\boldsymbol{r}_{t+h}$ are daily returns with minimal time series dependence, two business months appears more than sufficient to maintain the interchangeability of the blocks, and leaving plenty of room for bagging to fulfill its task.

A last hyperparameter for RF is the number of trees. It is not a tuning parameter *per se* because there is no statistical trade-off for its choice: the larger the better, with the only constraint being computational burden (Friedman et al., 2001). Given that RF predictions usually stabilize (by the law of large numbers for an average) well before 500 trees, 500 is usual the default setting for `number.of.trees` in many RF implementations and is what is used for the monthly application. Yet, there is a subtle twist, that leads us to increase `number.of.trees` up to 1500 for the daily application. We are generating regressors endogenously – in essence using lags of a continuously updated target –rather than taking $\boldsymbol{X}_t$ to be fixed observed predictors. In doing so, we may incur attenuation bias in RF attributable to the generated regressor problem. More precisely, "measurement error" can impair RF's ability to detect nonlinear time-series dependence because, $f_s^*(\mathbf{X}_t)$ being out-of-bag predictions, there are an average of $(1 -$ `subsampling.rate`$) \times$ `number.of.trees` trees, which falls to 100 with `number.of.trees` $= 500$. Bumping `number.of.trees` to 1500 makes it an average of 300 single tree predictions for $f_s^*(\mathbf{X}_t)$, which is enough to curb measurement error problems without exploding computational costs.

Finally, we must set a value for λ in the Ridge Regression step. While it could be tempting to crossvalidate λ internally at each step, the optimally chosen λ at each s may be well off from that of the final s, and lead optimization in a poor direction. For instance, in early steps, cross-validation can easily choose a λ that shrinks the portfolio excessively, because, at that early stage s, there is, indeed, very little or no predictability being detected. Moreover, on top of withstanding the additional computational demand, changing λ may lead to certain steps not improving the loss, thereby impairing the EM-style algorithm's ability to minimize the overall loss. Therefore, for the monthly application, we set λ such that it attains an in-sample R^2 of 0.05, which is a high yet not unreachable mark. In the daily application, facing the inevitability that $R^2_{s,\text{test}}$ is unlikely to stand above 1%, λ is chosen at each step so to target $R^2_{s,\text{train}} = 0.01$.

3 Application I: Daily Stock Returns Prediction

We begin our exploration of MACE by constructing maximally ML-predictable portfolios at the daily frequency. The high frequency brings both opportunities and difficulties. Among the former is the availability of more data points, and a lessened need to rely on data going way back to the late 1950s, as is typically the case in monthly exercises. In terms of the latter, an even more hostile signal-to-noise ratio comes to mind, as well as the scarcity of freely available predictors at such frequency.

Here, $\boldsymbol{r}_{t+1}$ comprises individual stock returns for firms listed on the NASDAQ. We keep $N \in \{20, 50, 100\}$ of them with highest market capitalization on January 3rd 2017, which is the date of the beginning of our test sample. Hence, there is no look ahead bias for the *test* sample and the stocks are as liquid as it gets. For computational reasons, we do not re-estimate models and consider a fixed train-test split of the data, as done in, e.g., Chen et al. (2021a) and Fallahgoul et al. (2020). The training set is thus 2000/03/02–2016/30/12 and the test set 2017/03/01–2022/07/12. Thus, the test set includes a fair level of variety in terms of "financial regimes". In chronological order, we have: a relatively quiet period, a crisis period with extremely high volatility, an unprecedentedly bullish bull market, and a long-lasting bear market.

As mentioned above, gathering many exogenous predictors available in real-time for $\mathbf{X}_t$ is not easy and often not cost-free. In this application, we will see whether MACE can generate predictability with nothing more than time-series properties of the portfolio. Creating portfolios or stock combinations that have exploitable persistence properties (for mean reversion- or momentum-based trading strategies) has been studied, for instance, in d'Aspremont (2011). Nonetheless, the near universe of following studies, by the limitations of relying on variants of canonical correlation analysis, are bounded to consider only linear autoregressive properties. Needless to say, if there is any remaining form of mean reversion that has not been drilled out yet, it will need to be complex in f or g – or both. Thus, f being a RF able to estimate any form of nonlinear nonparametric dependence may help in finding mean reversion patterns that remained undetected to the naked eye or simpler algorithms.

Beyond its coverage of heterogeneous financial conditions, the test set is interesting in its own right simply by the virtue of being *recent*. Krauss et al. (2017) finds substantial gains from a simple daily long-short strategy using signals from predicting single stocks separately via tree-based techniques (among others) with lag returns as predictors (as we use). However, and now a recurring theme, the improvements are circumspect to the pre-2010 era, which, as the authors note, is likely due to the widespread dissemination in the 2010s of the very techniques they use. Thus, an interesting question is whether MACE (and other less obvious uses of ML) will find exploitable nonlinear mean reversion in a period where simpler methods could not. In a similar spirit, localized episodes of predictability, reasonably frequent before 2000, are found to be a much rarer event afterwards (Farmer et al., 2022). A related question is whether MACE, through ML-based nonlinearities in f, can capture and exploit those in real time rather than only observing them ex-post.

It is natural to wonder whether MACE-based predictability will outlive its test set, that is, after the profitable pattern is openly communicated to other market participants. MACE's design partly protects against early depletion by (i) forecasting a synthetic security (rather than highly scrutinized stocks or portfolios) and (ii), doing so with a mostly opaque model. Additionally, as discussed in section 2, MACE can generate plenty of solutions, with most of them achieving the predictability goal in different ways. Hence, in principle, there should be no shortage of possibilities for enlarging the set of nonlinearly mean-reverting synthetic securities, especially keeping in mind that our application deliberately focuses on creating them from a narrow set of well-known stocks.

Nonlinear Mean Reversion Machine. In what follows, we describe the remaining building blocks necessary to go from plain MACE to a version specialized for daily data. Throwing many lags of many stocks at the daily frequency directly in $\mathbf{X}_t$ will be at the nexus of computational and statistical inefficiency. A more manageable and promising route is the parsimonious problem

$$\min_{\boldsymbol{w},\ f} \sum_{t=1}^{T} (\boldsymbol{w}'\boldsymbol{r}_{t+1} - f([\boldsymbol{w}'\boldsymbol{r}_{t-1}, \dots, \boldsymbol{w}'\boldsymbol{r}_{t-21}]))^2 + \lambda ||\boldsymbol{w}||^2 \quad \text{such that and } \boldsymbol{w}'\boldsymbol{\iota} = 1 \tag{5}$$

where features – lags of the portfolio returns – are created endogenously given the portfolio weights.[3] There are two substantial modifications to Algorithm 1. The first is that we drop the $\boldsymbol{w} \geq 0$ constraint. Keeping such a constraint in place – as in the monthly application of section 4.1 – would push MACE to find portfolios for which it will most often go long with, and avoid relying extensively on short-selling to turn a profit. Such a restriction comes with the prior that, at the lowest frequencies, the market is always expected to go upward and that a successful strategy should not immensely deviate from this evident fact about the unconditional mean. At the daily frequency, however, the importance of the low-frequency component justifying a long position for the long-run is much tinier– and we want the configuration of the daily MACE to absorb this knowledge. More importantly, $\boldsymbol{w} \geq 0$ appears unnecessary given that the

[3] Note that for a linear f, this optimization problem could be solved by nonlinear least squares or some sort of generalized eigenvalue problem as studied in d'Aspremont (2011) and others.

portfolio will be bought or sold plausibly every day, and single-stock short positions will be short-lived by construction. The relaxation of this constraint now allows MACE to simultaneously hold short and long positions over single assets, even though we may go long or short with the overall portfolio.

The second modification is, obviously, the inclusion of an additional step before the RF step that creates lags of the portfolio as it is constituted at iteration s. Given the SNR faced in this application, cleverly designed regularization is key. For that sake, we apply the MARX (Moving Average Rotation of X) transformation of such lags (Goulet Coulombe et al., 2021) and stack those in $\mathbf{X}_t$. As argued in Goulet Coulombe et al. (2021), the transformation implies more approximate implicit regularization at high frequencies than raw lags themselves. Precisely, in a linear model with an l_2 or l_1 norm on coefficients, this switches the model from shrinking each coefficient towards zero to being shrunk to one another successively. For more complex ML methods where the implicit regularization (almost always entailing the prior that each feature should contribute, but marginally) cannot easily be altered by changing the penalty (like RF or Neural Networks), MARX is a trivial additional step that can help bolster predictability by embedding a more appropriate prior in the model. Its implementation is simply a one-sided moving average of the lagged portfolio return (of increasing length, up to a month) instead of raw lags — which, in the current application, is analogous to a basket of momentum indicators. [4]

We benchmark MACE for each $N \in \{20, 50, 100\}$ against a set of relevant and informative competitors: equal weights and the minimum variance portfolio. The latter correspond to the initialization values of MACE. We also include the S&P 500. Those are all predicted with a prevailing mean and RF. Going forward we denote the prevailing mean models as EW (PM), MinVar (PM), S&P 500 (PM) and the RF models EW (RF), MinVar (RF), and S&P 500 (RF) respectively. RF models are also procured with MARX-transformed features. Finally, we complement those with MACE (PM), which is a portfolio constructed with Algorithm 1 but where RF predictions are substituted out-of-sample by MACE's portfolio prevailing mean. This serves the purpose of evaluating MACE's raw portfolio return (since, in effect, it comes from a well-defined mean-variance problem as per section 2.2's discussion) and quantifying how much of MACE's success comes form leveraging predictability.

Trading. Relative weights are fixed, but absolute weights (i.e., the overall position on the synthetic asset or portfolio) are changing in every period. To transform predictions into trading positions for economic evaluation metrics, we solve the prototypical mean-variance problem for a single return y_{t+1}

$$\underset{\omega_{t+1}}{\arg\max} \quad \omega_{t+1}\hat{y}_{t+1} \; - \; 0.5\,\gamma\,\omega_{t+1}^2\,\hat{\sigma}_{t+1}^2 \tag{6}$$

as is laid out in Filippou et al. (2021) and many others. The risk aversion parameter γ is set to 5 and $\hat{\omega}_{t+1}$ is constrained to lie between -1 for 2 for reasonable allocations.

[4] Note that in a linear model with no regularization, by the virtue of MARX being a rotation of $\mathbf{X}_t$, it does not alter the span of predictors and yield identical fitted values. However, this is not true of regularized and/or nonparametric methods, which is our prediction tool in this paper.

Evaluation Metrics. The evaluation metrics are the out-of-sample R^2, average annualized return (r^A), and the annualized Sharpe Ratio (SR) – where returns are collected from the trading exercise as described in Equation (6). Following Campbell and Thompson (2008) and the ensuing literature, the out-of-sample R^2 of model m for forecasting portfolio return y_{t+1} is defined as $1 - {}^{MSE_m^{\text{OOS}}}/_{MSE_{\text{PM}}^{\text{OOS}}}$ where PM stands for the prevailing mean and $MSE_m = \frac{1}{\#\text{OOS}} \sum_{t \in \text{OOS}} (y_{t+1} - \hat{y}_{t+1|t})^2$. PM is specified to be the historical mean of the training sample. Given the inherent unpredictability of financial markets, this not-so naive benchmark is in fact one that is notoriously difficult to beat.

To bring further enlightenment, we also report R^2 for key subsamples in Table 2. Namely, we report R^2_{CovidW1}, which is the R^2_{OOS} for the onset of the first wave of Covid-19, defined as February, March, and April 2020. Those 3 months were characterized by a level of volatility unseen since the financial crisis, and consists of the only recession in our test set. It is well documented that predictability is more likely to be found during bad economic times (see Li and Zakamulin (2020) and the many references therein). Thus, we wish to investigate whether (i) MACE follows that rule and (ii) if it can find predictability outside of the recessionary episode. Accordingly, $R^2_{\neg\text{CovidW1}}$ is the R^2_{OOS} excluding those three months. In a similar spirit to R^2_{CovidW1}, we include R^2_{2022} (the R^2_{OOS} from the first business day of 2022 until the end of our sample in December 2022) and r^A_{2022}, which is the corresponding annualized return for the same era. This allows to shed light on (i) whether there was any meaningful predictability to be found in the long-lasting bear market of 2022 and (ii) whether active daily trading with MACE or other RF-based strategies can avoid the sharp losses of the US stock market in 2022.

We complement this extended set of metrics with Keating and Shadwick (2002)'s Omega Ratio (Ω), an increasingly popular measure of the risk-reward ratio that leverages all the moments of the distribution of returns (whereas SR only exploits the first two). This measure is particularly useful in situations where the distribution of returns is skewed, as only negative deviations from a certain threshold – e.g. the mean return of a benchmark, or an investor's desired average return – contribute to the risk component. First, we do find *positive* skewness in MACE-based returns. Second, and in a more striking fashion, RF-based returns, when predictability is non-negligible, are found to be much more leptokurtic (i.e., Laplacian-looking) than those of other strategies, even when excluding CovidW1. This is not unheard of for ML-based strategies (Chen et al., 2021b). Thus, to compare apples with apples without the inherent assumption of normality in SR and to avoid penalizing similarly both large positive and negative returns, we include Ω in Table 2 as a supplementary risk-reward ratio.[5]

3.1 Results

We report relevant summary statistics for our daily exercise in Table 2. Log cumulative return plots for the small ($N = 20$) and large ($N = 100$) portfolios are shown in Figure 2 and R^2 Comparison of MACE to

[5] The expected benchmark model return in Ω used as a cutoff is the mean return of the S&P 500 in the training set (Balder and Schweizer, 2017). Doubling it does not alter rankings.

Table 2: Summary Statistics for Daily Stock Returns Prediction

	R^2_{OOS}	R^2_{CovidW1}	$R^2_{\neg\text{CovidW1}}$	R^2_{2022}	r^A	SR	r^A_{2022}	Ω
N = 20								
MACE	**3.42**	**7.86**	**0.56**	-0.55	**23.10**	0.99	5.03	**1.18**
MACE (PM)	0.01	-0.05	0.04	-0.12	18.71	**1.04**	0.75	1.13
EW (RF)	-4.71	-9.39	-1.21	**0.33**	9.15	0.34	**36.43**	1.02
EW (PM)	-0.04	-0.06	-0.03	-0.10	14.28	0.78	-10.23	1.08
MinVar (RF)	-1.57	-2.69	-0.80	-0.61	7.55	0.42	2.94	1.01
MinVar (PM)	-0.01	-0.04	0.02	-0.13	18.13	1.00	-3.32	1.12
N = 50								
MACE	**0.89**	**3.88**	-1.07	-0.14	**20.42**	0.91	**18.41**	**1.14**
MACE (PM)	-0.04	-0.04	-0.04	-0.03	14.52	0.78	4.66	1.08
EW (RF)	-1.01	-0.95	-1.06	-0.87	9.82	0.37	4.55	1.03
EW (PM)	-0.05	-0.05	-0.05	-0.06	13.23	0.73	-8.30	1.07
MinVar (RF)	0.75	1.94	**-0.01**	-0.25	17.32	**0.96**	3.48	1.11
MinVar (PM)	-0.04	-0.04	-0.03	**-0.02**	14.39	0.83	7.19	1.08
N = 100								
MACE	**4.05**	**12.20**	**0.86**	**0.23**	**41.36**	**1.59**	**23.93**	**1.33**
MACE (PM)	-0.02	0.01	-0.03	-0.22	15.20	0.91	-16.81	1.09
EW (RF)	0.00	1.17	-0.99	-0.94	9.88	0.38	-10.48	1.03
EW (PM)	-0.05	-0.04	-0.06	-0.03	12.62	0.69	-10.26	1.06
MinVar (RF)	0.96	2.06	0.25	-0.16	18.87	0.99	-7.01	1.12
MinVar (PM)	-0.02	-0.05	0.00	0.06	10.18	0.58	-6.39	1.04
S&P 500 (RF)	2.88	7.41	-0.14	0.09	13.29	0.64	2.80	1.09
S&P 500 (PM)	-0.01	-0.06	0.03	-0.32	11.65	0.69	-17.98	1.06

Notes: The first column-wise panel consists of out-of-sample R^2's for different test (sub-)samples. The second are economic metrics, where r^A := Annualized Returns, SR := Sharpe Ratio, r^A_{2022} := Annualized Returns for 2022, Ω := Omega Ratio. All statistics but SR and Ω are in percentage points. Returns and risk-reward ratios are based on trading each portfolio using a simple mean-variance scheme with risk aversion parameter $\gamma = 5$. PM means the prediction is based on the respective prevailing mean with a lookback period of ten years, while RF means using that of a Random Forest. The 4 row-wise panels are for portfolios of $N \in \{20, 50, 100\}$ stocks and the S&P 500 index. Numbers in **bold** are the best statistic within portfolios of the same size. Numbers in **green** are the best statistic of the whole column (that is, across all portfolio sizes and including S&P 500).

Random Alternatives plots are available in Figure 1. To alleviate notation, MACE ($N = 100$, PM) will be written as MACE$_{100}$ (PM) and other accordingly. Additionally, MACE using RF for prediction is simply denoted MACE.

Statistical Results. For all three MACE specifications, we find strong evidence of predictability through time-series dependence at the daily frequency. Out-of-sample R^2's are abnormally high and distance most of the competitors for all subsamples but 2022. In the latter case, only MACE$_{100}$ achieves a positive R^2 that narrowly beats that of S&P 500 (RF). The bulk of predictability is indeed found during the first wave of the Covid-19 pandemic, with local R^2 for the three MACEs ranging from 3.88% to a stunning 12.20%. While MACE hits the two highest marks for the era, unusually high R^2's (taking Farmer et al. (2022)'s local predictability results as a reasonable yardstick) are not its exclusivity. S&P 500 (RF) also delivers a large R^2 during the era (7.4%), and EW_{100} (RF) and MinVar_{100} (RF) are getting 1.17% and 2.06%, respectively.

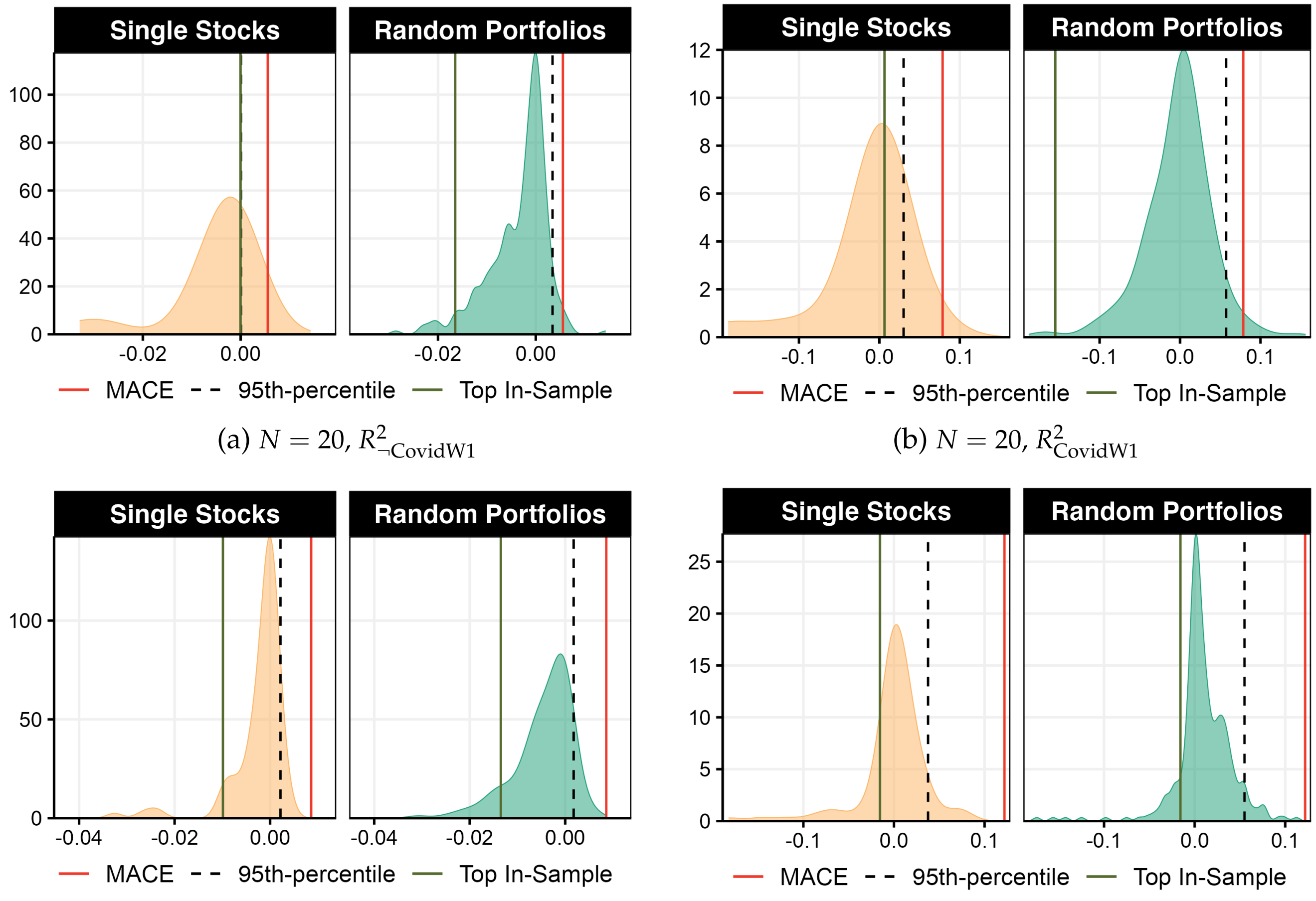
What is more exclusive, however, is predictability outside of the turbulent spring of 2020. MACE_{20} and MACE_{100} achieve it both with 0.56% and 0.86%, which are sizable at the daily frequency, especially in *good* times (Li and Zakamulin, 2020; Farmer et al., 2022). All other RF-based models fail to deliver $R^2_{\neg\text{CovidW1}} > 0$, all situated in the vicinity of -1% , except for MinVar_{100} (RF) at 0.25%, the closest competitor to MACE on this metric. Obviously, negative R^2's at such a frequency and with so little conditioning information are what one would expect. [6] Nonetheless, two MACEs out of three outperform this predicament. And it holds up in 2022. We see that MACE_{100} has a marginal outperformance during the bear market at 0.23%. For other models, negative R^2 are again the norm rather than the exception, with notable deviations by S&P 500 (RF) at 0.09% and EW_{20} (RF) at 0.33%. However, in the latter case, it delivers the worst R^2 of any model for the other three subsamples. In fact, MACE_{100} is the only model with four positive R^2 out of four.

It is interesting to get a sense where MACE's out-of-sample R^2's stand with respect to random alternatives, like single stocks (in effect corner solutions of the MACE) and random portfolios. Especially, in the latter case, it can be seen as an implicit out-of-sample statistical test for the procedure itself. It aims at answering: if we were to draw random stock combinations and predicting them with RF rather than running MACE, how many of those would fare better out-of-sample? The location and shape of the distribution will also be informative about how much room MACE had to find a MLPP. We draw 150 such random portfolios with $\boldsymbol{w} \geq 0$ imposed, and 150 without.

Figure 1 reports such results for $R^2_{\neg\text{CovidW1}} > 0$ and $R^2_{\text{CovidW1}} > 0$ (forming two complementary samples). First, we observe how the prospects of predictability change from CovidW1 to ¬CovidW1, with the random portfolios distributions showing negative means and clear negative skewness without CovidW1 data for both $N = 20$ and $N = 100$. $R^2_{\text{CovidW1}} > 0$'s for random portfolios see a sharp decrease in negative skewness for both portfolio size. In fact, for $N = 100$, skewness visibly becomes mildly positive. Additionally, we can see a shift in location, with $N = 20$ random portfolios' mean being approximately 0 and that of $N = 100$ being mildly positive. A similar pattern is observed for single stocks, but is inevitably rougher given $N < 300$ and single stocks having higher variance than linear combinations of them.

Those distributions, in themselves, highlight some things we already know, like the immense difficulty of finding predictability outside of "bad" economic times: almost 95% of random portfolios have a $R^2_{\neg\text{CovidW1}} < 0$, yet R^2's greater than 0 constitute approximately 50% of the R^2_{CovidW1}. Obviously, MACE's

[6] In a recent evaluation from an out-of-sample period overlapping with ours, Haase and Neuenkirch (2022) get nearly all negative R^2 for 20 models using vast conditioning information for prediction at the weekly frequency.



(a) $N = 20$, $R^2_{\neg\text{CovidW1}}$

(b) $N = 20$, R^2_{CovidW1}

(c) $N = 100$, $R^2_{\neg\text{CovidW1}}$

(d) $N = 100$, R^2_{CovidW1}

Notes: This plot shows distributions of OOS-R^2's, for different portfolio sizes and subsamples. The "Single Stocks" panel reports the distribution of the N R^2's obtained from predicting each stock in the panel separately with RF. The "Random Portfolios" panel shows the distribution of 300 R^2 obtained from predicting randomly drawn portfolios with RF. *Top In-Sample* denotes the OOS-R^2 of the single stock, random portfolio respectively, that achieved the highest R^2 during training. *95th-percentile* denotes marks the 95$^{\text{th}}$ percentile of the corresponding distribution shown in the graphs.

Figure 1: R^2 Comparison of MACE to Random Alternatives for the Daily Application

objective is not only to beat those odds by systematically landing on the "good" side of the distribution, but also to strive for the rightmost part of it. And it is what we see in all four cases of Figure 1. Indeed, the red line is to the right of the dashed line, meaning we can reject the null (at 5% confidence level, against a one-sided alternative) that MACE is randomly drawn portfolio. Also, the red line is always to the right of the best in-sample single stock or random combination.

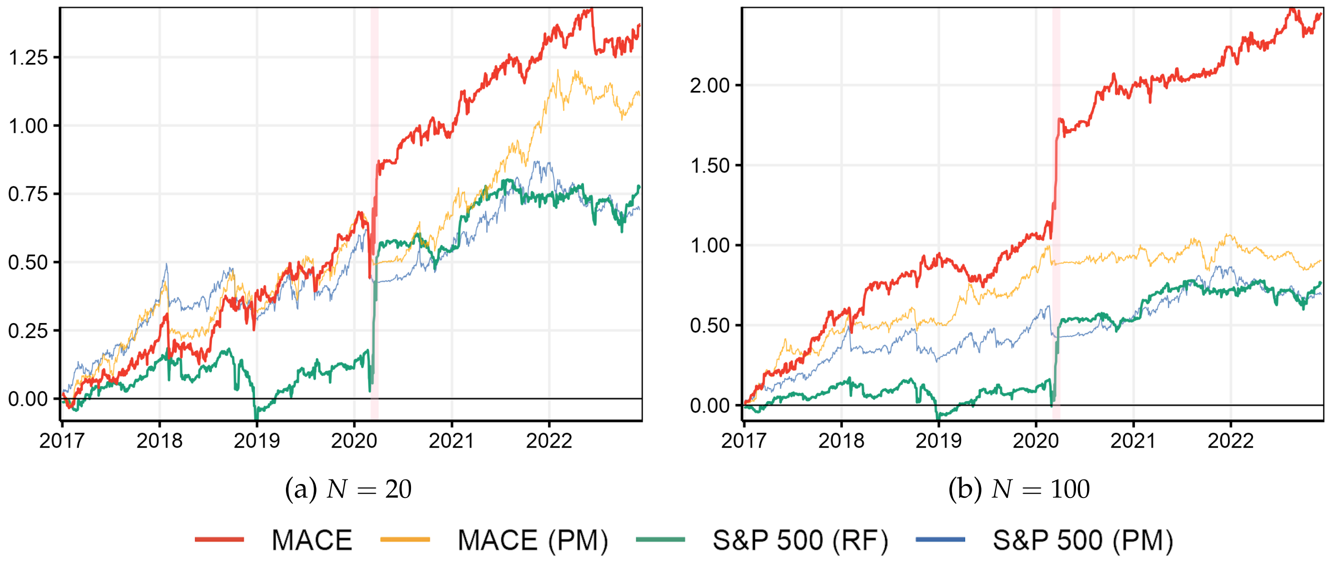
Economic Results. From Table 2, we see that MACE generates the highest return for all portfolio class sizes, with MACE$_{100}$ leading the march with a massive 41.36% annualized return over 2016-2022. MACE$_{20}$ and MACE$_{50}$ deliver more "reasonable" returns of 23.1% and 20.42%, which are nonetheless meaningfully higher than alternatives. Closest contenders include MACE (PM) itself and sometimes MinVar (PM or RF, depending on N). Hence, only MACE appears to consistently deliver the highest return. Obviously, that could all be at the expense of significantly more risk.

Indeed, we see that the variance of MACE's returns often appears to be higher than that of alternatives since its SR is narrowly behind that of alternatives for $N = 20$ (0.99 vs. 1.04) and $N = 50$ (0.91 vs 0.96). Note, however, that MACE is consistently among top contenders whereas, e.g. , MinVar (RF), delivers the leading 0.96 for $N = 50$ and the second-to-last 0.42 for $N = 20$. For $N = 100$, MACE gets a dominating annualized SR of 1.59, suggesting most of the 41.36% return it gets is not due to unshackled variance. The closest alternative among all N's is $SR = 1.04$ for MACE (PM, $N = 20$).

Figure 2 is telling about how that came about. First, there are the eye-grabbing flash gains during the onset of the Pandemic. These are unpacked in their own section 3.2. Given that major crisis only appear to occur on semi-decadal frequency, it is natural to wonder whether MACE$_{100}$'s SR would still be as startling without its miracle run in March 2020. By peaking at Figure 2b, it seems so: translating downward the red line by 0.6 from 2020 onward still make it land comfortably above competitors in terms of final cumulative returns. Moreover, those are increasing in a nearly linear fashion starting from 2018. The SR excluding the highly profitable month is 1.32, which confirms visual observations. Looking at MACE$_{100}$ (PM)'s cumulative returns is also revealing for MACE$_{100}$'s overall performance. The latter is magnified during the early stages of the Pandemic because the former (i.e., the raw portfolio itself) suffers minimal losses. However, we see that MACE$_{100}$ (PM) proves inferior to its RF-based counterpart by delivering approximately 0 returns in 2018, 2020 as well as 2021, and losses in 2022.

The r^A_{2022} column also helps in understanding overall returns. The variance among results for this metric is vast. Some strategies suffered important losses, yielding returns that are quite correlated with the overall bear market. Others turned into a massive profit. All three MACEs do fine, with MACE$_{100}$ delivering almost 24%, with apparently (Figure 2) higher variance than previous years, however. MACE$_{20}$ is lowest among the three, with 5.03% and facing a major setback midyear by losing 10% in a bit more than a week. We will see in section 3.3 that such setbacks can be smoothed out, most notably, by "bagging strategies". While MACE$_{20}$ and MACE arguably get a headstart for 2022 with raw portfolios (PM) delivering marginally positive returns, that of MACE$_{100}$ (PM) is a dramatic -16.81%. In all three cases, it is the effect of active trading using RF predictions that avoids the failure of many strategies in 2022. It is also true, to a lesser extent, for S&P 500 (RF) which turns in 2.80% while the PM version suffers major losses. In fact, there are quite a few RF-based strategies that are profitable in 2022, however, unlike MACEs, those are not accompanied with enviable returns in less turbulent years.

The large positive jumps in returns that many RF-based methods experience may be seen unfavorably by SR. From our inspections, many RF-based strategies generate returns that are always more leptukurtic – almost Laplacian-looking – than strategies based on the prevailing mean. For instance, excluding CovidW1, MACE$_{100}$ has a kurtosis of 5.96 vs 2.43 for MACE$_{100}$ (PM), and S&P 500 (RF) has 9.52 vs 6.87 for its prevailing mean version. Again with the aforementioned exclusion zone, we also notice that MACE$_{100}$ returns have positive skewness (0.2), which is highly preferable, whereas all other strategies have negative skewness. The other two RF-based MACE's do not have positive skewness,



(a) $N = 20$

(b) $N = 100$

Figure 2: Cumulative Returns

but it is always the least negative among its portfolio size group. Including CovidW1 only magnifies (substantially) the extent of such findings. Thus, a performance measure that appreciates such subtleties about return distribution, going beyond mean and variance, may give a different assessment of portfolios predicted by RF. This is indeed what we find. MACE (non-PM) portfolios always deliver the highest Ω within N-wise groups. Notable changes in ranking with respect to SR are MACE$_{20}$ going from (narrowly) third to clearly first within its group, MACE$_{50}$ moving from second to first with the $N = 50$ group, and S&P 500 (RF) beating S&P 500 (PM).

Among other stylized facts, we observe that, in line with theoretical observations in section 2.2 – equation 4 in particular – the variance of MACE portfolio return *before trading* is higher than all other portfolios. For instance, MACE$_{100}$ unconditional standard deviation is 1.54 out-of-sample excluding CovidW1 and 1.44 in-sample, whereas the MinVar portfolio (effectively MACE$_{100}$ at $s = 0$) has 0.56 and 0.69. Thus, in its quest for higher predictability and returns, MACE creates a synthetic asset which unconditional volatility is higher and which is tamed through more accurate predictions. We see that this higher volatility is no out-of-sample "surprise" as the training and test nearly coincides on this metric. This is not true for MinVar. Those observations highlight both opportunities and perils for the MACE. The opportunities have already been documented widely. The peril is that of overfitting: letting MACE overfit will lead to higher unconditional variance out-of-sample than what can be inferred from the in-sample performance. From a risk-reward perspective, this could be a lose-lose situation. Clearly, the MACEs studied here are doing fine in that regard, but one should always bear in mind the dual costs of overfitting in the MMLP problem.

In Appendix A.1, we report how and MACE$_{20}$ and MACE$_{100}$ performances are affected from introducing various levels of transaction costs (TC). MACE$_{20}$ is mostly unaffected under reasonable TC levels, loosing about 1% in returns for any 0.5 percentage points increase in TCs. However, given the highly-liquid nature of the considered stocks, a more realistic ballpark is 0.1% as recommended by De Nard et al. (2017) for trading Volatility Lab's 177 sustainable funds. MACE$_{100}$ inevitably takes larger hits (at worst -10% returns with 1% TCs). Nonetheless, even in such strenuous conditions, it remain by far the dominant strategy as per all performance metrics. Smaller TCs keep MACE$_{100}$'s annualized return still above 35% and risk-reward in a enviable place.

3.2 Understanding March 2020

It is quite a sight in Figure 2 that MACEs generate massive gains during the short-lived Pandemic recession. These can obviously be linked back to the abnormally high R^2's observed for that era.[7] A similar result is found for S&P 500 (RF). However, S&P 500 (RF) is brought down by a dismal performance for the three years preceding 2020. A similar pattern holds for EW (RF) (unreported), where returns are dismal for the entirety of the test set. This highlights that RF-based models can most definitely fail to outperform basic strategies and that the MACE "treatment effect" is paramount in giving them consistently the upper hand.

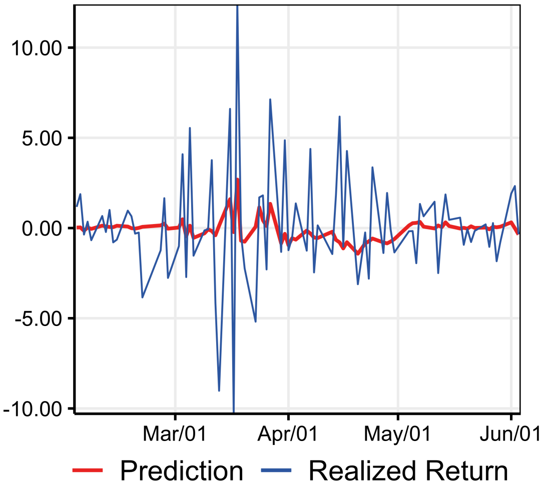


Figure 3: MACE$_{100}$ during CovidW1

MACEs' and S&P 500 (RF)'s winning streak (both for $N = 20$ and $N = 100$) spans approximately 21 business days starting from early March 2020. Its intensity is greater for MACE$_{100}$ and S&P 500 (RF) where wealth is approximately tripled in one month. Indeed, zooming in on these precise days, MACE$_{100}$ **hits a local R^2 of 20% and its sign prediction accuracy is 78 %**. Needless to say, losing money only once every five days, thus compounding returns most days of the week in a time of high volatility, is what generates the rocket increase in profits during March 2020.

Predictability patterns are clearly visible in Figure 3, where the RF embedded in MACE predicts many of the bounce backs and the overall zigzagging nature of the market in times of crisis. To the naked eye, it looks like RF, in part, uses strong negative autocorrelation from one day to the next–thus, fast-paced mean reversion. Table 3 verifies such intuition with a first order approximation to nonlinear day-to-day dynamics. We find that the AR(1) coefficient for both MACE$_{100}$ and S&P 500 are strongly negative and highly statistically significant (t-stats above 4) during the first four months of the Pandemic. RF predictions provide significant economic value in this period because they precisely capture just that—those are strongly negatively correlated with yesterday's return.

With Corr$(\hat{r}_t^{\text{RF}}, r_{t-1})$ for the two targets in vicinity of -0.6, it is natural to ask where RF may have

[7] Increased predictability in the early Covid era has also been noted in *in-sample* analyses such as Lalwani and Meshram (2020) who found that the stock returns in certain industries are statistically significantly more predictable during 2020Q1 than a few months prior.

Table 3: First Order Approximation to Nonlinear Dynamics in Returns

	CovidW1		¬CovidW1		2008	
	MACE_{100}	S&P 500 (RF)	MACE_{100}	S&P 500 (RF)	MACE_{100}	S&P 500 (RF)
Coefficient	-0.451	-0.402	-0.074	-0.036	-0.104	-0.156
Standard Error	0.097	0.100	0.027	0.027	0.063	0.062
$\text{Corr}(\hat{r}_t^{\text{RF}}, r_{t-1})$	-0.616	-0.577	0.102	0.014	-0.243	-0.173

Notes: This table reports the AR(1) coefficient and its standard error for two different return series on two non-overlapping subsamples of the test set spanning from 2016 to 2022 as well as 2008 from the training sample. $\text{Corr}(\hat{r}_t^{\text{RF}}, r_{t-1})$ is the correlation between Random Forest's prediction of the portfolio's return and the realized return on the previous business day.

learned that. It is equally natural to conjecture it did so during the last major financial crisis. The third panel of Table 3 verifies that. We see, albeit in a marginally more subtle way, that (i) returns are significantly negatively autocorrelated on a daily basis and (ii) RF predictions leverage the phenomenon to a non-trivial extent.

While some statistically significant mean-reversion remains in MACE_{100} outside of the pandemic, it is smaller by orders of magnitude. For the S&P 500, it is completely gone, one would expect. Thus, given the wide heterogeneity, a successful model must detect *ex-ante* whether we are in a state of strong daily mean reversion or one where there is little to none at all. It is an aspect in which the nonlinear nature of RF becomes key. This state-dependence is obviously only one of the many time series nonlinearities that tree ensembles can capture (Goulet Coulombe, 2020a). Finally, it is worth remembering that, even within those two hypothetical states, the linear approximation in Table 3 only captures a fraction of RF nonlinear predictive dynamics. This is particularly true for ¬CovidW1 where RF's prediction correlates very little with the first lag. Even during CovidW1, it is worth noting that the first lag explains less than 40% of the variance of RF predictions.

3.3 Bagging Strategies and Other Algorithmic Refinements

In a real-life implementation of a daily trading strategy, one may be more than willing to exchange transaction costs for computational costs. This is particularly true for MACE-based strategies, where all the computational burden is incurred while finding w, which remains fixed ever after, limiting daily computing costs to that of making one prediction with a Random Forest.

In this subsection, we activate two algorithmic refinements that help in improving the $N = 20$ results in terms of annualized returns and Sharpe Ratios. First, we introduce a modification to the Ridge Regression step that brings back part of the unconditionally expected return constraint that is usually part of classical mean-variance portfolio optimization. In our setup, it will have a second nature as unconditional return *regularization*. Second, we introduce "bagging of strategies" as a reasonably intuitive way to (i) tame the variance of returns and (ii) decrease the dependence of the solution on initialization values. This latter refinement, basically ensembling strategies, multiplies the computational cost by the size of the ensemble (B).

The Return of the Minimum Return Constraint. Given that MACE has a conditioning set and builds a portfolio purposely for the sake of actively trading it, imposing an unconditional expected return constraint in the Ridge step does not nearly appear as natural as it would in a traditional mean-variance optimization setup. Notwithstanding, the close relationship between MACE and MinVar made explicit in section 2.2 suggests that bringing back part of the constraint, in one form or another, could be beneficial. One such scenario occurs if MACE's predictive power is overstated in-sample, and leads it to overly rely on predictability to make an otherwise highly unprofitable portfolio profitable. Mere overfitting can lead the out-of-sample solution of MACE to be closer to MinVar (where the conditional mean is replaced by an unconditional mean). For those reasons, unconditional return regularization may prove a helpful foolproof.

The implementation is simple: in MACE$_{\mu \geq \underline{\mu}}$, we turn off the intercept in the Ridge Regression step and add a positive value $\xi = 1$ to $\hat{f}_s(\mathbf{X}_t)$. This has the effect of tilting the solution towards an allocation which has a higher unconditionally expected return in-sample. Intuitively, it pushes the ridge coefficients not only to reward each individual stock association with predictions $\hat{f}_s(\mathbf{X}_t)$, but also to reward historically higher growth stocks. Conversely, taking short positions on, e.g., Apple and Amazon, is discouraged unless completely hedged with other assets.

Bagging Strategies. Interestingly, and, in fact, as a matter of necessity, we start by showing that the ensemble of strategies can be reduced to a single aggregate strategy, hereby avoiding the bag of strategies to multiply transaction costs by B. First, it is worth remembering that, unlike when predicting a fixed target, it is not possible here to merely average predictions. Those are attached to inevitably heterogeneous targets, and there is no guarantee that the average prediction will be appropriate for the average portfolio. An extreme case is that, in an ensemble of two models with, for estimation b to be the mirror image of estimation b' so that $\boldsymbol{w}_b + \boldsymbol{w}_{b'} = \mathbf{0}$, then the predictor and the predictand is 0 for all observations (even though each estimation separately had a positive R^2). Thus, the ensembling logic must be pushed further than merely averaging models, and rather look for bagging *strategies*.

The proposed bagging scheme is the following: we run MACE B-times with different initializations, collect the B predictions, translate this into B positions through (6), and then, finally, average the returns of a total of B strategies. However, stated as such, this would imply carrying at worst $N \times B$ trades a day instead of N. Fortunately, the bag of strategies can be rewritten so that it collapses to a single strategy. Precisely, we have that

$$r_t^{\text{bag}} = \frac{1}{B}\sum_{b=1}^{B} \omega_{t,b} \sum_{j=1}^{N} w_{j,b} r_{j,t} = \sum_{j=1}^{N} r_{j,t} \frac{1}{B} \sum_{b=1}^{B} \omega_{t,b} w_{j,b} = \sum_{j=1}^{N} w_{j,t}^{\text{bag}} r_{j,t}$$

where $w_{j,t}^{\text{bag}} = \frac{1}{B}\sum_{b=1}^{B} \omega_{t,b} w_{j,b}$. In words, the bag of strategies is equivalent to a single strategy where the daily weight on stock j is $w_{j,t}^{\text{bag}}$, implying at most N transactions per day.

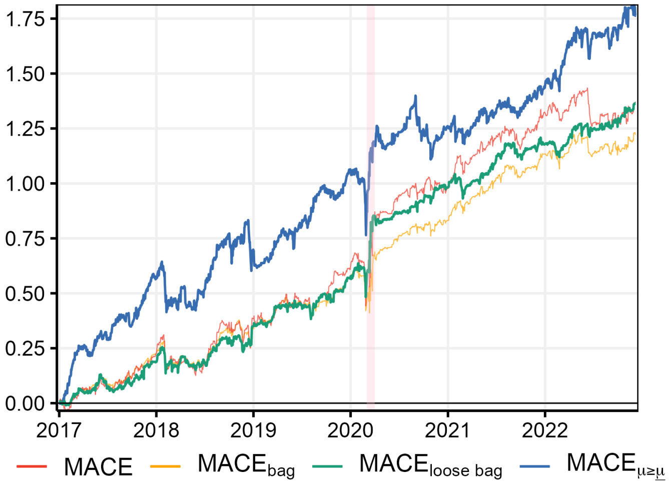
We introduce two sources of randomization to make $\boldsymbol{w}_b$'s differ. First, the minimum variance solution used for initialization is randomized by estimating the covariance matrix on a random subsample of 70% of the training data. Second, we use decreasingly stochastic optimization steps via random observation weights which variance decrease with iterations ($\kappa_{t,s} \sim \exp(s)$) in the Ridge part (Step 5 in Algorithm 1). This mild source of randomness is completely shut down when it becomes negligible ($\kappa_t = 1 \ \forall t$ if $s > \frac{s_{max}}{3}$). The inspiration behind this randomization agent are some implementations of Boosting where trees are fitted on subsamples of the training data, or simply stochastic gradient descent in neural networks. Beyond creating a diversified ensemble, it may help in avoiding early trivial overfitting solutions and in getting unstuck from local minima. The choice of the exponential distribution (vs. subsampling) allows to keep all observations in at all times and is motivated from the Bayesian Bootstrap (see, e.g., the treatment in Taddy et al. (2015) or Goulet Coulombe (2020a)) .

Table 4: Refinements for MACE$_{20}$

	r^A	SR	Ω
MACE	23.10	0.99	1.18
MACE$_{\text{bag}}$	20.60	1.07	1.20
MACE$_{\text{loose bag}}$	23.03	**1.36**	**1.25**
MACE$_{\mu \geq \underline{\mu}}$	**29.76**	1.17	1.19

Notes: Economic metrics are r^A := Annualized Returns, SR := Sharpe Ratio, Ω := Omega Ratio. All statistics but SR and Ω are in percentage points. Returns and risk-reward ratios are based on trading each portfolio using a simple mean-variance scheme with risk aversion parameter $\gamma = 5$. Numbers in **bold** are the best statistic of the column.

Figure 4: Cumulative Returns



We use $B = 50$. Note that, by construction, the annualized return of MACE$_{\text{bag}}$ will be the average of the B annualized returns (by the linearity of means). However, its volatility may be lower than the sum of each run's volatility, resulting in improved Sharpe Ratios. Another refinement is MACE$_{\text{loose bag}}$,which is inspired from RF itself. The rule for λ is changed from $R^2_{s,\text{train}}(\lambda) = 0.01$ to $R^2_{s,\text{train}}(\lambda) = 0.02$ so to decrease the "bias" of base learners at the cost of increased variance, and finally letting the ensembling step take care of bringing down the overall variance.

Results are reported in Table 4. All proposed extensions refine the original MACE$_{20}$ results. MACE$_{\mu \geq \underline{\mu}}$ increases dramatically expected returns, which comes at the cost of reasonably increased volatility. It outperforms the simpler MACE specification for both risk-reward ratios, although the improvement is incremental for Ω. The bagging portfolios have a markedly different behavior: they have a marginally decreased r_A but greatly decreased variance. As a result, MACE$_{\text{loose bag}}$ and MACE$_{\text{bag}}$ both ameliorate over the reference specification, with MACE$_{\text{loose bag}}$ being the superior refinement both in terms of SR and Ω. The reasons behind such remarkable improvements are apparent in Table 4: both

MACE$_{\text{bag}}$ and MACE$_{\text{loose bag}}$ follow an almost linear – in log-terms – trajectory without suffering from any outstanding drawdowns. In particular, the latter fares as well 2022 as in any other year, and mid-March 2020 is only a momentaneous interruption of its otherwise steady exponential growth path.

In Appendix A.1, we report that MACE$_{\text{loose bag}}$ and MACE$_{20}$ outperformance is mostly unabated when accounting for 1% transaction costs, a very estimate conservative in this application to the largest capitalization on the NASDAQ. Lower transaction costs deliver nearly identical returns to those reported above, and along with it SR's and Ω's.

4 Application II: Monthly Stock Returns Prediction

Lastly, we test MACE on monthly data. We shy away from any eccentricity and consider building portfolios with individual stock returns from CRSP for firms listed on the NYSE, AMEX, and NASDAQ (Gu et al., 2020) using the 16 macroeconomic indicators of Welch and Goyal (2007) as the basis for $\mathbf{X}_t$. We conduct a pseudo-out-of-sample expanding window experiment with a training set starting in 1957m3. The test set originally starts in 2005m1 and ends in 2019m12. We re-estimate MACE and the suite of competing models every 3 months and at each t, $\boldsymbol{r}_{t+1}$ is comprised of all the stocks that have been continuously present in the dataset from 1957m3 until t. Accordingly, the number of stocks included in 2005 is 192 and shrinks to 113 in 2019.[8] We also report results when starting the test set in 1987 (as Gu et al. (2020) do) but regard those starting in 2005 as more indicative of performance since MACE not only needs to learn a complex f (as in any ML-finance paper) but also g, and all that with time series of limited length. Also, predictability is known to be harder to find starting from the mid-2000s, with many studies reporting important gains from ML-based stock returns forecasting, but those nearly all take place before the start of the new millennium (Gu et al., 2020; Babiak and Baruník, 2020).

Regarding variable transformations, we subtract the risk-free rate from each stock return in $\boldsymbol{r}_{t+1}$. Considering the exact composition of $\mathbf{X}_t$, we first-difference the clearly non-stationary series in Welch and Goyal (2007) and include 12 lags of each. The evaluation metrics are similar to those introduced in section 3.1. We complement those with the maximum drawdown (DD^{MAX}) as in Gu et al. (2020):

$$DD^{\text{MAX}} = \max_{0 \leq t_1 \leq t_2 \leq T} (Y_{t_1} - Y_{t_2})$$

where Y_t is the log cumulative return from t_0 through t. For computing $MSE^{\text{OOS}}_{\text{PM}}$ in the denominator of the out-of-sample R^2 we again compute PM as the historical mean of the training-set.

Given that available stocks are changing throughout the dataset, so does the composition of EW and MACE portfolios. Moreover, even with a fixed basket of available stocks, MACE's portfolio weights can change slowly as new data points enter the training set. Accordingly, the R^2 (and the other metrics as

[8] Naturally, if the attrition were to become problematic, one could alleviate it by considering a rolling window instead.

Table 5: Summary Statistics for Monthly Stock Returns Prediction

	01/2005 - 12/2019					01/1987 - 12/2004				
	R^2_{OOS}	r^A	SR	DD^{MAX}	Ω	R^2_{OOS}	r^A	SR	DD^{MAX}	Ω
Main Results										
MACE	**4.13**	**18.70**	**1.05**	**27.02**	**1.89**	-7.99	8.73	0.39	71.80	1.14
MACE (PM)	-0.30	13.29	0.63	70.84	1.36	-0.43	8.89	0.43	84.57	1.15
Benchmarks										
EW (RF)	1.88	12.60	0.71	42.73	1.42	-8.24	9.71	0.47	66.24	1.21
EW (PM)	-0.24	10.74	0.48	113.95	1.23	**-0.39**	**11.62**	**0.58**	**53.16**	**1.30**
S&P 500 (RF)	-3.53	11.98	0.70	45.57	1.40	-13.77	5.26	0.23	125.03	1.01
S&P 500 (PM)	-0.52	7.72	0.42	101.70	1.13	-0.48	9.54	0.54	78.27	1.23
Refinements										
MACE_{bag}	**4.84**	16.90	0.97	**23.43**	1.73	-7.61	9.24	0.43	65.71	1.17
$\text{MACE}_{\mu \geq \underline{\mu}}$	4.27	**19.42**	1.01	38.32	1.78	-4.04	**13.26**	**0.61**	59.30	**1.34**

Notes: The first column-wise panel consists of out-of-sample R^2's for different test (sub-)samples. The second are economic metrics, where r^A := Annualized Returns, SR := Sharpe Ratio, and DD^{MAX} = Maximum drawdown. All statistics but SR are in percentage points. Returns and risk-reward ratios are based on trading each portfolio using a simple mean-variance scheme with risk aversion parameter $\gamma = 3$. PM means the prediction is based on the respective prevailing mean with a lookback period of twenty years, while RF means using that of a Random Forest. Numbers in **bold** are the best statistic within the first two panels (that is, excluding MACE refinements). Numbers in **green** are the best statistic of whole column.

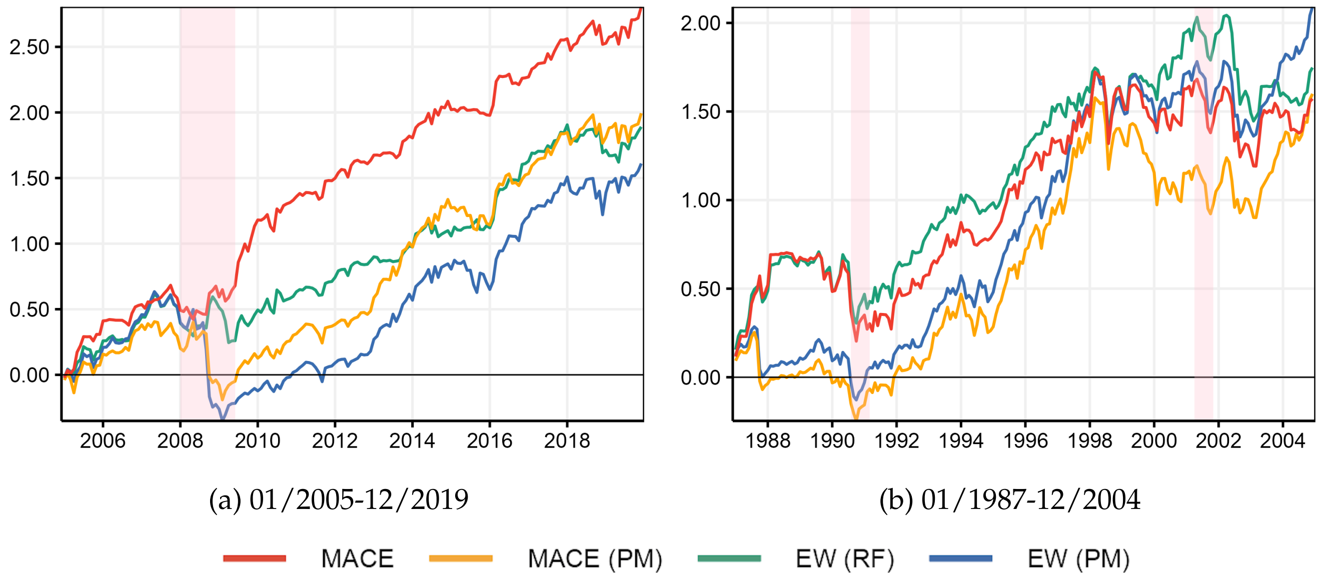
well) are not one for a fixed target, but rather for a fixed "strategy". In other words, when $y_{t'+1}$ and $y_{t''+1}$ enter MSE_{MACE} and belong to two different estimation windows, they are likely coming from two distinct time series. Since those share the same unconditional variance by construction (1 or some other standardization necessary for identification in (3)), they can be aggregated without window t' errors driving results more than that of window t'' for mechanical reasons.

When trading the MMLP, we again solve the prototypical mean-variance problem for a single return y_{t+1} as stated in Equation (6). As in Filippou et al. (2021), the risk aversion parameter γ is set to 3 and $\hat{\omega}_{t+1}$ is constrained to lie between -1 for 2 for reasonable allocations.

4.1 Results

We report relevant summary statistics for our monthly exercise in Table 5. Similar to section 3.1, we show results for MACE and its corresponding modifications/refinements as well as EW and S&P 500. We report the relevant statistics for two distinct out-of-sample periods as outlined above. Log cumulative return plots are shown in Figure 5 and R^2 comparison of MACE to Random Alternatives plots are available in Figure 6.

Economic Results. Table 5 shows that MACE is clearly outperforming any other model along each evaluation metric. The annualized average return of $r^A = 18.70\%$ beats the closest competitor (EW



(a) 01/2005-12/2019

(b) 01/1987-12/2004

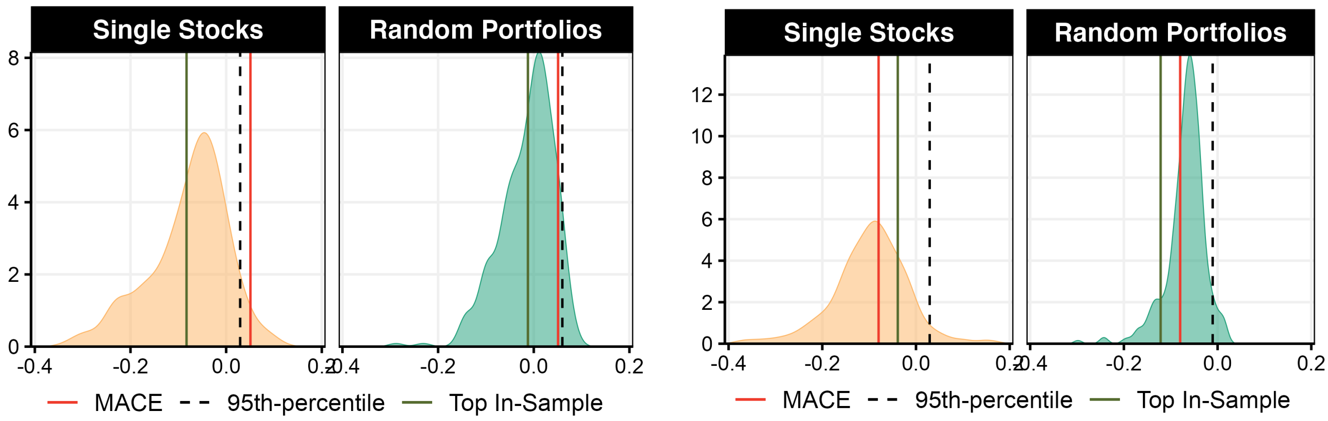
Figure 5: Cumulative Returns

(RF)) by a stunning six percentage points. Yet, this achievement does not seem to come at the cost of extreme volatility with the Sharpe Ratio and the maximum drawdown both dominating the competitors by large margins. Even though the proposed refinements might even achieve superior performance in a specific metric – MACE_{bag} gives the investor a smaller max drawdown, whereas $\text{MACE}_{\mu\geq\underline{\mu}}$ can even boost the annualized return – both modified MACEs cannot keep up with the Sharpe Ratio of $SR = 1.05$. This suggests that MACE strikes an appealing balance between risk and reward. In Appendix A.1, we document that those findings are unchanged when accounting for various levels of transaction costs.

This dominance, however, fades for the earlier period between 1987 and 2004. As outlined above, the shorter T dimension limits MACE's ability to learn a complex signal, if there is any, resulting in diminished out-of-sample performance. As discussed in section 3.3, in such a situation it may merit to tilt MACE away from pure predictability and towards a safeguard of higher unconditional return instead. This might lead to sacrificing some in-sample predictability, but prove beneficial out-of-sample. We achieve such a tilting with $\text{MACE}_{\mu\geq\underline{\mu}}$, which generates both the highest annualized average returns and scores the highest Sharpe Ratio ($SR = 0.61$).

Figure 5 gives us deeper insights into the underlying return dynamics that are cushioned by the single summary figures in Table 5. It is interesting to see how well MACE navigates the Great Recession (GR). While all competitors take either a deep hit or drift sideways (EW (RF)), MACE takes off already during the first half of GR and even accelerates its growth-rate at the outset. This behavior is different from MACE's behavior during earlier recessionary periods in the early 1990s and 2000s where MACE takes visible hits. Yet, the results for the period 2005-2019 suggest that MACE has learned from earlier mistakes.

With a R^2 of 1.88% during the 2005-2019 subperiod, EW (RF) also fares well, most notably by avoiding heavy losses during GR. However, and as it often the case with more basic approaches, predictability



(a) 01/2005-12/2019

(b) 01/1987-12/2004

Notes: This plot shows distributions of OOS-R^2's, for different portfolio sizes and subsamples. The "Single Stocks" panel reports the distribution of the N R^2's obtained from predicting each stock in the panel separately with RF. The "Random Portfolios" panel shows the distribution of 300 R^2 obtained from predicting randomly drawn portfolios with RF. *Top In-Sample* denotes the OOS-R^2 of the single stock, random portfolio respectively, that achieved the highest R^2 during training. *95th-percentile* denotes marks the 95th percentile of the corresponding distribution shown in the graphs.

Figure 6: R^2 Comparison of MACE to Random Alternatives for the Monthly Application

is heavily localized during tumultuous economic times, leading EW (RF) (and also similarly S&P 500 (RF)) to overall underperform their prevailing mean counterparts both in terms of returns and volatility.

Statistical Results. In Figure 6 we see that the share of predictable Single Stocks is very similar across the two out-of-sample periods, whereas the distribution for Random Portfolios clearly shifts to the right during the 2005-2019 era. Thus, there *is* exploitable predictability, and MACE's job is to find promising w ex-ante. We see that it does so: it is superior to about 95% of randomly drawn portfolios. In the first subperiod, where more than 95% of such portfolios deliver negative R^2, MACE suffers a setback along with the crowd. With only ~5% of portfolios found to be predictable ex-post, it is quite a daunting task to land in the promising region ex-ante. As was also observed in daily results, MACE appears particularly apt at finding the MPP when there is a reasonable number of possibly successful candidates to work with. In the opposite scenario where no or very few RFs attain any predictability, MACE inevitably struggles.

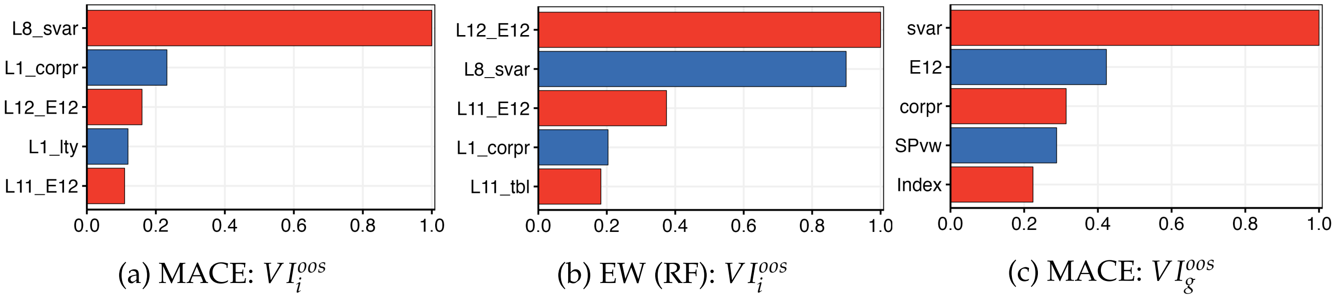
Note that the large mass to the right of 0 in Figure 6a is not necessarily indicative of market inefficiency given that one still needs to find the relevant vectors ex-ante. It, however, highlights a non-trivial number of possibilities for it to occur. On the other hand, the absence of a mass on the right side of 0 (as in Figure 6b) is suggestive of efficiency, conditional on choices for stocks, information set, and predictive function.

4.2 Predictability through the (Nonlinear) Time-Varying Risk Premia?

Figure 5 makes clear that MACE's stellar performance results predominantly from not tanking during the Great Recession and climbing steeply at its outset. This pattern is different from the other models.

While EW (RF) did neither tank during the crisis, its engine sputtered post-crisis. In contrast, the MACE (PM)'s and EW (PM)'s growth picked up relatively quickly in the aftermath of the Great Recession, however, only after having tanked deeply throughout the crisis period. Only MACE seems to get the market timing right for both the crisis period and the subsequent recovery.

To shed light on which economic indicators are driving this success, we use Shapley Values, a well established and evermore popular tool to quantify the contribution of predictors in opaque models (Shapley, 1953; Lundberg and Lee, 2017). We refer the reader to Molnar (2019) for a generic textbook treatment and Borup et al. (2022) for a focus on its applicability to financial and macroeconomic forecasting. Here, we dedicate our attention to the *out-of-sample* period 01/2008 - 12/2009. Relevant details regarding the construction of our variable importance metric from expanding windows are relegated to Appendix A.2.



(a) MACE: VI_i^{oos} (b) EW (RF): VI_i^{oos} (c) MACE: VI_g^{oos}

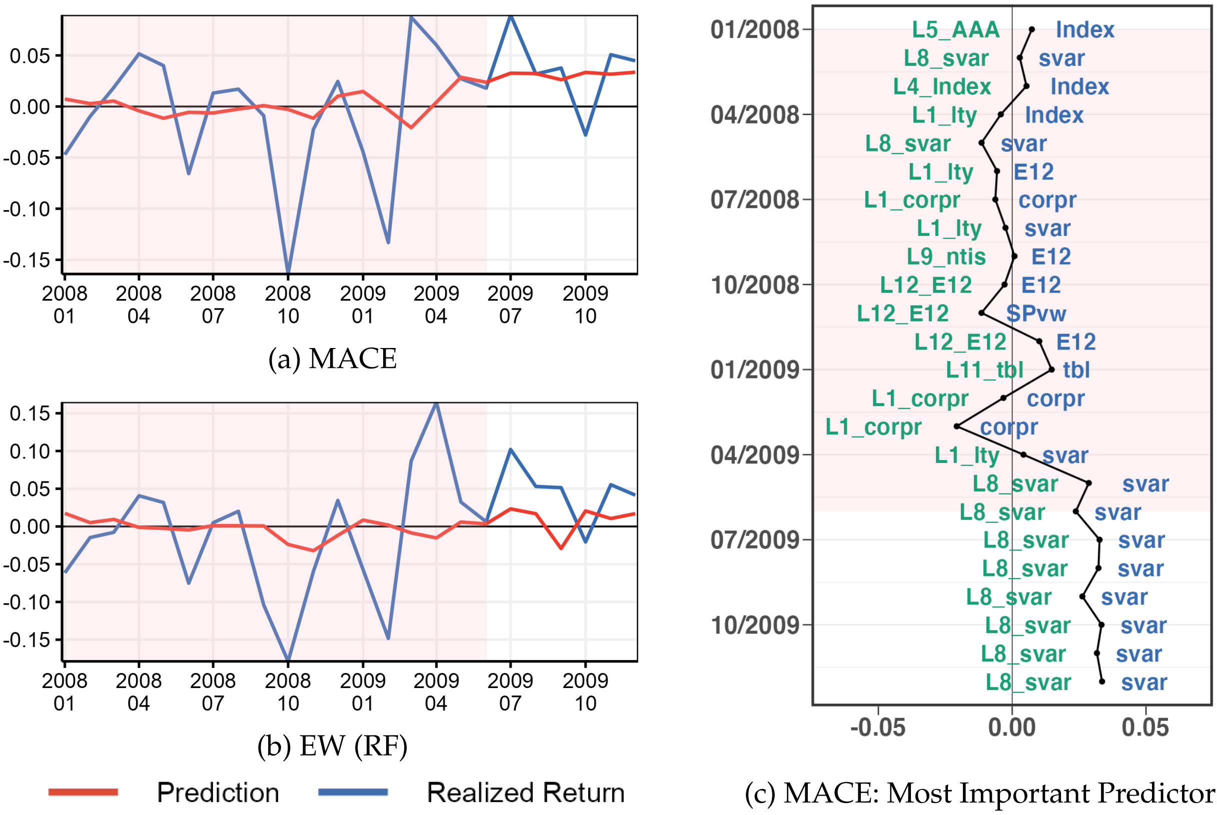
Notes: The bars represent the VI of predictor i and grouped versions (VI_g, i.e. , summing the Shapley Values across all lags of variable i) as in Equation A.2, scaled by the corresponding maximum value.

Figure 7: Shapley Value Importance: 01/2008 - 12/2009

Figure 7 reports the five most important predictors for MACE and EW (RF) separately. The last panel combines the importance of all the lags of a given indicator. For MACE, the picture is dominated by a single strong predictor: the eight-month lagged stock market volatility (`L8_svar`).[9] Grouping all lags together in Figure 7c reinforces the case for the overall importance of `svar` itself. This indicates that MACE may leverage a subtle form of time-varying risk premium, as originally formalized in the ARCH-M model of Engle et al. (1987), its extension with time-varying parameters (TVP ARCH-M, Chou et al. 1992), or the GARCH-in-mean of French et al. (1987). These models allow for an asset's volatility to directly feed into the conditional mean of the asset's return, allowing for time-varying risk premia. "Subtle" refers here to the pattern being less evident than what one would expect from these classic models. The reason for this is threefold: first, there is a significant delay. Second, the volatility metric undergoes a highly non-linear transformation. Third, it is not MACE's previous volatility that enters the conditional mean, but that of the overall market as proxied by the S&P 500.

When it comes to EW (RF), a few differences stand out: the contributions are more evenly distributed, with `E12` coming in first, followed by stock market volatility. The other features are mostly shared with MACE's prediction. Yet, EW (RF) predictions partly go awry post 2008, and MACE's peculiar use of

[9] Of course, the predictor itself had high variance during this period. In Appendix A.3, we address this concern and show that `L8_svar` stands the test of adjusting the Shapley Values for the indicator's volatility.



(a) MACE

(b) EW (RF)

(c) MACE: Most Important Predictors

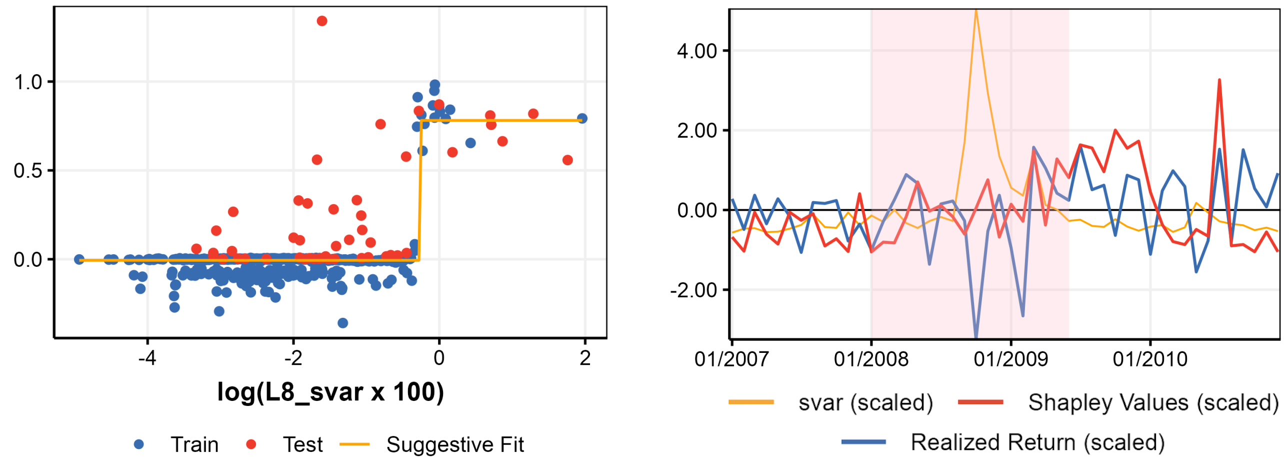
Notes Panels (a) and (b): we plot the realized return of the MACE and an equally weighted portfolio in blue. In red, we plot the corresponding predicted return of MACE and EW (RF).
Notes Panel (c): the solid line shows again MACE's prediction (see the solid blue line in panel (a)). To the left of the solid line, we show in green the most important single predictor in each month *t* based on a Shapley Value decomposition. To the right of the solid line, we show in blue the most important grouped predictor in each month *t*.

Figure 8: MACE and EW (RF) during the Great Recession

svar is the most plausible explanation for it avoiding this predicament. This is quite visible in Figure 8a and 8b, where we show the predictions of MACE and EW (RF) compared to the corresponding realized return. MACE's portfolio has a slightly less volatile return starting from 2009 and the associated predictions lie confidently in positive territory. EW's realized return has higher highs and lower lows and predictions are much more timid, that is, they are much closer to their unconditional mean.

In Figure 8c, we plot again the prediction for MACE and in each month, we report the single most important feature to its left-hand side. On the right, it is the single most important group (of lags).[10] It is striking that the string of positive predictions at the outset of the Great Recession are all attributable to svar, and in particular, its 8th lag. Yet, the prior evidence on the relevance of svar is mixed. In applications with sample periods ending prior to the Great Recession, Guo (2006) gets positive results while Welch and Goyal (2007) and Rapach et al. (2010) get negative ones. Using data through 2013, Lima and Meng (2017) find svar gaining forecasting power for S&P 500 excess returns post-1985. MACE differs by being nonlinear and not looking at a pre-specified index. However, nonlinearity by itself

[10] See Appendix A.2 for further details about the calculation. For the grouped version, we sum the absolute Shapley Values at a given point in time across all lags of a particular feature *i*. This plotting scheme is inspired by Borup et al. (2022).



(a) Shapley Values: L8_svar

(b) MACE and svar over time

Notes Panel (a): we plot the Shapley Values for the 8th lag of svar against the observed stock-market volatility, lagged by 8 months (L8_svar). The points in blue represent the local Shapley Values for the in-sample period of the expanding-window ending in 12/2006. The points in red depict the local out-of-sample Shapley Values of our expanding-window exercise. The orange line is fitted to the joint distribution of in- and out-of-sample points.
Notes Panel (b): in blue, we show the *scaled* realized return of the MACE portfolio from a buy-&-hold strategy. The red line shows the *scaled* localized Shapley Values of the grouped svar on the left, and of the 8th lag of svar on the right.

Figure 9: MACE & Stock-Market Volatility (svar) around the Great Recession
01/2007 - 12/2010

appears to be insufficient as per EW (RF) not leveraging svar in any meaningful way.

Lastly, we investigate *how* L8_svar appears to contribute. Figure 9a shows a scatter plot of the Shapley Values for L8_svar (the "local contributions") with $\log\left(\texttt{L8\_svar} \times 100\right)$ on the x-axis.[11] The yellow line represents a suggestive fit as the mean of all point realizations of $\log\left(\texttt{L8\_svar} \times 100\right) < -0.25$ and the mean of all realizations of $\log\left(\texttt{L8\_svar} \times 100\right) \geq -0.25$. While being inherently imperfect because of L8_svar's various interactions with other predictors in RF, this fit is nonetheless instructive. First, the sign is right: more risk commands a higher premium. At a level of around $\texttt{L8\_svar} \approx 0.0078$, which translates into a measure of daily stock-market volatility of $\sigma = \sqrt{\texttt{L8\_svar}} \approx 0.088 \equiv 8.8\%$, market uncertainty seemingly triggers a regime of higher expected rate of return on the MACE portfolio. This observation speaks to the findings in Campbell and Hentschel (1992) that the volatility feedback channel emerges during times of elevated volatility. Here, the suggestive fit points to a simple two regimes relationship: a first one where there is basically no risk premium, and second where there is a constantly higher premium, irrespective of the specific values of L8_svar as long as it is above a certain threshold.

Figure 9b shows how this nonlinear relationship plays out in the time space. We plot the *scaled* versions of the realized stock variance (svar), the local Shapley Values for the grouped svar, and the realized MACE returns. The positive relationship between svar and the realized MACE returns is clearly emerging from the midst of the Great Recession in late 2008 onwards. The delay is visible from the red

[11] The dots in blue represent the Shapley Values over the training-set of the expanding window with an OOS start date in 01/2007. The red dots are the OOS Shapley Values collected for the expanding window 01/2007-12/2010. See Appendix A.2 for a detailed description of the collection process.

bump appearing much later than the original `svar` impulse. The nonlinearity is also discernible from the red line following a very different pattern than the orange one – perfect linearity would imply a mere rightward translation of the orange line. The red plateau is well timed with high (unconditional) MACE realized returns at the outset of the Great Recession. From that and other observations, we can conclude that part of MACE's success in that era is uncovering a portfolio with a well dissimulated, yet stronger reaction to changes in volatility regimes.

5 Concluding Remarks

We introduce the MACE algorithm to construct maximally machine-learnable portfolios. It does so by directly optimizing the portfolio weights to make life easier for the prediction function. As we have discussed, this does not neglect variance, quite to the contrary, as the MMLP problem is intimately linked to traditional mean-variance optimization. Advantages with respect to the various strands of literature building *linear* mean-reverting portfolio is MACE's flexibility through the use of Random Forest and its scalability. Peaking into the future, those qualities are essential to discover increasingly complex patterns of predictability in an era where a flock of humans and machines are constantly on the lookout for those. With respect to key ML applications in empirical asset pricing, MACE provides a low maintenance (data- and computations-wise) alternative which can deliver the goods leveraging only basic time series data, or lagged returns themselves. Our two applications, daily and monthly trading, illustrate that by scoring enviable returns and Sharpe Ratios in evaluation periods where gains from using ML methods have often been anticlimactic.

There are quite a few directions for future research, beyond more or less straightforward applications to new assets and information sets, and changing ridge regularization for any other shrinkage one's heart desires. First, MACE could be extended to solely learn buy-sell signals where the cutoff point itself is trainable within the loop. In that way, we could potentially construct "episodic portfolios" where trading rarely occurs and typically does so when a rarer event is expected with moderate uncertainty. Second, some structured form of nonlinearities could be accommodated on the left-hand side of the equation. While from a statistical standpoint, nothing is impossible, from a financial one, the LHS must remain a tradeable combination of securities. Nonetheless, some nonlinear transformations of returns can be approximated by appropriately designed options and MACE could learn a maximally predictable combination of financial instruments. Third and more ambitiously, MACE's alternating EM-style algorithm could potentially be replaced by a single hemisphere neural network (à la Goulet Coulombe (2022)) that minimizes directly the MMLP loss function combined with bagging strategies to deal with the inevitability of overfitting and finding a trivial solution. As discussed earlier, there are numerous headwinds to such modifications and bagging by itself may not be enough. But, keeping in mind deep learning's edge with large and non-traditional data, the additional efforts could perhaps bring MMLPs to new highs.

References

Andrew, G., Arora, R., Bilmes, J., and Livescu, K. (2013). Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR.

Athey, S., Tibshirani, J., and Wager, S. (2019). Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178.

Babiak, M. and Baruník, J. (2020). Deep learning, predictability, and optimal portfolio returns. *arXiv preprint arXiv:2009.03394*.

Balder, S. and Schweizer, N. (2017). Risk aversion vs. the omega ratio: Consistency results. *Finance Research Letters*, 21:78–84.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. (2020). Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Bergmeir, C., Hyndman, R. J., and Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120:70–83.

Borup, D., Coulombe, P. G., Rapach, D., Schütte, E. C. M., and Schwenk-Nebbe, S. (2022). The anatomy of out-of-sample forecasting accuracy.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L. and Friedman, J. H. (1985). Estimating Optimal Transformations for Multiple Regression and Correlation. *Journal of the American Statistical Association*, 80(391):580–598. Available at https://www.tandfonline.com/doi/abs/10.1080/01621459.1985.10478157.

Campbell, J. Y. and Hentschel, L. (1992). No News is Good News: An Asymmetric Model of Changing Volatility in Stock Returns. *Journal of Financial Economics*, 31(3):281–318.

Campbell, J. Y. and Thompson, S. B. (2008). Predicting excess stock returns out of sample: Can anything beat the historical average? *The Review of Financial Studies*, 21(4):1509–1531.

Carrasco, M. and Noumon, N. (2011). Optimal portfolio selection using regularization. *Citeseer, Tech. Rep.*

Chen, L., Pelger, M., and Zhu, J. (2021a). Deep Learning in Asset Pricing. Available at: https://ssrn.com/abstract=3350138.

Chen, Q., Syrgkanis, V., and Austern, M. (2022). Debiased machine learning without sample-splitting for stable estimators. *arXiv preprint arXiv:2206.01825*.

Chen, W., Zhang, H., Mehlawat, M. K., and Jia, L. (2021b). Mean–Variance Portfolio Optimization Using Machine Learning-Based Stock Price Prediction. *Applied Soft Computing*, 100:106943.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters.

Chordia, T., Subrahmanyam, A., and Tong, Q. (2014). Have capital market anomalies attenuated in the recent era of high liquidity and trading activity? *Journal of Accounting and Economics*, 58(1):41–58.

Chou, R., Engle, R. F., and Kane, A. (1992). Measuring Risk Aversion from Excess Returns on a Stock Index. *Journal of Econometrics*, 52(1):201–224.

Cong, L., Tang, K., Wang, J., and Zhang, Y. (2021). AlphaPortfolio: Direct Construction Through Deep Reinforcement Learning and Interpretable AI. Available at SSRN: https://ssrn.com/abstract=3554486.

Cuturi, M. and d'Aspremont, A. (2013). Mean reversion with a variance threshold. In *International Conference on Machine Learning*, pages 271–279. PMLR.

d'Aspremont, A. (2011). Identifying small mean-reverting portfolios. *Quantitative Finance*, 11(3):351–364.

De Nard, G., Engle, R. F., and Kelly, B. T. (2017). Factor Mimicking Portfolios for Climate Risk. *University of Zurich, Department of Economics, Working Paper*, (429). Available at SSRN: https://ssrn.com/abstract=4388326.

Elsayed, S., Thyssens, D., Rashed, A., Jomaa, H. S., and Schmidt-Thieme, L. (2021). Do we really need deep learning models for time series forecasting? *arXiv preprint arXiv:2101.02118*.

Engle, R. F., Lilien, D. M., and Robins, R. P. (1987). Estimating Time Varying Risk Premia in the Term Structure: The Arch-M Model. *Econometrica*, 55(2):391–407.

Fallahgoul, H., Franstianto, V., and Lin, X. (2020). Asset pricing with neural networks: Significance tests. *Available at SSRN 3889777*.

Farmer, L., Schmidt, L., and Timmermann, A. (2022). Pockets of Predictability. *Journal of Finance, forthcoming*.

Filippou, I., Rapach, D., Taylor, M. P., and Zhou, G. (2022). Out-of-sample exchange rate prediction: A machine learning perspective. *Available at SSRN 3455713*.

Filippou, I., Rapach, D. E., and Thimsen, C. (2021). Cryptocurrency Return Predictability: A Machine-Learning Analysis. *Working Paper*. Available at https://drive.google.com/file/d/1jyysOL1qDqA8uPnkGqy_m10aQWM-EFu4/view.

Firoozye, N., Tan, V., and Zohren, S. (2022). Canonical portfolios: Optimal asset and signal combination. *arXiv preprint arXiv:2202.10817*.

Fogarasi, N. and Levendovszky, J. (2013). Sparse, mean reverting portfolio selection using simulated annealing. *Algorithmic Finance*, 2(3-4):197–211.

French, K. R., Schwert, G. W., and Stambaugh, R. F. (1987). Expected stock returns and volatility. *Journal of financial Economics*, 19(1):3–29.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The Elements of Statistical Learning*, volume 1. Springer series in statistics New York, NY, USA:.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

Gopinathan, R. and Durai, S. (2019). Stock market and macroeconomic variables: new evidence from india. *Financial Innovation*, 5(1):1–17.

Gotoh, J.-y. and Fujisawa, K. (2014). Convex optimization approaches to maximally predictable portfolio selection. *Optimization*, 63(11):1713–1735.

Goulet Coulombe, P. (2020a). The macroeconomy as a random forest. *arXiv preprint arXiv:2006.12724*.

Goulet Coulombe, P. (2020b). To bag is to prune. *arXiv preprint arXiv:2008.07063*.

Goulet Coulombe, P. (2022). A neural phillips curve and a deep output gap. *Available at SSRN*.

Goulet Coulombe, P., Leroux, M., Stevanovic, D., and Surprenant, S. (2021). Macroeconomic data transformations matter. *International Journal of Forecasting*, 37(4):1338–1354.

Goulet Coulombe, P., Leroux, M., Stevanovic, D., and Surprenant, S. (2022). How is Machine Learning Useful for Macroeconomic Forecasting? *Journal of Applied Econometrics, forthcoming*.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on tabular data? *arXiv preprint arXiv:2207.08815*.

Griveau-Billion, T. and Calderhead, B. (2021). Efficient computation of mean reverting portfolios using cyclical coordinate descent. *Quantitative Finance*, 21(4):673–684.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies*, 33(5):2223–2273.

Guo, H. (2006). On the Out-of-Sample Predictability of Stock Market Returns. *The Journal of Business*, 79(2):645–670.

Haase, F. and Neuenkirch, M. (2022). Predictability of bull and bear markets: A new look at forecasting stock market regimes (and returns) in the us. *International Journal of Forecasting*.

Han, Y., He, A., Rapach, D., and Zhou, G. (2018). What firm characteristics drive us stock returns. *Available at SSRN*, 3185335.

Harris, R. D., Shen, J., and Yilmaz, F. (2022). Maximally predictable currency portfolios. *Journal of International Money and Finance*, 128:102702.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. (2019). Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*.

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Keating, C. and Shadwick, W. F. (2002). A universal performance measure. *Journal of performance measurement*, 6(3):59–84.

Konno, H., Morita, Y., and Yamamoto, R. (2010a). A maximal predictability portfolio using absolute deviation reformulation. *Computational Management Science*, 7(47).

Konno, H., Takaya, Y., and Yamamoto, R. (2010b). A maximal predictability portfolio using dynamic factor selection strategy. *International Journal of Theoretical and Applied Finance*, 13(3):355–366.

Krauss, C. (2017). Statistical arbitrage pairs trading strategies: review and outlook. *Journal of Economic Surveys*, 31(2):513–545.

Krauss, C., Do, X. A., and Huck, N. (2017). Deep Neural Networks, Gradient-Boosted Trees, Random Forests: Statistical Arbitrage on the S&P 500. *European Journal of Operational Research*, 259(2):689–702.

Kreiss, J.-P. and Lahiri, S. N. (2012). Bootstrap methods for time series. In *Handbook of statistics*, volume 30, pages 3–26. Elsevier.

Lalwani, V. and Meshram, V. V. (2020). Stock Market Efficiency in the Time of COVID-19: Evidence from Industry Stock Returns. *International Journal of Accounting & Finance Review*, 5(2):40–44.

Ledoit, O. and Wolf, M. (2004). Honey, i shrunk the sample covariance matrix. *The Journal of Portfolio Management*, 30(4):110–119.

Li, X. and Zakamulin, V. (2020). Stock Volatility Predictability in Bull and Bear Markets. *Quantitative Finance*, 20(7):1149–1167.

Lima, L. R. and Meng, F. (2017). Out-Of-Sample Return Predictability: A Quantile Combination Approach. *Journal of Applied Econometrics*, 32(4):877–895.

Lo, A. W. and MacKinlay, A. C. (1997). Maximizing predictability in the stock and bond markets. *Macroeconomic dynamics*, 1(1):102–134.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774.

Lütkepohl, H. (2011). Forecasting aggregated time series variables: A survey. *OECD Journal: Journal of Business Cycle Measurement and Analysis*, 2010(2):1–26.

Makur, A., Kozynski, F., Huang, S.-L., and Zheng, L. (2015). An efficient algorithm for information decomposition and extraction. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 972–979. IEEE.

Medhat, M. and Schmeling, M. (2022). Short-term momentum. *The Review of Financial Studies*, 35(3):1480–1526.

Michaeli, T., Wang, W., and Livescu, K. (2016). Nonparametric canonical correlation analysis. In *International conference on machine learning*, pages 1967–1976. PMLR.

Molnar, C. (2019). *Interpretable machine learning*. Lulu.com.

Nagel, S. (2021). *Machine Learning in Asset Pricing*, volume 8. Princeton University Press.

Olson, M. A. and Wyner, A. J. (2018). Making sense of random forest probabilities: a kernel perspective. *arXiv preprint arXiv:1812.05792*.

Rao, J., Ji, C., Wen, J., Wang, J., and Sun, W. (2022). Nonstationary process monitoring based on alternating conditional expectation and cointegration analysis. *Processes*, 10(10):2003.

Rapach, D. E., Strauss, J. K., and Zhou, G. (2010). Out-of-Sample Equity Premium Prediction: Combination Forecasts and Links to the Real Economy. *The Review of Financial Studies*, 23(2):821–862.

Shapley, L. S. (1953). A Value for N-Person Games. *Contributions to the Theory of Games*, 2(28):307–317.

Ta, B. Q., Huynh, V. T., Nguyen, K. Q. H., Nguyen, P. N., and Ho, B. H. (2022). Maximal predictability portfolio optimization model and applications to vietnam stock market. In Sriboonchitta, S., Kreinovich, V., and Yamaka, W., editors, *Credible Asset Allocation, Optimal Transport Methods, and Related Topics*, pages 559–578. Springer International Publishing.

Taddy, M., Chen, C.-S., Yu, J., and Wyle, M. (2015). Bayesian and empirical bayesian forests. *arXiv preprint arXiv:1502.02312*.

Takaya, Y. and Konno, H. (2010). A maximal predictability portfolio subject to a turnover constraint. *Asia-Pacific Journal of Operational Research*, 27(1):1–13.

Welch, I. and Goyal, A. (2007). A Comprehensive Look at The Empirical Performance of Equity Premium Prediction. *The Review of Financial Studies*, 21(4):1455–1508.

Yamamoto, R., Ishii, D., and Konno, H. (2007). A maximal predictability portfolio model: Algorithm and performance evaluation. *International Journal of Theoretical and Applied Finance*, 10(6):1095–1109.

Zhao, Z. and Palomar, D. P. (2016). Mean-reverting portfolio design via majorization-minimization method. In *2016 50th Asilomar Conference on Signals, Systems and Computers*, pages 1530–1534.

Zhao, Z. and Palomar, D. P. (2018). Mean-reverting portfolio with budget constraint. *IEEE Transactions on Signal Processing*, 66(9):2342–2357.

A Appendix

A.1 Transaction Costs

In the section, we quantifying the reduction in economic performance due to transactions costs (TC). To do so, we calculate

$$TC_t = \mathfrak{c} \times \sum_{n=1}^{N} \left| r_{n,t} \left(\omega_t w_{n,t} - \omega_{t-1} w_{n,t-1} \right) \right| , \tag{A.1}$$

where $r_{n,t}$ is the return of stock n in period t, $w_{n,t}$ is the portfolio-weight of stock n at time t, ω_t is the trader's positioning at time t and $\mathfrak{c}$ is the share of the absolute return after trading that is to be allotted to transaction costs (Harris et al., 2022).

Daily Results. Given that our strategy utilizes highly liquid stocks listed on the NASDAQ, we expect transaction costs (TC) to be low. Thus, we set $\mathfrak{c} \in \{0.1\%, 0.5\%, 1\%\}$ The lowest $\mathfrak{c}$ is recommended by De Nard et al. (2017) for trading Volatility Lab's 177 sustainable funds that are very large and liquid.[12]

Table 6: Daily Stock Returns after Transaction Costs

	0.1%			0.5%			1.0%		
	r^A	SR	Ω	r^A	SR	Ω	r^A	SR	Ω
MACE$_{20}$	22.90	0.98	1.18	21.88	0.93	1.17	20.59	0.88	1.15
MACE$_{\text{loose bag}}$	22.91	1.35	1.25	22.18	1.31	1.24	21.27	1.26	1.22
MACE$_{100}$	40.49	1.56	1.32	36.94	1.43	1.28	32.49	1.27	1.24

Notes: This table reports annualized returns (r^A), Sharpe Ratio (SR) and the Omega Ratio (Ω) for various MACE portfolios after accounting for transaction costs with $\mathfrak{c} \in \{0.1\%, 0.5\%, 1\%\}$.

Table 6 reports the corresponding summary statistics for various MACE portfolios after subtracting TC_t from the realized returns r_t. For the two portfolios with $N = 20$ stocks, annualized returns before TC amounted to 23.16% for MACE$_{20}$ (Table 2 and to 23.10% for MACE$_{\text{loose bag}}$ (Table 4). As Table 6 shows, the fallout due to transaction costs is well contained for these portfolios. Evidently, the degradation has to be higher for MACE$_{100}$ since it implies trading five times more stocks. It peaks at a reduction of about 10% when assuming the most pessimistic $\mathfrak{c}$. The other reductions are smaller by construction, and in all cases, MACE$_{100}$ still dominates alternatives in terms of the three economic metrics. Note that for all three MACEs, TC-adjusted r^A, SR, and Ω are still unquestionably well above what is reported for competing strategies , including passive ones (with TCs ≈ 0) and more proactive ones.

Monthly Results. Table 7 shows the monthly returns for MACE and its refinements after accounting for several levels of transaction costs. We now use $\mathfrak{c} \in \{0.5\%, 1\%, 2\%\}$ which is highly conservative, and accommodates for the fact that the out-of-sample covers about 3 decades. It is also a more precautions choice given that, unlike the daily applications, considered stocks are certainly large-caps, but not

[12] See vlab.stern.nyu.edu.

necessarily the largest caps in an era of lessened TCs. Overall, we see that TCs only eat up a minor fraction of average monthly returns, such that also the corresponding risk-metrics, SR and Ω remain in the neighborhood of those reported in Table 5.

Table 7: Monthly Stock Returns after Transaction Costs

	0.5%			1.0%			2.0%		
	r^A	SR	Ω	r^A	SR	Ω	r^A	SR	Ω
				01/2005 - 12/2019					
MACE	18.54	1.04	1.88	18.39	1.03	1.87	18.07	1.02	1.84
MACE$_{\text{bag}}$	16.76	0.96	1.72	16.63	0.96	1.71	16.35	0.94	1.69
MACE$_{\mu \geq \underline{\mu}}$	19.33	1.00	1.77	19.23	1.00	1.76	19.04	0.99	1.75
				01/1987 - 12/2004					
MACE	8.54	0.38	1.13	8.35	0.37	1.13	7.98	0.36	1.11
MACE$_{\text{bag}}$	9.11	0.43	1.16	8.97	0.42	1.16	8.71	0.41	1.15
MACE$_{\mu \geq \underline{\mu}}$	13.15	0.61	1.33	13.05	0.60	1.33	12.84	0.59	1.32

Notes: This table reports annualized returns (r^A), Sharpe Ratio (SR) and the Omega Ratio (Ω) for various MACE portfolios after accounting for transaction costs with $\mathfrak{c} \in \{0.5\%, 1.0\%, 2.0\%\}$. See Equation (A.1) for the exact calculation.

A.2 Variable Importance Calculations

In our monthly expanding window exercise, both periods are obviously not static but "evolving". With $e = 1, ..., E$ expanding windows, T_e^{ins} denotes the end of the in-sample period for window e. Hence, we collect the corresponding Shapley Values for the OOS period as follows: for each variable i, we collect only those Shapley Values that fall into the interval starting with the month following the end of the current window's in-sample period ($T_e^{ins} + 1$) and ending with the end of the in-sample period of the next expanding window (T_{e+1}^{ins}). As we expand our in-sample period each quarter by another three months, the period between $T_e^{ins} + 1$ and T_{e+1}^{ins} amounts to three months. The corresponding OOS variable importance of variable i (VI_i^{oos}) is thus calculated as follows:

$$VI_i^{oos} = \sum_{e=1}^{E} \sum_{t=T_e^{ins}+1}^{T_{e+1}^{ins}} \mid \phi_{i,t} \mid \quad . \tag{A.2}$$

Taking Figure 9 as an example, where the OOS period runs from 01/2007 through 12/2010: we start in 12/2006 and collect the first three local Shapley Values of the OOS-period (01/2007-03/2007). We then expand our training set until 03/2007. Hence we collect the first three local Shapley Values of the new OOS period (04/2007-06/2007). We proceed until our training set ends in 09/2010.

Summarizing indicator i's contribution across all it's lags, we calculate the *grouped* VI for group g (VI_g^{oos}) as follows:

$$VI_g^{oos} = \sum_{e=1}^{E} \sum_{t=T_e^{ins}+1}^{T_{e+1}^{ins}} \sum_{i \in g} |\, \phi_{i,t} \,| \quad . \tag{A.3}$$

where g includes all lags of with which indicator i is represented in the feature set.

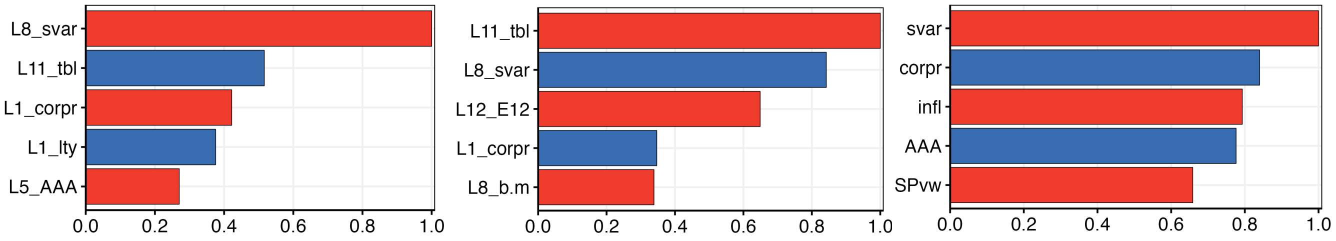
A.3 Volatility-Adjusted VI-Plots

In Figure 10 we show volatility-adjusted VI plots. That is, we adjust VI_i^{oos} in Equation (A.2) by indicator i's ratio of OOS to in-sample standard deviation:

$$AdjVI_z^{oos} = VI_z^{oos} \times \left(\frac{\sigma_z^{oos}}{\sigma_z^{ins}} \right)^{-1} \qquad \text{for } z = i, g \quad , \tag{A.4}$$

where σ_i^{oos} is the standard deviation over the OOS period (here: 01/2008 - 12/2009) and σ_i^{ins} the standard deviation over the in-sample period (here: 03/1957 - 12/2007) respectively.

For the grouped case ($AdjVI_g^{oos}$), the standard deviation (σ_g^{oos}) is calculated as the standard deviation of the moving-average of indicator i, where the length of the moving average corresponds to the number of lags (here 12) with which i enters the predictor matrix.



(a) MACE: $AdjVI_i^{oos}$ (b) EW (RF): $AdjVI_i^{oos}$ (c) MACE: $AdjVI_g^{oos}$

Notes: The bars represent the volatility-adjusted VI of predictor i and grouped versions ($AdjVI_{\text{grouped}}$, i.e. , summing the Shapley Values across all lags of variable i) as in Equation A.4. We scale indicator i's Shapley Values by the ratio of i's out-of-sample and in-sample standard deviation. The in-sample period runs from 03/1958 to 12/2007. In the *grouped* version, we calculate the standard deviation of the 12-month moving average, as each indicator i enters the predictor matrix with 12 lags. Afterwards, the Shapley Values are further scaled by the maximum Shapley Value.

Figure 10: Volatility-Adjusted Shapley Value Importance: 01/2008 - 12/2009