INVESTMENT

REPLICA

Giorgia Nicoletti
Francesco Panichi
Filippo Puricelli
Fabrizio Raeli
Fiamma Ruscito

HEDGE FUNDS

Hedge funds are investment funds that employ a range of strategies to achieve high returns for their investors. These strategies can include leveraging, short selling and using derivatives, allowing hedge funds to profit in both bullish and bearish markets.

INVESTMENT REPLICA

- **Cost efficiency**
- **Transparency:** futures are standardized contracts traded on exchanges

PROS

- **High Potential Returns**: aggressive strategies which can outperform traditional investments
- **Diversification:** access to wide range of asset classes

CONS

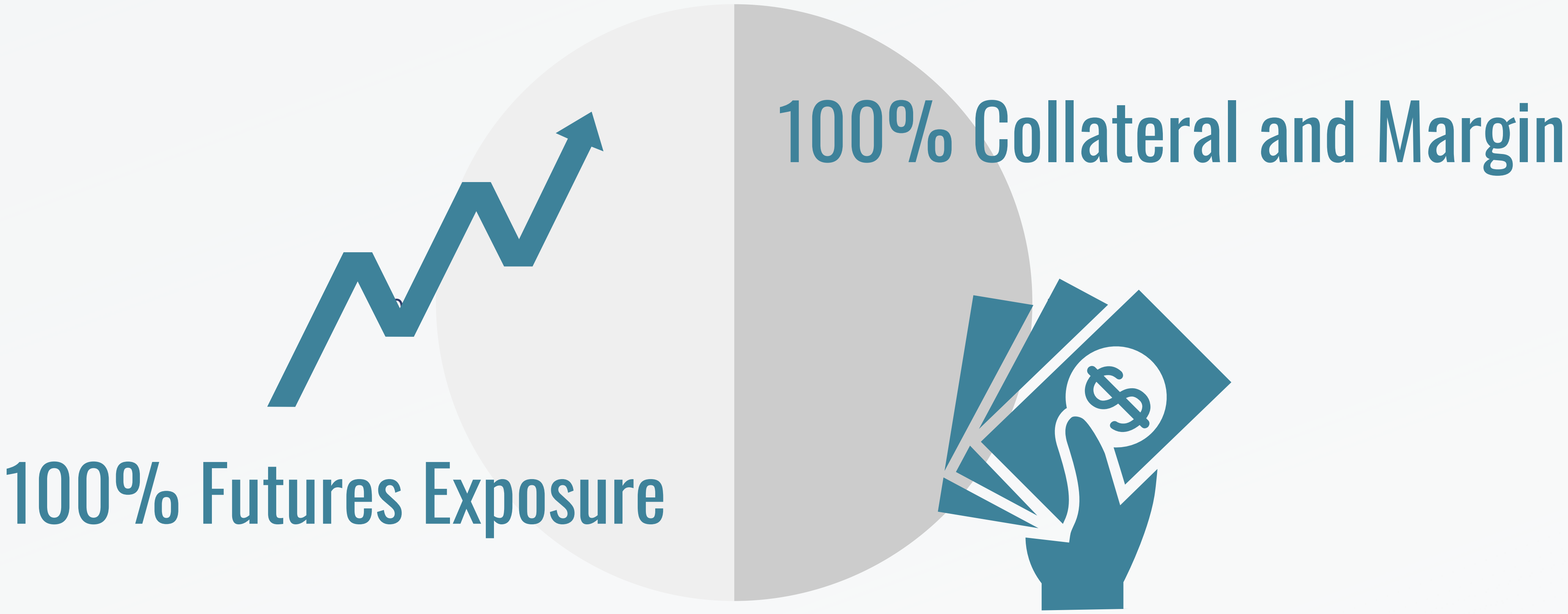
- **High Fees**: a management fee (usually 2% of assets) and a performance fee (often 20% of profits)
- **Lack of Transparency:** unknown portfolio composition

FINANCIAL LEVERAGE

Max Leverage 200%: Futures and Collateral Allocation

In a futures portfolio, leverage allows an investor to control a larger position with a relatively small amount of capital

The total allocation to futures contracts is capped at 100% of the portfolio's value



DATA ANALYSIS

Correlation Analysis and Data Set Optimization

ADDING SPREADS

- 2-year and 10-year Germany bonds
- 2-year and 10-year US Treasuries
- Tech Equity and US Equity

Removed due to low correlation:

- 2-year Germany government bond
- Gold
- 10-year Germany government bond
- 10-year US Treasury
- German spread
- US spread

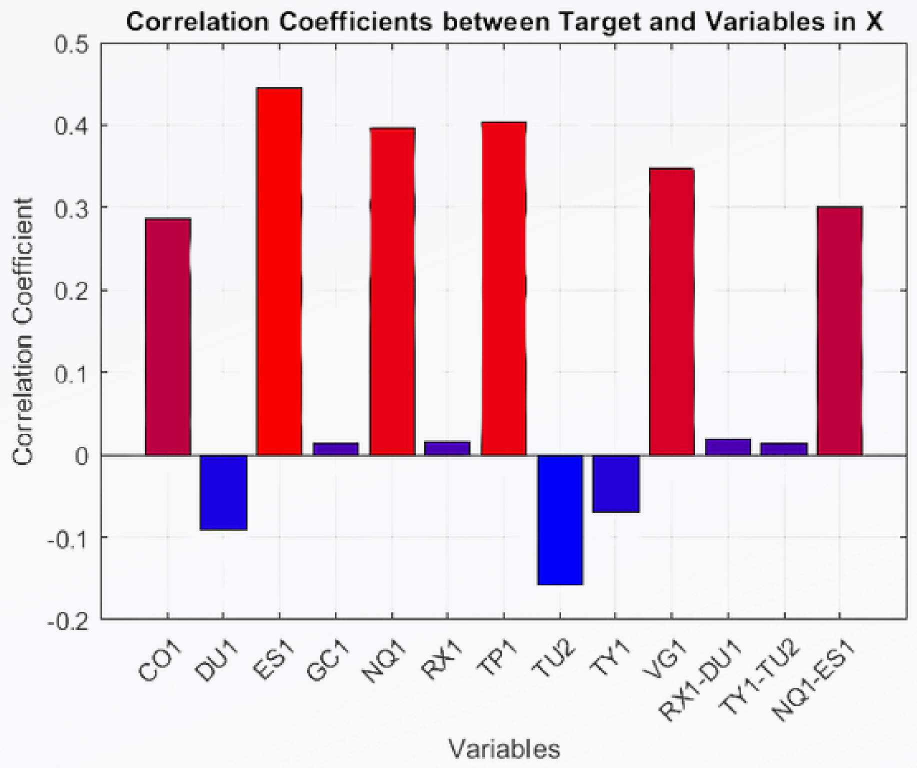
CONSIDERATIONS

Excluded data from the financial crisis to avoid biased results

5 JAN 2010

The stability and predictability of bonds and gold make them less attractive for hedge funds that seek higher returns through dynamic, high-volatility assets and complex strategies

CORRELATION ANALYSIS



Selected Futures

CO1 = Brent (Oil)
ES1 = S&P 500 (US Equity)
NQ1 = Nasdaq 100 (Tech Equity)
TP1 = Topix (Jap. Equity)
TU2 = 2Yrs US Treasury (US Gvt)
VG1 = Eurostoxx 50 (EU Equity)
NQ1-ES1 spread

Normalized as:

$$X_{norm} = \frac{X - \mu_{train}}{\sigma_{train}}$$

LASSO REGRESSION

We used the **Elastic Net** approach implemented with the lasso command in MATLAB. The **alpha** coefficient in Elastic Net determines the balance between lasso (L1) and ridge (L2) regression. An alpha close to 1 makes the model behave more like lasso, resulting in more zero coefficients, while an alpha close to 0 makes it behave more like ridge regression.

In our case, alpha variation had **minimal impact** on the results, so we chose an **alpha** of **0.85** to favor lasso and increase the number of zero weights

Loss Function

$$\min_{\beta_0,\beta}\left(\frac{1}{2N}\sum_{i=1}^{N}(y_i-\beta_0-x_i^T\beta)^2+\lambda\left(\frac{1-\alpha}{2}||\beta||_2^2+\alpha||\beta||_1\right)\right)$$

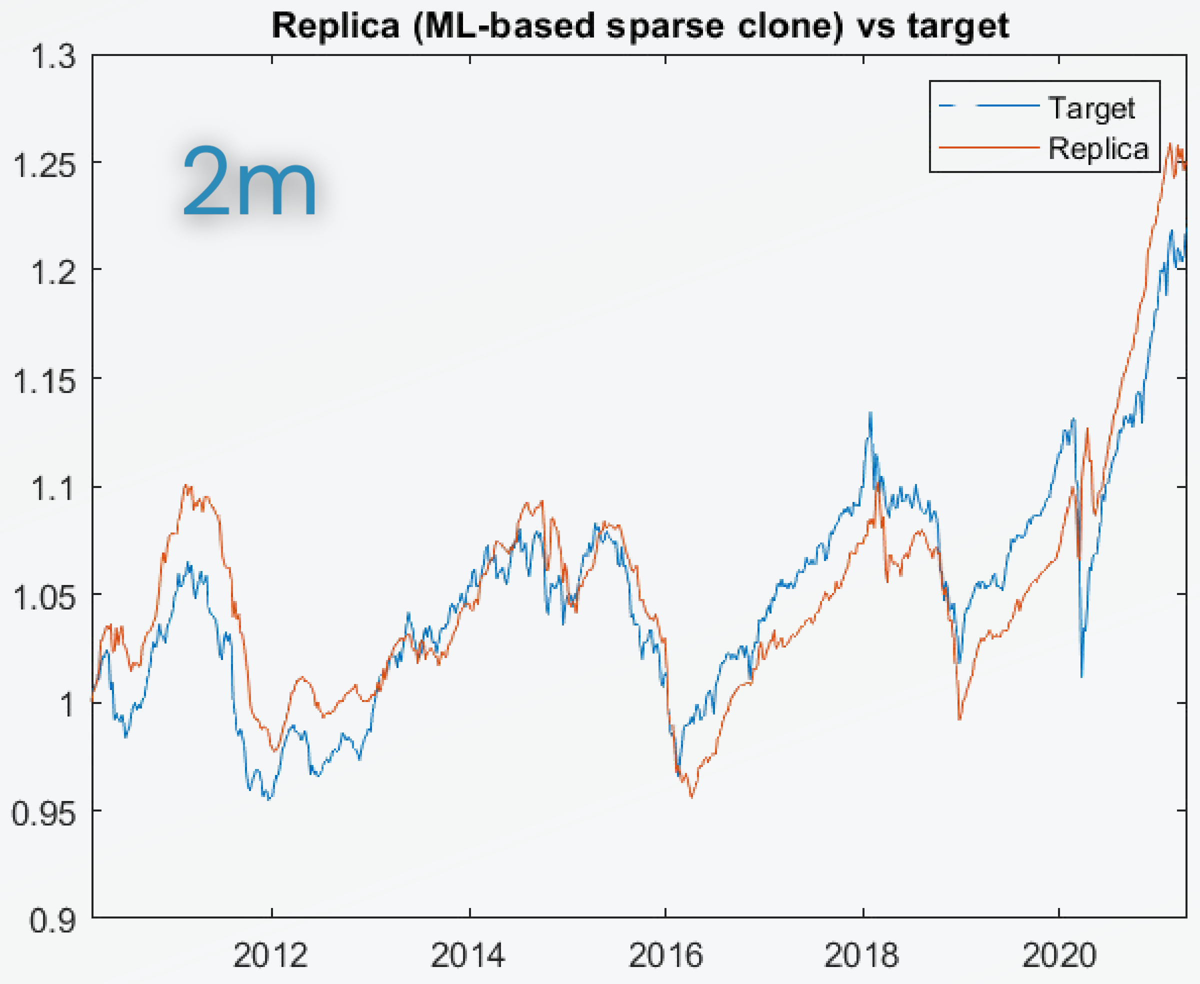
Sliding Window Analysis

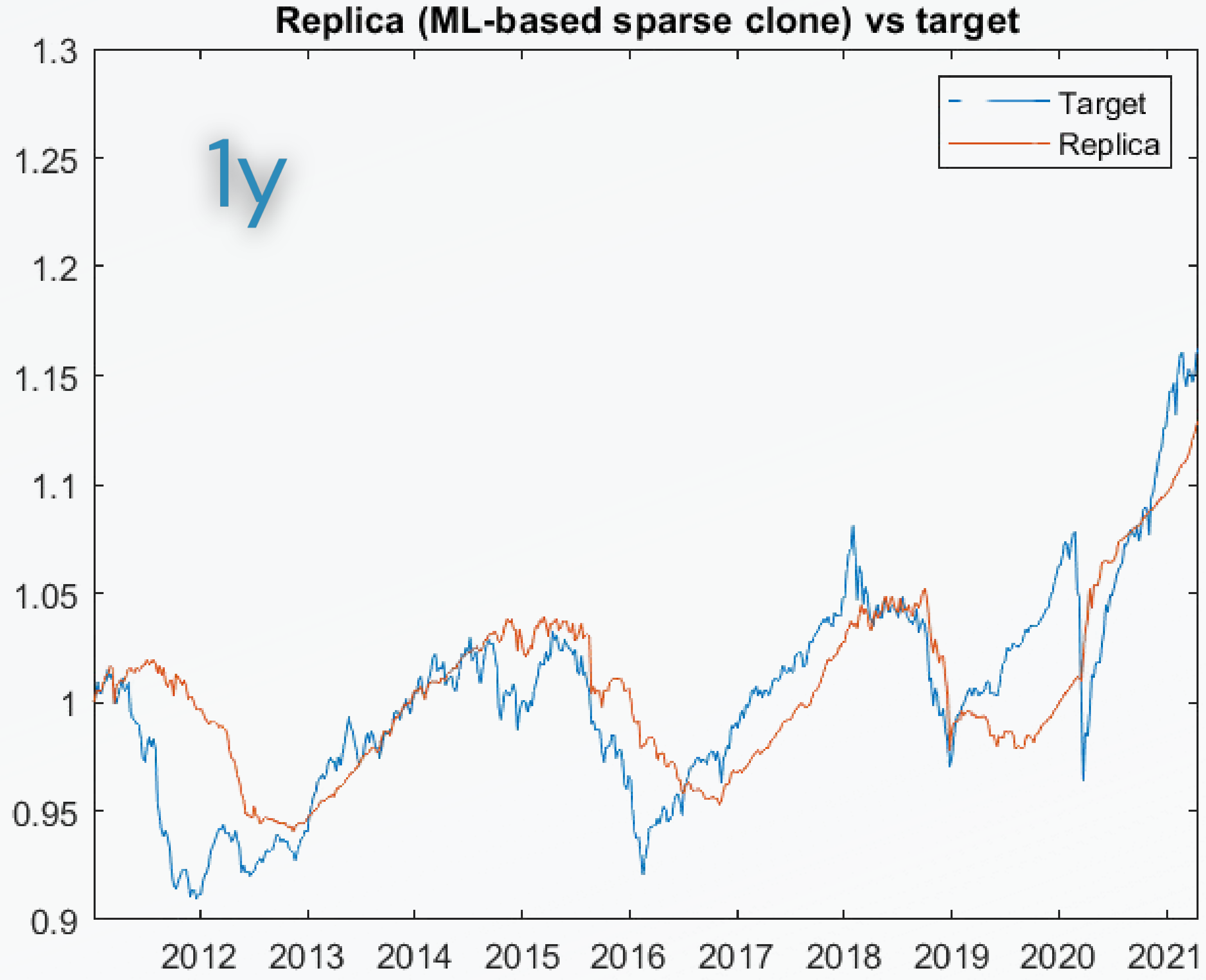
To train our model, we experimented with different sliding window lengths: 2 months and 1 year.

The **2-months sliding window** provided the **best results** for our portfolio replication. However, the **trade-off** with a 2-months window is **high weight volatility**, leading to high trading costs.
The choice of sliding window length can vary based on **client preferences:**

- **High-frequency trading** firms are accustomed to frequent rebalancing and thus do not face significant issues with high trading costs.
- **Private clients** may prefer a longer window to minimize rebalancing frequency and reduce commission costs.

Furthermore, the **forecasting horizon** influences the choice of sliding window length. If the goal is to forecast more than one day ahead, increasing the training window can help reduce extrapolation errors, making the model more robust.





Trading Costs Optimization

To expand our model adding an **optimization** of the **trading costs** we decided to add some additional **constraints**.

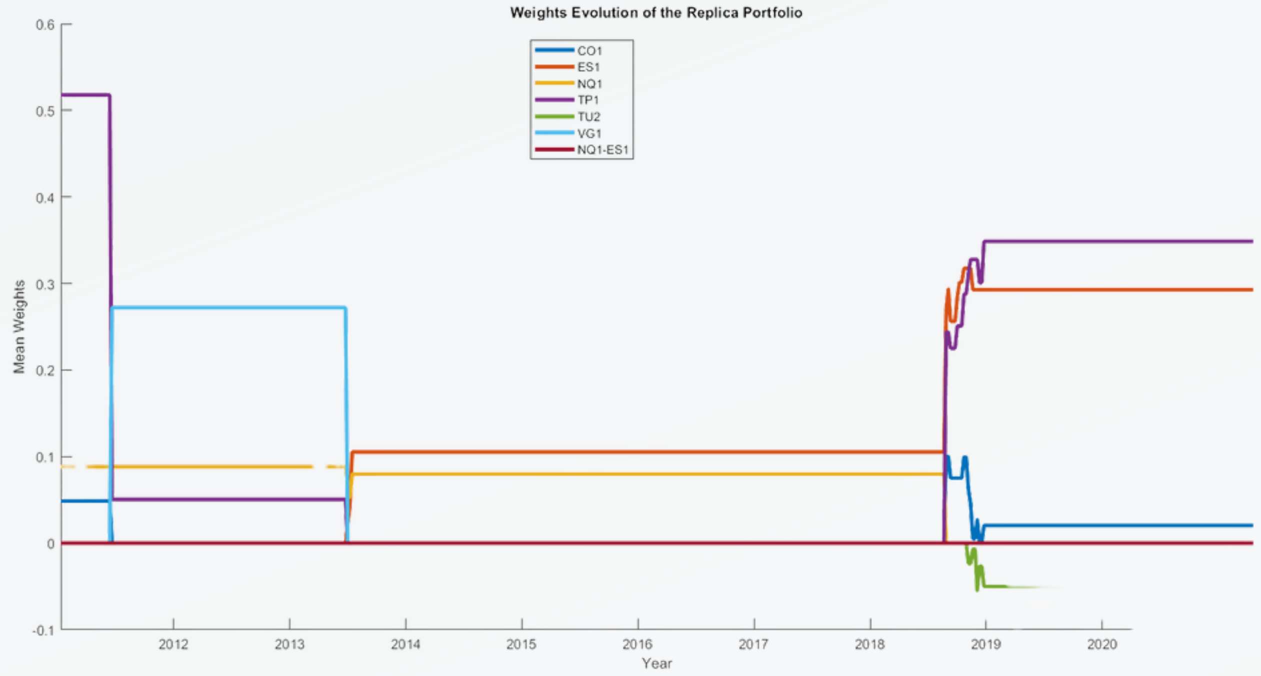
The first constraint was a maximum cap to the **total** absolute **weekly turnover.** Moreover, this check was suspended during periods of **high volatility**.

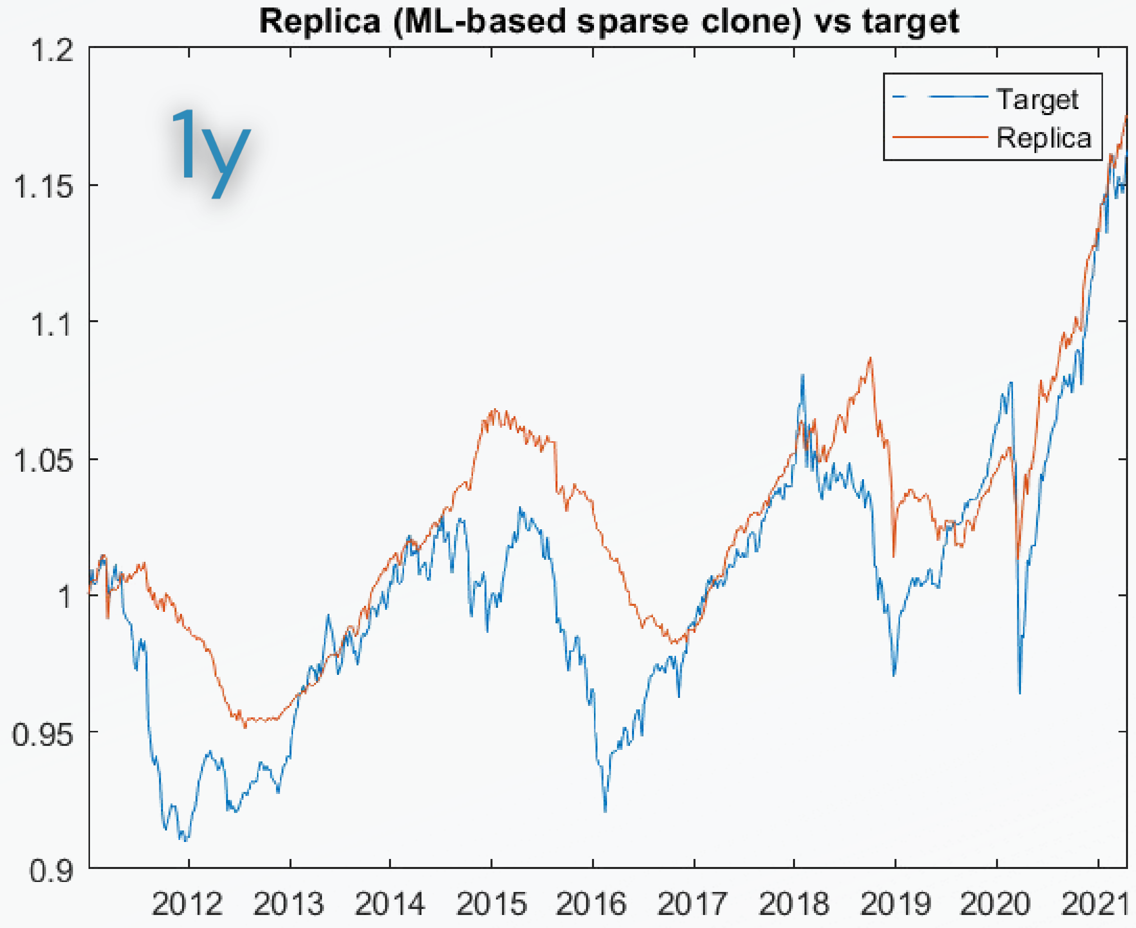
The second constraint instead was on the turnover of each **single weight**, reverting it to the previous week's value in case it was too low.

All these changes are aimed at **reducing unnecessary trades** in our portfolio, while still maintaining a **good performance**.

We observed good **results** on the **1 year** sliding window: as shown on the right we heavily reduced the weights' volatility while still keeping a good tracking.

Understandably, though, the **2 month** model was not satisfactory, as these constraints are limiting the frequent trading it needs.





EVALUATION METRICS

In evaluating our tracking model, we utilize a comprehensive set of **metrics** to ensure its **accuracy** and **efficiency**, as well as to observe the evolution of the model's **behaviour** when adding constraints or changing the sliding windows.

Mean Square Error (MSE) and **Mean Absolute Error** (MAE), which measure the model's prediction **accuracy**, taking into account the distances between our replica and the target index.

Tracking Error Volatility assesses the **variability** of the tracking error, so lower value indicates that the model's returns are more consistent and closer to the benchmark's returns, reflecting better performance and **stability**.

Information Ratio evaluates the risk-adjusted returns relative to the benchmark, compared to their returns, similarly to the Sharpe Ratio

Net Metrics also account for **trading costs**, providing a realistic view of the model's performance in practical scenarios.

Our results:

2m Lasso without constraints	
MAE	0.0281
TEV	0.0499
IR	0.0123
Mean Turnover	32.79
Mean Trading Costs	0.0131
Net Total Return	0.0055
Net Excess Return	-0.0125

By studying these metrics applied to our simplest model, even though MSE and MAE were great, we identified clear **issues**, particularly related to turnover and trading cost metrics.

These findings highlighted the need for a more **efficient** approach to reduce **excessive trading** and associated expenses. Analyzing these shortcomings guided us in **refining our strategy**, allowing us to develop a more **robust** and practical model suitable for real-world implementation.

MODEL EVOLUTION

Effectiveness of the Trading Costs Constraints

Before

1y Lasso without constraints	
MAE	0.0227
TEV	0.0414
IR	-0.0904
meanTurnover	11.00
Mean Trading Costs	0.0044
Net Total Return	0.0065
Net Excess Return	-0.0081

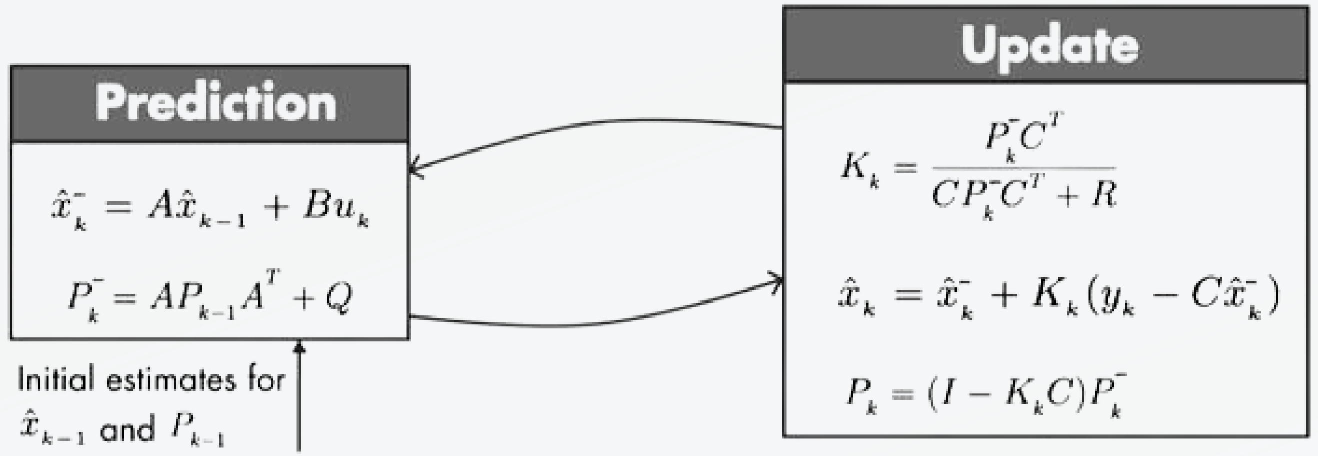
VS

After

1y Lasso with constraints	
MAE	0.0252
TEV	0.0399
IR	0.025
meanTurnover	0.3006
Mean Trading Costs	1.20E-04
Net Total Return	0.0156
Net Excess Return	8.79E-04

Upon comparing the **performance** of our two models, we find that the Mean Square Error (MSE), Mean Absolute Error (MAE), and Tracking Error Volatility (TEV) metrics show marginal superiority in the first model. This outcome is understandable given the absence of **constraints** in the first model, allowing for more **flexibility**. However, the second model exhibits a significantly lower **turnover**, indicating the effectiveness of the added constraints. Consequently, the **net metrics** show remarkable improvement in the second model. As a **result**, we are highly satisfied with the enhancements made, as they have led to a more efficient and practical model.

KALMAN FILTER

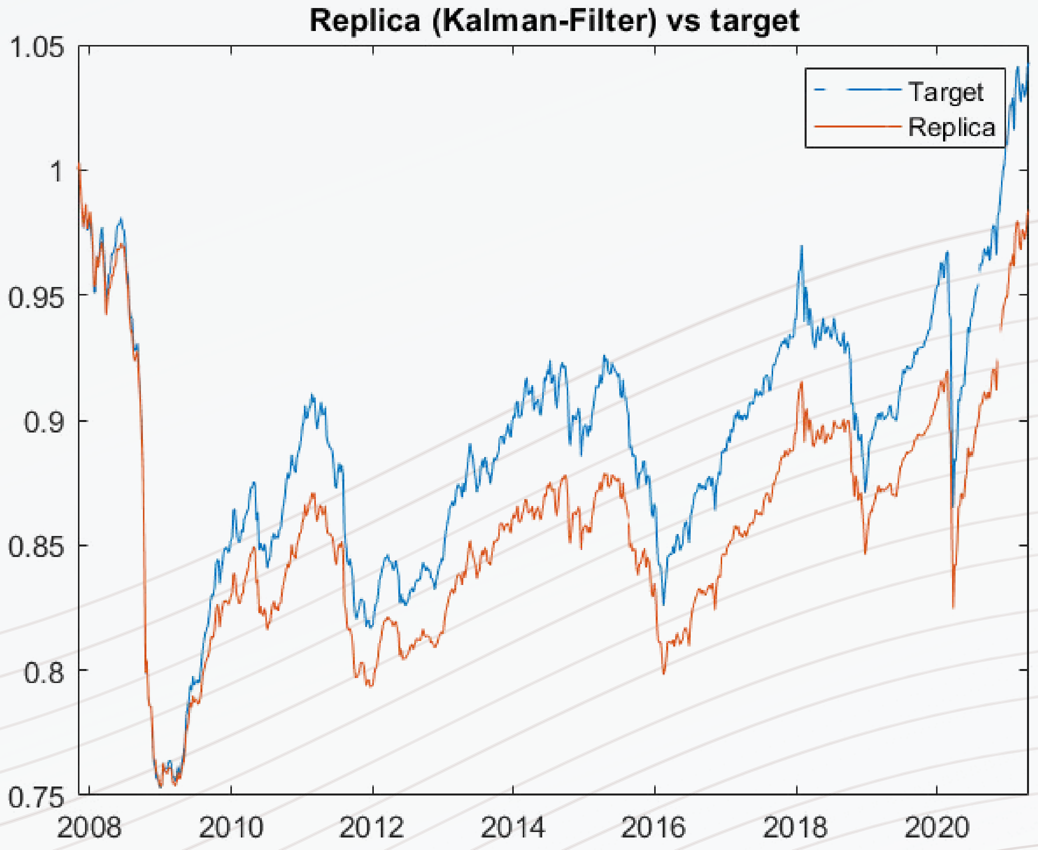


It’s an algorithm based on the state space representation that, through data, is able to create a distribution over the variables.
Using the filter we are able to predict the best weights each week.

The main driver of the algorithm is the gain:

- with a high gain the filter puts more weights on the most recent residual and thus conforms to them more responsively
- with a low gain the filter puts more weight on the previous prediction

A key step is the hyperparameter tuning. In particular, we experimented with the choice of Q, the noise of the process, and R, the noise of the observation.



BAYESIAN LINEAR MODEL

The idea is to formulate linear regression using probability distributions rather than point estimates.

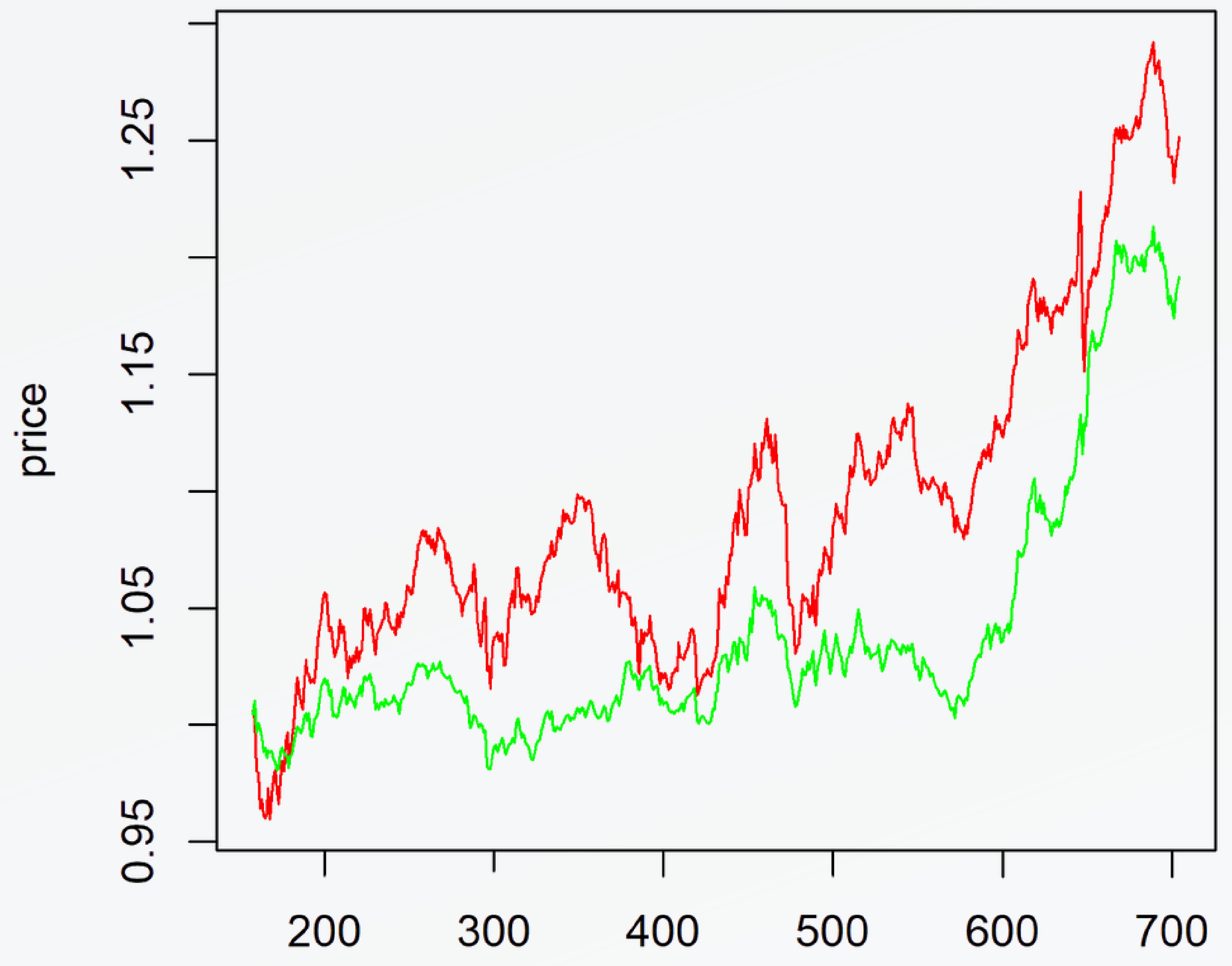
We exploited two different types of models:

1. **Non-hierarchical linear model:**

```
//priors:

    for ( j in  1 : K) {  beta[j] ~ normal(0,1) }
    sigma~ cauchy(0,5);

//likelihood

    y ~ normal(x * beta, sigma) ;
```

2. **Hierarchical linear model:**

```
//priors:

    mu ~ normal(0,3);
    sigma ~ cauchy(0,5);
    for ( j  in 1 : K  ){ beta[j] ~ normal(mu,sigma)}

// Likelihood
    y ~ normal(x * beta, 0.01);
```

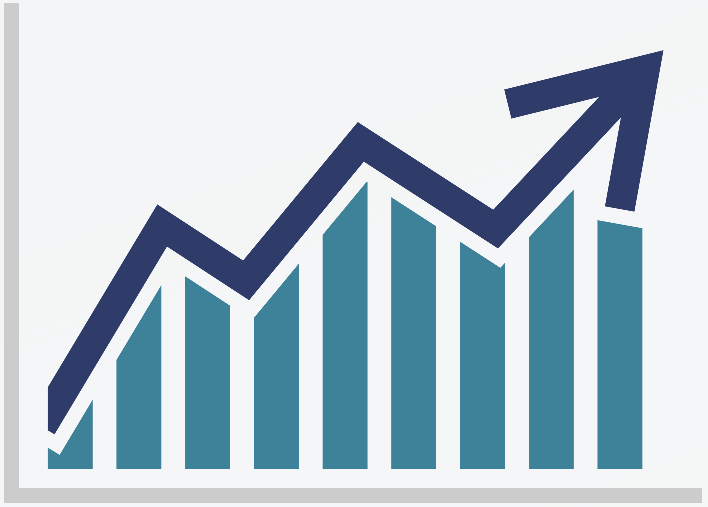


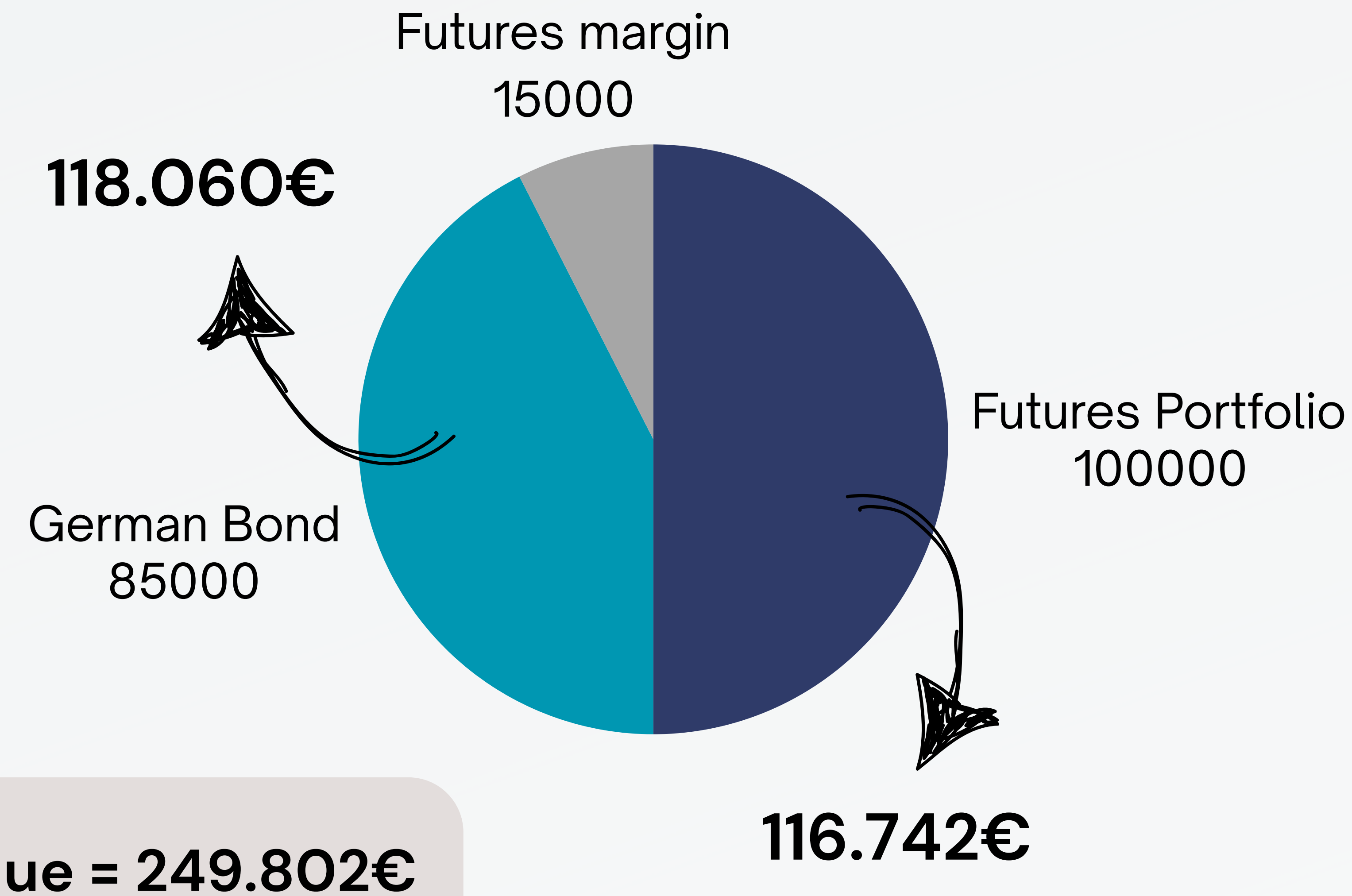
PROS AND CONS OF USING BAYESIAN APPROACH:

- **PROS**: more **flexibility** on the parameters
- **CONS** : high **computational cost** and necessity to make **assumptions** on priors and posteriors

FINAL RESULTS

Real life simulation of our cloning strategy





Final Value = 249.802€

Our results are highly favorable, as this approach eliminates subscription costs, management fees, and performance fees associated with hedge funds, while also avoiding limitations on minimum investment amounts.

For our replication strategy, we exploit the financial leverage to invest our notional in the futures portfolio and then we allocate 15% of the capital as collateral for futures margins, leaving 85% available to invest in safe bonds. Assuming we invest this entire amount in a single bond (in our case an AAA type of bond, such as the Germany Bond) with a maturity date at the end of the test period, we look up historical data to determine its performance if purchased at the start of the test period.
We use the net return of our model to evaluate the results of the replication portfolio and add the bond's yield.
In particular we employ the 1-year model with constraints, as it proves to be the best considering trading costs and tracking error, as evidenced by the net metrics, and consider a notional of 100k € (lower amounts often make investing in a hedge fund impossible).