University of Southampton

School of Mathematical Sciences

Data and Decision Analytics

Signature-Kernel Regime Detection and a Regime-Aware Lead–Lag Overlay

by

Zhe GUAN

September 2025

Supervisor: Dr. Sasan Barak
Second Examiner:

A dissertation submitted in partial fulfilment of the degree
of MSC Mathemetics

University of Southampton

ABSTRACT

SCHOOL OF MATHEMATICAL SCIENCES
DATA AND DECISION ANALYTICS

Master of Science

by Zhe GUAN

This dissertation studies lead–lag structure in the cryptocurrency market and its use for return prediction and portfolio construction. Rolling directed lead–lag matrices are built with a signature estimator on 30 business–day windows (update every day, maximum lag 7) after winsorising daily returns at the 2.5/97.5% tails. A baseline long–short hedge goes long on “followers” and short “leaders” identified by row–mean scores. A strict walk–forward regime detector is then introduced: Bitcoin path groups are compared using a signature–kernel MMD distance, clustered on historical data under a visibility constraint, mapped to bear/neutral/bull, and compressed to a daily regime by a rolling vote. The trading overlay conditions the baseline by the sign of each asset’s relation to the anchor and applies a fixed neutral scaling while preserving the baseline hedge.

The dataset contains daily closes for 72 Binance-listed assets from 2021-01 to 2024-06. At the same time–in–market, the regime overlay improves headline performance relative to the baseline: cumulative return 132.8% vs. 59.5%, CAGR 31.4% vs. 16.3%, Sharpe 1.46 vs. 0.58, annualised volatility 20.1% vs. 38.1%, and maximum drawdown -14.2% vs. -35.3%. It outperforms in three of four calendar years and turns a negative 2023 baseline (-5.42%) into a positive outcome (+7.79%). Robustness checks across estimator variants and parameter settings indicate stable signals; performance declines when detection and rebalancing are less frequent, consistent with slow information diffusion. Accounting for turnover-based transaction costs, the overlay remains attractive at 5–10 bps per side owing to lower switching, but net performance compresses sharply beyond $\sim$25 bps.

Keywords: lead–lag, return prediction, portfolio construction, regime detection, signature kernel, MMD, cryptocurrencies.

Acknowledgements

Completing this master's dissertation has been both an academic challenge and a personal milestone. Coming from a physics background, taking a first step into data–driven finance has felt like a small but meaningful achievement. I am grateful to everyone who helped make that transition possible.

I would like to express my sincere thanks to my supervisor, **Dr. Sasan Barak**, for clear guidance, steady support, and timely, constructive feedback at every stage. His encouragement to test ideas rigorously and try new methods helped me navigate the technical parts of this project and sharpen my analytical thinking.

I am deeply thankful to my parents for giving me the opportunity to study in the UK and for their constant support, both practical and emotional.

Finally, thank you to my friends in Southampton for the long study sessions, dissertation discussions, and all the small moments that made this year memorable. :)

Statement of Originality

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.
- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.
- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

You must change the statements in the boxes if you do not agree with them.

We expect you to acknowledge all sources of information (e.g. ideas, algorithms, data) using citations. You must also put quotation marks around any sections of text that you have copied without paraphrasing. If any figures or tables have been taken or modified from another source, you must explain this in the caption and cite the original source.

I have acknowledged all sources, and identified any content taken from elsewhere.

If you have used any code (e.g. open-source code), reference designs, or similar resources that have been produced by anyone else, you must list them in the box below. In the report, you must explain what was used and how it relates to the work you have done.

I have not used any resources produced by anyone else.

You can consult with module teaching staff/demonstrators, but you should not show anyone else your work (this includes uploading your work to publicly-accessible repositories e.g. Github, unless expressly permitted by the module leader), or help them to do theirs. For individual assignments, we expect you to work on your own. For group assignments, we expect that you work only with your allocated group. You must get permission in writing from the module teaching staff before you seek outside assistance, e.g. a proofreading service, and declare it here.

I did all the work myself, or with my allocated group, and have not helped anyone else.

We expect that you have not fabricated, modified or distorted any data, evidence, references, experimental results, or other material used or presented in the report. You must clearly describe your experiments and how the results were obtained, and include all data, source code and/or designs (either in the report, or submitted as a separate file) so that your results could be reproduced.

The material in the report is genuine, and I have included all my data/code/designs.

We expect that you have not previously submitted any part of this work for another assessment. You must get permission in writing from the module teaching staff before re-using any of your previously submitted work for this assessment.

I have not submitted any part of this work for another assessment.

If your work involved research/studies (including surveys) on human participants, their cells or data, or on animals, you must have been granted ethical approval before the work was carried out, and any experiments must have followed these requirements. You must give details of this in the report, and list the ethical approval reference number(s) in the box below.

My work did not involve human participants, their cells or data, or animals.

ECS Statement of Originality Template, updated August 2018, Alex Weddell aiofficer@ecs.soton.ac.uk

Contents

List of Figures

List of Tables

Nomenclature

AMH	Adaptive Markets Hypothesis
EMH	Efficient Market Hypothesis
TSMOM	Time-Series Momentum
FX	Foreign Exchange
RCM	Regime Classification Measure
DGP	Data-Generating Process
ARCH	Autoregressive Conditional Heteroskedasticity
GARCH	Generalized ARCH
SWARCH	Markov-Switching ARCH
NYSE	New York Stock Exchange
S&P 500	Standard & Poor's 500 Index
S&P Composite	Standard & Poor's Composite Index
FTSE 100	Financial Times Stock Exchange 100 Index
CAC 40	Cotation Assistée en Continu 40 (French equity index)
ECM	Error-Correction Model
COC	Cost of Carry
ECM-COC	Error-Correction Model with Cost of Carry
ARMA	Autoregressive Moving-Average
VAR	Vector Autoregression
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
LLR	Lead–Lag Ratio
HY	Hayashi–Yoshida (estimator / cross-correlation)
CET	Central European Time
CAPM	Capital Asset Pricing Model
FF3	Fama–French Three-Factor Model
FF5	Fama–French Five-Factor Model
SBM	Stochastic Block Model
DI-SIM	Directed spectral clustering (DI–SIM)
CCF	Cross-Correlation Function
AUC	Area Under the Curve
ARI	Adjusted Rand Index

MI	Mutual Information
BTC	Bitcoin (anchor asset)
WF	Walk–forward (rolling forward)
MMD	Maximum Mean Discrepancy
RBF	Radial Basis Function
NaN	Not a Number (missing value)
Inf	Infinity
bdays	Business days
z_t	Daily market regime label (0=bear, 1=neutral, 2=bull)
p_τ	BTC close price at time τ
γ	Path embedding $\{(\tau, p_\tau)\}$
$\widehat{\gamma}$	Transformed path $T(\gamma)$
T	Path transformer (level standardization, time normalization)
n_{steps}	Subpath length (number of steps)
n_{paths}	Number of subpaths per group
o	Overlap offset in subpath stride
G_g	The g-th path group
$[s_g, e_g]$	Calendar span covered by group G_g
K_Σ	Signature kernel
$\Phi(\cdot)$	Feature map induced by K_Σ
σ	RBF bandwidth
d	Dyadic order
λ	Regularization parameter
$D_{g,h}$	Group distance $\mathrm{MMD}(G_g, G_h)$
D	Precomputed distance matrix $\in \mathbb{R}^{G\times G}$
w	WF history window length (in groups)
H	Forward horizon (days)
n_c	Number of clusters
τ_+, τ_-	Thresholds for semantic mapping (bull/bear)
E_g	Eligible history set at decision g
m_c	Medoid of cluster c
μ_c	Mean H-day forward return of cluster c
$\psi(\cdot)$	Mapping from cluster to {bear, neutral, bull}
dir_label$_g$	Directional label of group G_g ($\in \{0, 1, 2\}$)
k	Count parameter: daily voting window size / minimum names (context-dependent)
L	Rolling window length (bdays)
f	Update/rebalance spacing (bdays)
$\ell_{\max}$	Maximum forward lag
m	Signature truncation order
$\mathcal{T}$	Set of update dates

N	Number of assets
$P_{i,t}$	Close price of asset i on day t
$r_{i,t}$	Log return of asset i on day t
M_t	Signature-based lead–lag directed matrix at date t
$x_i^{(t,\ell)}$	Past sequence of i on W_t (aligned to j's future)
$y_j^{(t,\ell)}$	Future sequence of j on W_t (shifted by ℓ)
$\Phi_m(\cdot)$	Signature feature vector up to order m
$\widehat{\Phi}_m(\cdot)$	Normalized signature features
$s_{ij}^{(\ell)}(t)$	Directional similarity score ("i then j" minus "j then i")
$\ell_{ij}^{\star}(t)$	Lag maximizing $\lvert s_{ij}^{(\ell)}(t)\rvert$
$s_t(i)$	Row-mean leadership score of asset i
$\mathcal{L}_t,\ \mathcal{F}_t$	Leaders / followers sets (by quantiles)
$Q_q(\cdot)$	Quantile operator (the q-quantile)
$w_{t^+}(i)$	Portfolio weight of asset i executed on t^+
R_{t^+}	Baseline strategy daily return realized on t^+
a	Anchor asset (here $a = \mathrm{BTC}$)
$\mathrm{rel}_t(i)$	Signed anchor relation for asset i (from M_t)
$\mathcal{B}^{\mathrm{long}},\ \mathcal{B}^{\mathrm{short}}$	Long / short baskets
$\mathcal{F}_t^+,\ \mathcal{F}_t^-$	Followers with $\mathrm{rel}_t(i) > 0$ / < 0
$\mathcal{L}_t^+,\ \mathcal{L}_t^-$	Leaders with $\mathrm{rel}_t(i) > 0$ / < 0
$R_{t^+}^{\mathrm{overlay}}$	Regime-aware overlay daily return
α	Neutral-period scaling coefficient (set to 0.5)
$\overline{r}_{\mathcal{B},t^+}$	Average return of basket $\mathcal{B}$ on t^+
t^+	Next trading day (relative to update date t)
$\mathcal{U}$	Investable universe tradable set

Chapter 1

Introduction

Financial markets, like cryptocurrencies, are widely considered to be non-stationary Schmitt et al. (2013); Lo (2004) as their statistical properties—including volatility, correlations, and return distributions change frequently over time. The non-stationarity brings a major challenge for trading and risk management, as optimised strategies under one regime often fail when market condition changes Ang and Bekaert (2002).

Traditional trading strategies generally assume a stable market environment, therefore the strategy performance probably varies across different market states. For example, a momentum-based method may perform well in a trending market but much worse in volatile periods Moskowitz et al. (2012); Similarly, risk measures calculated under a volatility regime may underestimate the risk during a crisis regime Hamilton and Susmel (1994).

To address the challenges from market non-stationarity, a range of methods are developed, such as Markov switching models Hamilton (1989), and clustering approaches Ang and Bekaert (2002). More recently, mathematical tools from rough path theory have given rise to signature methods Chevyrev and Kormilitzin (2025); Issa and Horvath (2023), which are capable of extracting rich path-wise features from time series data. Combined with clustering, these features can be used to identify distinct market regimes, such as bull, bear, and neutral phases.

In this study, we applied signature methods Lyons (1998); Chevyrev and Kormilitzin (2025); Issa and Horvath (2023) to extract path-wise features from BTC prices and perform walk-forward clustering to obtain regimes.

Lead–lag refers to directed, time-shifted dependence in returns across assets. A number of literature develops practical estimators and documents this structure via futures–spot studies, high-frequency (Hayashi–Yoshida–style) measures, and daily directed-network clustering, which shows economic value for cross-sectional trading Brooks et al. (2001);

Huth and Abergel (2012); Bennett et al. (2022). Building on this, we implement a cross-sectional lead–lag hedge Lyons and Qian (2002); Gatheral et al. (2018) as our baseline, and then overlay our previously estimated regime signal to adjust position signs and scale exposures.

Given our motivation for this study, this study attempted to address the following research question:

RQ1: How can market regimes be detected in a using signature–kernel *path-wise* features of an asset?
RQ2: Holding market participation fixed, do regime-conditioned signature lead–lag hedges improve risk-adjusted performance over an unconditional baseline?

RQ3: How sensitive are the results to key hyperparameters and trading frictions—like path segmentation, voting windows, and transaction costs?

In relation with the questions, the following objectives are aimed to achieve in this study:

RO1: Develop a data-driven regime detector based on signature-kernel MMD with strict walk-forward clustering and medoid semantics, producing daily bull/neutral/bear labels without look-ahead.
RO2: Build (i) a baseline signature-based lead–lag portfolio (long followers / short leaders) and (ii) a regime-aware overlay that preserves the hedge while routing names by their signed relation to a BTC anchor.
RO3: Evaluate performance with headline metrics (Sharpe, Sortino, drawdowns), track year-by-year behaviour, analyze sensitivity to hyperparameters (n_{steps}, n_{paths}), and consider transaction-cost scenarios.

To achieve these objectives, we devise the following methodology: **First**, we build a rolling, signature-based lead–lag matrix M_t from daily returns. Per-asset leadership is the row mean, and the baseline hedge goes long followers and short leaders, executed at t^+. **Second**, we detect regimes in a strict walk-forward way: slice BTC into rolling path groups; compute pairwise distances via signature-kernel MMD; cluster the eligible history within a window w; map clusters to bull/neutral/bear using the mean forward H-day return; then turn group labels into a daily series z_t with a k-vote. **Third**, we make the hedge regime-aware: split leader/follower sets by each asset's signed relation to BTC, $\text{rel}_t(i) = M_t(\text{BTC}, i)$. In bulls we go long co-moving names and short anti-moving names; in bears we flip; in neutral we keep the hedge but scale exposure by α. **Finally**, we evaluate: align signals to next-day execution, report Sharpe/Sortino, volatility, drawdowns, and yearly attribution, etc; run parameter grid (n_{steps}, n_{paths}); and test transaction-cost scenarios.

This study makes three contributions to the literature. First, the study introduces a leakage-free regime detector for crypto markets. The detector relies on signature-kernel MMD. Second, the study specifies a simple and reproducible signature-based lead–lag hedge and also defines a regime-aware overlay. The combined design improves risk-adjusted returns and does not increase time in the market. Third, the study sets out an evaluation protocol which applies a turnover-based cost model. The protocol includes sensitivity analyses over key hyperparameters and trading frictions.

The study evaluates results from January 2021 to June 2024.

The dissertation is organised as follows. Chapter 2 reviews the relevant literature on financial market non-stationarity, regime detection methods, and cryptocurrency trading strategies. Chapter 3 introduces the methodology, including signature methods, clustering techniques, and the construction of baseline and regime-aware strategies. Chapter 4 presents the empirical results and performance evaluation. Chapter 5 discusses the results and limitations. Finally, Chapter 6 concludes with a summary of contributions and directions for future research.

Chapter 2

Literature Review

2.1 Non-stationarity in Financial Markets

Non-stationarity is a defining feature of financial returns. In practice, the conditional mean, the variance, the higher moments, and the cross-sectional dependence structure evolve over time as market composition, policy, and technology change. Accordingly, this subsection organises the literature into four strands that motivate our empirical design. First, time-varying covariances can generate heavy tails. Second, adaptive market dynamics imply state- and time-dependent predictability. Third, cross-asset time-series momentum provides evidence of time-varying serial dependence. Fourth, volatility often switches across regimes. Consistent with these strands, we emphasise approaches that model non-stationarity explicitly, such as Markov switching and time-varying volatility. We also emphasise approaches that infer non-stationarity non-parametrically via rolling windows and clustering. Finally, we relate these ideas to our later use of *path-wise* signature features and strict walk-forward clustering for regime identification.

Schmitt et al. (2013) place non-stationarity at the center of their analysis. The authors study 306 continuously traded S&P 500 stocks from 1992–2012. In rolling windows, they show that volatilities and correlations vary strongly over time. When the covariance matrix is held fixed over short windows, multivariate returns appear approximately Gaussian. To capture the fact that covariances fluctuate, the authors replace the fixed covariance with a Wishart random matrix. This substitution yields an ensemble-averaged return distribution governed by a single parameter N, which measures the strength of correlation and covariance fluctuations. A smaller N indicates stronger non-stationarity and heavier tails. As $N \to \infty$, the model approaches Gaussian behaviour. Empirically, the model fits the market-wide return distribution well and attributes heavy tails to time-varying covariances. The estimated N rises with the return horizon. The estimate is about $N \approx 5$ for daily returns and about $N \approx 14$ for 20-day returns. This pattern implies that non-stationarity is stronger at short horizons. However, some deviations remain in the extreme tails. The study does not address the underlying economic mechanisms. Even so, the framework provides a consistent way to capture heavy tails through covariance non-stationarity.

Lo (2004) places non-stationarity at the core of the Adaptive Markets Hypothesis (AMH). In this view, market efficiency changes over time as the market ecology evolves. Moreover, the risk-return relation changes as conditions shift. Likewise, the equity risk premium and strategy profitability change with the environment. Specifically, the ecology includes participants, institutions, and regulation, which shift continually. Consequently, empirical relationships are time-varying and path-dependent rather than fixed. Within this framework, arbitrage opportunities emerge and disappear. Strategies wax and wane as conditions change. Thus, the AMH replaces the EMH's stationary convergence with cycles, trends, and episodes of panic and recovery. For evidence, a rolling five-year first-order autocorrelation for the S&P Composite (1871-2003) shows cyclical efficiency. The pattern does not march monotonically toward zero autocorrelation. Hence, the result challenges stationary-equilibrium implementations of the EMH. Practically, the model implies that adaptation and innovation are necessary responses to a changing environment. Accordingly, survival rather than static optimality becomes the organising objective. Finally, the AMH interprets behavioural regularities as evolutionary heuristics whose market impact varies with population composition. Moreover, emotion and selection shape which agents remain active. Taken together, these features provide additional evidence that the data-generating process is non-stationary.

Similarly, Moskowitz et al. (2012) study 58 liquid futures and forwards across equities, FX, commodities, and government bonds. The sample covers mainly 1985–2009. For comparability, the authors scale positions by ex-ante volatility. Consequently, the authors document strong 1-12-month continuation followed by partial multi-year reversal. Moreover, a diversified TSMOM factor delivers large alphas relative to standard equity factors and "everywhere" factors. The factor performs best in extreme markets. Hence, the payoffs appear state-dependent. In a formal decomposition, the authors attribute profits mainly to positive auto-covariance, which reflects time-series dependence. By contrast, the cross-asset lead-lag and mean-return components are small or even negative. In addition, position data show that speculators ride trends, whereas hedgers take the other side. Finally, these patterns support a non-stationary, state-dependent return-generating process rather than one with constant parameters.

In line with the above evidence on non-stationarity, Hamilton and Susmel (1994) replace a stationary GARCH view with a Markov-switching ARCH (SWARCH) specification. The specification lets volatility parameters jump across latent regimes. The authors estimate the model on weekly NYSE value-weighted returns from 1962 to 1987. The model allows 2-4 regimes and student's t innovations with a leverage term. The results show that standard GARCH overstates persistence and forecasts poorly. By contrast, SWARCH captures discrete regime shifts. The shifts include quiet, moderate, high, and an extreme state that isolates the October 1987 crash. The model improves short- and multi-week variance forecasts, especially for the four-state case. The paper also reframes persistence. The authors interpret persistence as state persistence, meaning long-lived regimes, rather than the slow decay of shocks. Once regime changes are modelled, the ARCH component becomes much less persistent. High-volatility states line up with business recessions. Negative returns raise volatility more than equal-size positive returns, which confirms leverage. These patterns further underscore time variation in the data-generating process. Overall, the paper treats equity volatility as a regime-switching, non-stationary phenomenon. Consequently, the framework yields cleaner inference and better forecasts than stationary ARCH/GARCH benchmarks.

Taken together, these strands support an empirical design. The design treats regimes as evolving objects and detects regimes from data rather than imposing them *ex ante*. Accordingly, our approach uses signature-based path features. Our approach also applies strict walk-forward clustering. Together, these tools yield daily bull, neutral, and bear states. We then condition a cross-sectional lead-lag hedge on those states.

2.2 Path Signatures for Sequential Data

A *path signature* is an infinite, ordered list of iterated integralsIssa and Horvath (2023). The signature captures both the shape and the time ordering of a multivariate path. The concept comes from rough path theory. Lyons (1998) formalise the signature as a unique encoding up to time reparameterisation. The representation is non-parametric and coordinate-free. For a path $X : [0, T] \to \mathbb{R}^d$, the k-th level is

$$S^{(k)}(X) = \int_{0<t_1<\cdots<t_k<T} dX_{t_1} \otimes \cdots \otimes dX_{t_k}.$$

This term is a tensor in $(\mathbb{R}^d)^{\otimes k}$. The level collects all k-fold, time-ordered integrals. These levels encode increments, signed areas, and higher-order interactions. Consequently, the signature summarises how the path evolves through time, not only where it ends. The signature is invariant to the speed of traversal, and also obeys key algebraic laws. The *shuffle product* and *Chen's identity* link products and concatenations of paths Chen (1954). These laws give the object a rich and tractable structure. A truncation at order m yields a finite feature vector. The vector summarises the path up to order-m effects. Thus, the signature provides practical features with clear theoretical meaning.

Over the past decade, path signatures have moved to the forefront of machine learning for sequential data. Lyons and collaborators have led this shift. A growing body of work shows that signature features are powerful inputs for time-series classification and prediction. For example, Chevyrev and Kormilitzin (2025) provides a concise primer. The authors highlight reparameterisation invariance, the existence of a log-signature in a free Lie algebra, and a universal approximation view. Consequently, the properties justify using signatures as a basis for feature learning. In practice, standard pipelines follow three steps. One first embeds raw discrete observations into a continuous path, for example via piecewise-linear interpolation. One then computes a truncated signature. The approximation result provides the key rationale. Polynomial functionals of a path reduce to linear functionals of signature terms. Therefore, a linear model on sufficiently high-order signature features can approximate any continuous functional on a suitable path space. This universality allows complex path dependence without manual feature engineering. Instead, the model learns weights on signature terms.

For financial applications, the second-order signature terms represent *areas* spanned by pairs of variables over time. These signed areas encode temporal ordering information. Consequently, researchers use these areas as measures of lead-lag or causal asymmetry between time series. For instance, if an asset A consistently loops ahead of another asset B in price-path space, the signed area becomes large. This pattern implies that A tends to lead B. Moreover, Bennett et al. (2022) include a rough-path signature *area* term in their lead–lag correlation measures. Likewise, Meng et al. (2017) proposes a hypothesis test for lag/lead relations based on the statistical significance of the signed area between two time series. Taken together, these applications show that path signatures detect subtle time-directed dependencies. Standard correlation or cross-covariance methods often miss such effects.

Another advantage is that signatures combine naturally with kernel methods. In particular, the *signature kernel* defines an inner product between signature feature maps. The kernel enables efficient comparison of paths in a high-dimensional feature space. The approach avoids explicit truncation of the signature. Building on this idea, Issa and Horvath (2023) develop an online two-sample test for regime shifts. The test uses signature features as the basis for a maximum mean discrepancy (MMD) statistic. Consequently, the method can flag, and flag quickly, when the distribution of recent market data departs from that of the past. More broadly, signature features capture fine-grained sequential structure and coarse aggregate effects. Therefore, the approach suits non-stationary financial problems.Accordingly, the remainder of this thesis will use path signature features via the signature kernel. We will embed asset price trajectories into a feature space. We will identify regimes and measure lead–lag effects in that space. Finally, we will remain agnostic about specific functional forms of dependence while exploiting the properties above.

2.3 Approaches to Regime Detection

Regime detection methods broadly fall into two families. *Parametric* models posit a finite set of latent states with state-dependent moments and Markovian transitions. As a result, these models yield smoothed or filtered state probabilities and transition matrices that aid forecasting. By contrast, *non-parametric* approaches infer state changes directly from the data's geometry. These approaches typically use rolling windows, distances or kernels, and clustering. Therefore, non-parametric methods relax distributional and Markov assumptions and also facilitate online updates. In our review, we compare methods along five dimensions. We consider modelling assumptions, estimation mode (batch vs. online and leakage control), output type (hard labels vs. probabilities), scalability to multivariate paths, and interpretability for portfolio use.

Ang and Bekaert (2002) specify regimes as latent states of a multivariate normal DGP with state-dependent means, volatilities, and correlations. The authors infer regimes and transition probabilities via the Hamilton/Gray filter. The authors summarise classification quality with the Regime Classification Measure (RCM). The model recovers economically interpretable states; for example, the model finds a persistent high-volatility, high-correlation bear state and a lower-risk normal state. The model also reproduces downturn exceedance correlations where Gaussian and asymmetric GARCH fail. Key strengths include probabilistic outputs, expected durations, and tractable forecasts. However, key trade-offs include reliance on parametric structure, often with constant transition probabilities, and misspecification risk when the true dynamics deviate from a finite-state Markov chain.

Issa and Horvath (2023) develop an online regime detector on *path space* using a two-sample MMD with rough-path signatures as features. The method uses the *signature kernel* to compare paths. The pipeline ingests small streaming batches for fast reactivity. The pipeline scales to multidimensional series and handles non-Markovian, path-dependent structure. A rank-2 variant incorporates filtration (conditional) information. A kernel-trick implementation avoids explicit signature truncation. The same machinery also supports clustering and outlier detection. The demonstrations cover synthetic data and real markets, including equity baskets and crypto. The main appeal is minimal modelling assumptions and online operation. The main cost is that outputs are often hard labels or distance-based scores without an explicit transition model.

Chevyrev and Kormilitzin (2025) provide a concise primer on path signatures. The authors describe signatures as an infinite sequence of iterated integrals that compactly capture time ordering. Core properties include time-reparametrisation invariance, the shuffle product, Chen's identities, time reversal, and the log-signature. These properties justify signatures as non-parametric feature maps for sequential data. A practical pipeline follows a simple flow: data $\rightarrow$ path $\rightarrow$ signature $\rightarrow$ features $\rightarrow$ learning. In many cases, second-order (area-type) terms improve class separability and extend naturally to multivariate time series. Linear functionals of signature terms approximate continuous functionals on compact path sets. Hence, signature features provide a principled basis for often linear models.

Guided by the above evidence, we adopt a non-parametric and *strictly walk-forward* design. Our procedure embeds BTC paths via the signature kernel. Our rolling group distances then feed a clustering step on *eligible history* only. Our mapping step assigns clusters to bull, neutral, or bear using forward H-day effects. Our voting step converts hard labels into a daily series via a k-vote. Compared with Markov-switching models, our design avoids parametric transition assumptions and emphasises leakage control. Compared with generic online detectors, our design targets portfolio use by tying cluster semantics to forward returns and by conditioning a cross-sectional lead–lag hedge on the detected state.

2.4 Clustering-Based Regime Identification

Beyond model-based switching frameworks, researchers identify market regimes with *unsupervised clustering* of historical data. These approaches partition time periods into groups with internally similar characteristics. The characteristics often include return distributions and volatility levels. These methods do not require an explicit likelihood or a Markov structure. The main advantage is that clustering makes minimal assumptions about dynamics. As a result, clustering can detect novel or irregular regimes. The main challenge is that clustering typically produces hard classifications with no inherent temporal continuity. Unless researchers add an exogenous transition process, the labels do not evolve smoothly. Nevertheless, prior studies show that clustering can uncover economically meaningful states in financial markets.

One influential example is the Wasserstein k-means algorithm proposed by Horváth et al. (2021, 2024). This method treats each time window as a distribution. This method then clusters windows using the p-Wasserstein distance between empirical return distributions. This design contrasts with classical k-means, which clusters raw returns by Euclidean distance. Because the method considers the full distributional shape, the method is robust to heavy tails and skewness. The authors show that Wasserstein k-means can detect distinct market phases in an unsupervised fashion. The authors also show that the method significantly outperforms classical k-means and other baselines in separating regimes. The resulting clusters correspond to intuitive states, such as low-volatility and high-volatility regimes. Importantly, the method achieves this separation without a parametric state model. Follow-up work validates the method by computing maximum mean discrepancy scores between clusters. These studies confirm that the distributional differences are statistically significant. In both simulations and real financial time series, the Wasserstein clustering distinguishes regime segments reliably. In addition, the method adapts more flexibly than models that assume Gaussian returns or fixed transition probabilities.

Clustering methods can also incorporate cross-sectional information. Chen et al. (2022) argue that many regime detectors cannot identify new regimes on the fly or ignore relationships across assets. The authors propose a dynamic clustering model that jointly learns regime assignments for multiple time series while allowing time-varying transition probabilities. The idea is to group time periods by common cross-asset behaviour rather than by univariate patterns. This design yields cross-sectional regimes that improve multi-asset return forecasts. In this framework, regimes emerge as clusters of high-dimensional observations across assets. A flexible transition mechanism then captures how these clusters evolve. This line of work underscores the breadth of clustering approaches. The spectrum runs from simple rolling-window similarity checks to sophisticated high-dimensional models. The shared goal is to let the data speak for itself about regime structure.

Another related strand is non-parametric change-point detection. Researchers can view this strand as clustering with implicit two-segment clusters, namely pre-break and post-break. Matteson and James (2014) develop an e-divisive means algorithm to detect multiple change points in multivariate series without parametric distributional assumptions. The method identifies points where the distribution shifts and thereby segments the series into distinct regimes. These techniques complement clustering. Whereas clustering often groups windows that need not be contiguous in time, change-point methods explicitly seek contiguous regime segments. Practitioners can also combine the two ideas by first detecting candidate breakpoints and then clustering segments.

Clustering-based methods typically output hard labels rather than smoothed probabilities. This property yields clear ex-post identification of regimes. However, this property also requires care in online or trading contexts to avoid lookahead bias. Many studies perform clustering on an entire historical sample or use rolling re-training that inadvertently uses future information. For practical trading signals, a strict walk-forward implementation is necessary. Such an implementation must use only information that is available at the decision time. Our design follows this principle by using an expanding window for clustering. Our design does not re-label past data when new data arrives. Our design defines regimes via forward-looking returns on a training set and then applies those definitions prospectively.

Clustering methods generally do not attach economic semantics to regimes by default. Users usually attach semantics by examining cluster characteristics, such as high volatility or large drawdowns, and then assigning names. Our approach bridges this gap by mapping clusters to bull, neutral, or bear labels based on forward H-day return outcomes in a training period. This mapping injects an economic interpretation directly into the clustering output. Thus, our approach forms a hybrid. The hybrid combines unsupervised identification of patterns with a supervised labeling step based on subsequent performance. This blend, together with strict walk-forward operation, aims to harness the adaptability of clustering while making the results usable in a live strategy.

2.5 Lead–Lag Signals and Regime-Driven Overlay Trading Strategies

Lead–lag captures a *directed, time-shifted* dependence between assets. Specifically, the empirical evidence comes from two main strands. The first strand covers high-frequency studies that document futures-led price discovery and microstructure asymmetries. The second strand examines daily, multi-lag dependence that researchers organise as directed networks and convert into predictive signals. A recurring constraint in this literature is the role of *trading frictions and execution latency.* These frictions determine whether statistical leadership translates into net returns. Accordingly, we summarise representative studies and we connect them to our regime-driven overlay design.

Brooks et al. (2001) study FTSE 100 futures and cash with 10-minute data from 1996–1997. The authors first establish cointegration via Engle–Granger. The authors then estimate an error-correction model (ECM) and a cost-of-carry variant (ECM–COC) against ARMA and VAR benchmarks. Lagged futures returns significantly forecast spot returns. The carry-adjusted cointegration term is also significant. In a May 1997 hold-out, ECM–COC delivers the best out-of-sample accuracy with 68.75% correct direction and the lowest RMSE and MAE. However, realistic round-trip costs (spot $\approx$ 1.70%, futures $\approx$ 0.116%) and a 10-minute execution delay remove the edge. Therefore, the key takeaway is clear. *Futures lead price discovery, but frictions and latency can exhaust exploitable profits.*

Huth and Abergel (2012) analyse CAC40 stocks and index futures with tick-by-tick data from March to May 2010. The authors use the *Hayashi–Yoshida* correlation and an asymmetric lead–lag ratio (LLR). The simulations show that previous-tick estimators spuriously assign leadership to more active assets. The Hayashi–Yoshida estimator mitigates this bias. The empirical results show that futures lead constituents at sub-second lags with strong asymmetry (future/stock LLR $\approx$ 2). The stock–stock links are weaker with peak lags near one second. The leadership aligns with liquidity through shorter intertrade durations, narrower spreads, lower midquote volatility, and higher turnover. The leadership also strengthens around macro releases and the U.S. open. A simple predictor that uses only the leader's past yields about 60% accuracy for the lagger's next midquote change. Nevertheless, naïve market-order strategies do not clear the spread. Hence, the evidence supports a second conclusion. *Leadership–liquidity co-moves are robust, but profitability is fee- and latency-constrained.*

Bennett et al. (2022) propose an unsupervised pipeline for daily data. The authors construct a directed network from pairwise lead–lag scores. The scores include cross-correlation functionals across lags (Pearson, Kendall, distance correlation, and mutual information; *ccf-lag1* and *ccf-auc*). The scores also include a rough-path *signature* area term. The authors then extract communities with high flow imbalance via Hermitian spectral clustering, with DI-SIM and bibliometric symmetrisation as baselines. On synthetic factors and on 434 U.S. equities (CRSP daily closes, 2000–2019), *ccf-auc* with non-linear dependence plus Hermitian or DI-SIM recovers ground-truth communities with high ARI. The procedure uncovers leading clusters that are not reducible to sectors. The procedure also passes a permutation test on the top Hermitian eigenvalue ($p < 0.005$). The learned structure yields a statistically significant trading signal. Thus, the results show that *daily data also contain stable directed structure.* However, the results remain sensitive to the metric and to the symmetrisation step.

Lu et al. (2025) decompose daily returns into overnight and daytime components. The authors build directed and signed networks for overnight→daytime and daytime→overnight links. The authors introduce d-LE-SC, which derives from a directed SBM likelihood with flow objectives, to extract leader and lagger communities. The portfolios trade within laggers by using signals from leaders. The design isolates cross-stock spillovers from within-stock autocorrelation. In U.S. equities (CRSP; 2000–2024), the overnight→daytime strategy delivers a 32.11% annualised return with a Sharpe ratio of 2.37 and a Calmar ratio of 1.84. The strategy outperforms the reverse link and a close-to-close benchmark. The alphas remain significant under CAPM, FF3/FF5, and momentum-augmented models. The turnover is manageable, and the costs attenuate but do not eliminate performance. Therefore, the results highlight *networked price discovery and correction* beyond single-stock reversals.

Taken together, the literature suggests two practical lessons. The first lesson is that leadership is strongest at high frequency but remains fragile to frictions. The second lesson is that, at daily horizons, multi-lag directed dependence can be organised into communities and used for prediction. Building on these insights, we operate at *daily* frequency and we use *signature* features and construct an *antisymmetric* lead–lag matrix whose row means rank leaders and followers. We implement a baseline hedge that goes long followers and short leaders with next-day execution. We then add a *regime-driven overlay.* We route assets by the sign of their BTC-anchored relation and by strictly walk-forward signature–kernel regime labels, and tilt toward co-movers in bull regimes, flip in bear regimes, scale down in neutral states, and preserve the hedge at all times. Consequently, our design uses *state information* to mitigate mismatch and left-tail risk. At the same time, our design emphasises *leakage control* through strict walk-forward operation, ties cluster semantics to forward H-day returns, and integrates regimes directly into portfolio construction.

2.6 Research Gap

The literature reports extensive evidence on non-stationarity. The literature also documents a growing body of work on lead–lag structure. However, an framework that couples path-wise regime detection with a deployable lead–lag trading overlay remains underdeveloped. This gap is especially visible in cryptocurrency markets.

First, on regime detection, parametric Markov-switching models offer interpretable state probabilities and transition matrices. However, these models rely on restrictive distributional and Markov assumptions that may shift across episodes. By contrast, nonparametric and online approaches on path space relax these assumptions. Yet these approaches rarely deliver an *economic* mapping that directly informs portfolio decisions. In particular, prior studies seldom link clusters to forward returns in a strictly walk-forward manner. Moreover, many unsupervised classifiers are applied to full samples or recalibrated with hindsight. Such practices create problems for live implementation.

Second, on lead–lag estimation and trading, the strongest leadership effects appear at high frequency. Microstructure frictions and latency often erase profits in those settings. Daily-horizon studies document usable directed dependence in equities via multilag networks. However, the evidence in crypto markets remains limited. The use of *signature-based, antisymmetric* scores that encode temporal order beyond linear correlation also remains rare. Furthermore, the idea of routing cross-sectional positions by a *signed relation to an anchor* (for example, BTC) is largely unexplored. Such anchors may plausibly concentrate market-wide propagation.

Third, at the strategy–design interface, most papers develop regime signals or lead–lag signals *in isolation*. The literature lacks regime-*aware* overlays that preserve a market-neutral hedge while reassigning names dynamically by state. A practical overlay would favour co-movers in bull regimes. A practical overlay would flip in bear regimes. A practical overlay would scale in neutral periods. The overlay would also hold *the same market participation*. Without this control, many comparisons conflate signal quality with changes in gross exposure or time in market. Such conflation obscures the true source of improvements.

Finally, many evaluation protocols underplay practical constraints. Several studies do not enforce strict walk-forward retraining. Several studies also do not align signals to next-day execution and treat costs only superficially. These gaps limit external validity and deployability.

This dissertation addresses these gaps with three steps. First, we construct strict walk-forward, signature-kernel regime labels whose semantics tie to forward H-day returns. Second, we estimate a signature-based, antisymmetric lead–lag matrix and form a simple followers-long/leaders-short hedge. Third, we overlay the hedge with regime-aware routing via a BTC-anchored signed relation while preserving market neutrality and aligning signals to t^+ execution under explicit transaction-cost scenarios. Remaining open questions include probabilistic (soft) regime labels, joint multi-asset regime estimation, and cost-aware optimisation that endogenises turnover. These questions define directions for future work.

Chapter 3

Methodology

3.1 Overview

This chapter presents an end-to-end pipeline. The pipeline links path-wise feature extraction, walk-forward regime detection, portfolio construction, and performance evaluation. We start from daily BTC closes. We then compute signature-based features that preserve temporal ordering, detect market regimes in a strictly leakage-free manner and finally use these regimes to condition a cross-sectional lead–lag hedge. All signals that we generate on day t are executed on the next trading day t^{+}. All evaluations follow a unified daily calendar.

First, we extract *path-wise* information with signature methods. We work with signature features or the signature kernel. These tools capture both the magnitude of price moves and the order of increments. This choice yields a representation that goes beyond scale-based indicators such as volatility. This representation encodes the geometry of the path.

Second, we run *walk-forward clustering* on rolling path groups to obtain bull, neutral, and bear labels. At each decision date, our clustering uses only eligible historical groups whose forward windows are fully observed. Our mapping assigns clusters to economic states by their mean H-day forward returns. Our voting rule converts group-time labels into a daily series via a rolling vote. This procedure estimates semantics strictly from the past and avoids look-ahead.

Third, we form a baseline *lead–lag* hedge and we overlay regime information. The baseline computes a rolling, directed lead–lag matrix from daily returns, and then forms an equal-weight long–short portfolio that is *long followers* and *short leaders.* The regime-aware overlay preserves this hedge structure. The overlay *routes* constituents by the signed relation to a BTC anchor. In bull states, the overlay places co-moving assets on the long side and anti-moving assets on the short side. In bear states, the overlay flips

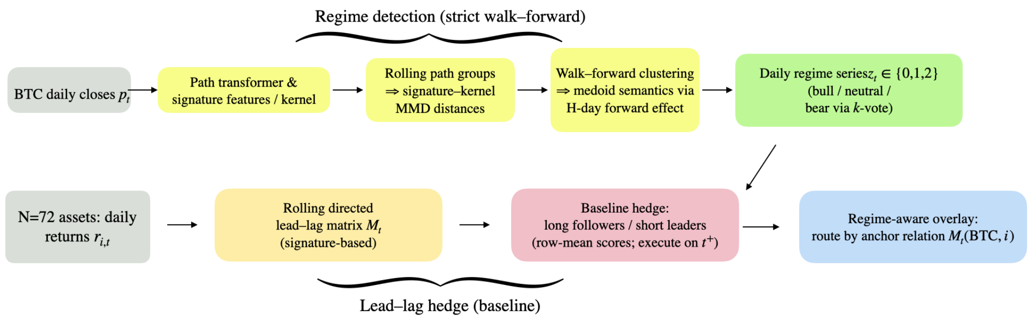
Regime detection (strict walk–forward)
BTC daily closes p_t
Path transformer & signature features / kernel
Rolling path groups ⇒ signature–kernel MMD distances
Walk–forward clustering ⇒ medoid semantics via H-day forward effect
Daily regime series$z_t \in \{0,1,2\}$ (bull / neutral / bear via k-vote)
N=72 assets: daily returns $r_{i,t}$
Rolling directed lead–lag matrix M_t (signature-based)
Baseline hedge: long followers / short leaders (row-mean scores; execute on t^+)
Regime-aware overlay: route by anchor relation $M_t(\mathrm{BTC}, i)$
Lead–lag hedge (baseline)

FIGURE 3.1: BTC-regime-aware lead-lag pipeline

the assignment. In neutral states, the overlay scales its effect by a fixed factor while keeping the hedge intact.

Finally, we evaluate both the baseline and the regime-aware strategies under a common protocol. We also report Sharpe and Sortino ratios, annualised volatility, cumulative return, and maximum drawdown. We finally analyse transaction-cost sensitivity separately.

3.2 Regime Detection

3.2.1 Data and Pre-processing

This study constructs a daily market regime label $z_t \in \{0, 1, 2\}$, where $0 =$ bear, $1 =$ neutral, and $2 =$ bull. The label is built from the anchor asset BTC from January 1, 2021 to June 30, 2024. The construction relies on path-wise statistics based on the signature kernel. Accordingly, the signal remains purely backward-looking at each date. The downstream portfolio module later uses this signal to modulate weights (Section 3.4.2).

The notation defines p_τ as the BTC close price at time τ on a daily calendar. The time series is embedded as a path $\gamma = \{(\tau, p_\tau)\}_\tau$. The methodology operates on rolling *subpaths* of fixed length. All timestamps are aligned to a unified timezone. Timestamps are also deduplicated before segmentation.

3.2.2 Path Segmentation and Transformation

The configuration fixes integers $n_{\text{steps}} \geq 2$ and $n_{\text{paths}} \geq 2$ and an overlap offset $o \geq 0$. The procedure slices γ into consecutive subpaths of length n_{steps} with stride $(n_{\text{steps}} - o)$:

$$\text{sub\_paths} = \{\gamma^{(k)}\}_{k=1}^{N}, \qquad \gamma^{(k)} = \big((\tau_{k,1}, p_{k,1}), \ldots, (\tau_{k,n_{\text{steps}}}, p_{k,n_{\text{steps}}})\big).$$

Next, the method collects *groups* of n_{paths} consecutive subpaths for two-sample comparisons:

$$\text{groups} = \{G_g\}_{g=1}^{G}, \quad G_g = \left(\gamma^{(g)}, \gamma^{(g+1)}, \dots, \gamma^{(g+n_{\text{paths}}-1)}\right),$$

so that each group G_g covers a calendar span $[s_g, e_g] = [\tau_{g,1},\ \tau_{g+n_{\text{paths}}-1, n_{\text{steps}}}]$.

Before computing similarities, a *path transformer* T standardises levels to the initial value and normalises time:

$$\widehat{\gamma} = T(\gamma); \quad T = (\text{standardise\_path}, \text{time\_normalisation}).$$

For clarity, the baseline disables other transforms. The exclusions cover differences, cumulants, lead–lag lifts, and invisibility.

For each subpath $\gamma^{(k)} = ((\tau_{k,1}, p_{k,1}), \dots, (\tau_{k,n_{\text{steps}}}, p_{k,n_{\text{steps}}}))$, the transformer applies level standardisation and time normalisation:

$$\tilde{p}_{k,j} \ = \ \frac{p_{k,j}}{p_{k,1}}, \qquad \tilde{\tau}_{k,j} \ = \ \frac{j-1}{n_{\text{steps}}-1}, \quad j = 1, \dots, n_{\text{steps}}.$$

Thus, the transformed subpath $\widehat{\gamma}^{(k)} = \{(\tilde{\tau}_{k,j}, \tilde{p}_{k,j})\}_{j=1}^{n_{\text{steps}}}$ starts at level 1 and lives on $[0, 1]$.

3.2.3 Signature–Kernel MMD Distance

The representation uses the path signature, a sequence of iterated integrals that encodes time ordering. The design avoids explicit truncation and explicit coordinates. Instead, the approach employs the *signature kernel* K_Σ, which provides an implicit feature map $\Phi(\cdot)$ on path space. Consequently, the kernel enables two-sample testing and clustering without manual feature design Chevyrev and Kormilitzin (2025).

The implementation instantiates K_Σ via `higherOrderKME.sigkernel` with an ambient RBF on $(\text{time}, \text{level})$. The settings use bandwidth $\sigma = 1$, dyadic order $d = 2$, and regularisation $\lambda = 1$ Salvi et al. (2021). In turn, this configuration yields *order-aware* similarities that suit regime discrimination.

The comparison between two groups of paths G_g and G_h uses maximum mean discrepancy (MMD) under K_Σ. The analysis denotes the implicit feature map by $\Phi(\cdot)$. The unbiased MMD^2 between bags of paths $A = \{x_a\}$ and $B = \{y_b\}$ is

$$\text{MMD}^2(A, B) = \frac{1}{|A|(|A|-1)} \sum_{a \neq a'} K_\Sigma(x_a, x_{a'}) + \frac{1}{|B|(|B|-1)} \sum_{b \neq b'} K_\Sigma(y_b, y_{b'}) - \frac{2}{|A||B|} \sum_{a,b} K_\Sigma(x_a, y_b).$$

Accordingly, the procedure defines a symmetric *group distance* as $D_{g,h} = \text{MMD}(G_g, G_h)$. The system computes and stores the full precomputed distance matrix $D \in \mathbb{R}^{G \times G}$ for each $(n_{\text{steps}}, n_{\text{paths}})$ configuration.

Algorithm 1: Walk–forward regime detection (signature kernel)

```
Input  : BTC daily closes $p_t$ on a unified calendar (2021-01-01 to 2024-06-30)
Params: Segmentation ($n_{\text{steps}}, n_{\text{paths}}, o{=}0$); signature kernel $K_\Sigma$ (RBF, $\sigma{=}1$,
        dyadic order 2, $\lambda{=}1$); WF window $w{=}30$ (groups); forward horizon
        $H{=}30$ (days); clusters $n_c{=}3$; thresholds $(\tau_+, \tau_-){=}(0,0)$; daily vote $k{=}7$
Output: Group labels dir_label$_g \in \{0,1,2\}$ and daily regimes $z_t \in \{0,1,2\}$
Segment & transform: slice the path into subpaths of length $n_{\text{steps}}$, stack $n_{\text{paths}}$
  to form groups $G_g$ with spans $[s_g, e_g]$; standardise level and normalise time
Distances: compute precomputed $D$ with $D_{g,h} = \text{MMD}(G_g, G_h)$ under $K_\Sigma$; clean
  $D$ (diag=0, symmetrise, fix NaN/Inf, clamp $< 0$ to 0)
for $g \leftarrow w - 1$ to $G - 1$ do
    Eligible history $E_g \leftarrow \{u \in [g - w + 1,\ g - 1]\ \mid$ the $H$-day forward return
      after $e_u$ is fully observed by just after $e_g\}$; if $|E_g| < n_c$, continue
    Cluster & prototypes: run Agglomerative (complete, precomputed) on
      $D[E_g, E_g]$ to get $n_c$ clusters; for each cluster $c$, take medoid $m_c$ (min row-sum
      distance) and its forward effect $\mu_c$
    Semantic mapping: set $\psi(c){=}2$ if $\mu_c > \tau_+$ (bull); $\psi(c){=}0$ if $\mu_c < \tau_-$ (bear);
      otherwise $\psi(c){=}1$ (neutral)
    Label current group: assign $G_g$ to nearest medoid by $D$; let $c^\star$ be its cluster;
      set dir_label$_g \leftarrow \psi(c^\star)$
Daily series: for each day $t$, set $z_t$ to the majority label among the last $k$
  completed groups ($e_u \le t$), breaking ties by recency; optionally merge consecutive
  equal labels for plotting
```

Overall, grouping n_{paths} consecutive subpaths forms two short bags of recent behaviour. The unbiased MMD on K_Σ then serves as a path-wise two-sample distance between historical windows. Consequently, the distance highlights persistent geometric changes rather than pointwise volatility alone.

3.2.4 Strict Walk–Forward Retraining and Labeling

The calendar span of group G_g is $[s_g, e_g]$. The historical window length is $w = \mathbf{30}$ groups, and the forward horizon is $H = \mathbf{30}$ trading days. The loop iterates for $g = w$–$1, \ldots, G$–1:

1. **Eligibility (visibility constraint).** The engine computes forward log returns $r_u^{(H)}$ for all groups u using bars strictly after each group's end time ("`start_from=next`"). At decision g, the training set includes only those historical groups in $\{g-w+1, \ldots, g\}$ whose H-day future is fully observed by just after e_g. The current group g is excluded from training.

2. **Clustering on eligible history.** The clustering step runs Agglomerative Clustering (complete linkage, precomputed distances) on the eligible submatrix. The procedure

yields $n_c = \mathbf{3}$ clusters. The prototype step computes each cluster's *medoid* as the member with the smallest row-sum distance.

3. **Semantic mapping (per-step).** The scoring step uses the eligible groups' forward returns $\{r_u^{(H)}\}$. The procedure requires **min_count**$=$ **10** eligible samples. The mapping assigns label 0 (bear) to the cluster with the lowest mean if its effect $< \tau_-$. The mapping assigns label 2 (bull) to the cluster with the highest mean if its effect $> \tau_+$. The thresholds are $(\tau_+, \tau_-) = (\mathbf{0}, \mathbf{0})$. All remaining clusters receive label 1 (neutral). No label is forced if thresholds are unmet.

4. **Label the current group only.** The assignment step maps G_g to the nearest medoid by D. The selected cluster is denoted $c^\star$. The algorithm sets $\text{dir\_label}_g \in \{0, 1, 2\}$ to the semantic label of $c^\star$. Earlier groups remain unchanged.

This loop re-estimates mappings strictly from history at every g. Consequently, the semantics adapt over time without leakage.

3.2.5 Daily Regime Series via Rolling Vote

Group labels live on irregular end times $\{e_g\}$. A majority vote produces a daily series z_t using the last $k = \mathbf{7}$ *completed* groups at each day t. A tie-breaking rule resolves ties by recency. A post-processing step merges consecutive days with the same label into non-overlapping spans for visualisation. An exported date-indexed daily series (values 0/1/2) then supports backtesting.

3.3 Lead–Lag Dependence

Algorithm 2: Rolling signature–based lead–lag matrix

```
Input   : Daily log returns $r_{i,t}$ for $N$ assets (winsorized at 2.5/97.5)
Params: Window $L$=30 (bdays); update spacing $f$=1; max lag $\ell_{\max}$=7; signature
        order $m$ (e.g. $m$=2); per-window feature normalisation; update dates $\mathcal{T}$
Output: Dated matrices $\{M_t\}_{t\in\mathcal{T}}$ with $M_t \in \mathbb{R}^{N\times N}$ and $M_t(i,i)$=0
for $t \in \mathcal{T}$ do
    Define the window $W_t = \{t-L+1,\dots,t\}$
    for each ordered pair $(i,j)$ with $i \neq j$ do
        Initialise $s^\star \leftarrow 0$
        for $\ell = 1$ to $\ell_{\max}$ do
            Build aligned sequences on $W_t$:
            $x_i^{(t,\ell)} = (r_{i,t-L+1},\dots,r_{i,t-\ell})$,
            $y_j^{(t,\ell)} = (r_{j,t-L+1+\ell},\dots,r_{j,t})$
            Map to normalised signature features (order $m$):
            $\hat{\phi}_i \leftarrow \Phi_m(x_i^{(t,\ell)})/\|\Phi_m(\cdot)\|$,
            $\hat{\phi}_j \leftarrow \Phi_m(y_j^{(t,\ell)})/\|\Phi_m(\cdot)\|$
            Form the reverse ordering (swap $i,j$) to get $\hat{\phi}'_i, \hat{\phi}'_j$
            Directional score:
            $s_{ij}^{(\ell)}(t) \leftarrow \langle\hat{\phi}_i,\hat{\phi}_j\rangle - \langle\hat{\phi}'_j,\hat{\phi}'_i\rangle$
            If $|s_{ij}^{(\ell)}(t)| > |s^\star|$, update $s^\star \leftarrow s_{ij}^{(\ell)}(t)$
        Set $M_t(i,j) \leftarrow s^\star$
    Set $M_t(i,i) \leftarrow 0$ for all $i$
return $\{M_t\}_{t\in\mathcal{T}}$
```

3.3.1 Data and Pre-processing

This subsection defines the sample. The dataset contains daily close prices for $N = 72$ assets from January 1, 2021 to June 30, 2024. Table A.1 lists the universe and basic descriptors. The source records end-of-day closes. Therefore, the analysis uses close prices as the reference level. The pipeline converts all timestamps to a single time zone. It also removes duplicate timestamps before alignment.

The close price of asset $i \in \mathcal{U}$ on day t is defined by $P_{i,t}$ and the log return calculated by $r_{i,t} = \ln P_{i,t} - \ln P_{i,t-1}$. The analysis then aligns the panel $P = \{P_{i,t}\}$ on a daily calendar. It retains only assets with sufficient coverage. The algorithm skips a pair when either shifted sequence has fewer than $L - \ell$ valid observations. The portfolio step also omits asset i temporarily when row i has fewer than q valid links, so that ranks remain stable. Finally, the pre-processing winsorises returns at the 2.5/97.5 percentiles, which reduces level effects.

3.3.2 Signature-based Lead–Lag Matrix

The analysis uses a rolling window and a maximum forward lag to set the comparison frame. It first sets $W_t = \{t - L + 1, \ldots, t\}$ and denotes the maximum lag by $\ell_{\max}$. Under this frame, the algorithm builds a directed matrix $M_t \in \mathbb{R}^{N \times N}$. It relies on a *path-signature* (sequence-kernel) score. The path signature captures order information that linear correlation ignores. Therefore, the analysis adopts it as the primary measure of direction. For clarity, the analysis writes $r_{i,\tau}$ for the winsorised log return of asset i on day τ.

The procedure defines two sequences for each ordered pair (i, j) and for each lag $\ell \in \{1, \ldots, \ell_{\max}\}$. It sets

$$x_i^{(t,\ell)} = \big(r_{i,t-L+1}, \ldots, r_{i,t-\ell}\big), \qquad y_j^{(t,\ell)} = \big(r_{j,t-L+1+\ell}, \ldots, r_{j,t}\big).$$

This alignment pairs the *past* of i with the *future* of j shifted by ℓ days. Therefore, any detected dependence may indicate that i *leads* j.

The mapping sends each sequence to a truncated signature vector $\Phi_m(\cdot) \in \mathbb{R}^{D(m)}$ up to degree m (e.g., $m = 2$ or 3). The analysis normalises features within each window and sets $\widehat{\Phi}_m(z) = \Phi_m(z)/\|\Phi_m(z)\|$. This step removes scale effects. It also preserves temporal order through iterated increments. Hence, the features support consistent comparisons across assets and lags.

The algorithm then computes an *antisymmetric* directional similarity for each ℓ. It contrasts the forward ordering with the reverse ordering:

$$s_{ij}^{(\ell)}(t) \;=\; \underbrace{\big\langle \widehat{\Phi}_m\big(x_i^{(t,\ell)}\big),\, \widehat{\Phi}_m\big(y_j^{(t,\ell)}\big) \big\rangle}_{\text{“}i\text{ then }j\text{”}} \;-\; \underbrace{\big\langle \widehat{\Phi}_m\big(x_j^{(t,\ell)}\big),\, \widehat{\Phi}_m\big(y_i^{(t,\ell)}\big) \big\rangle}_{\text{“}j\text{ then }i\text{”}}.$$

The first term measures how well the history of i explains the future of j. The second term measures the reverse. Therefore, their difference isolates directionality and equals zero under perfect symmetry.

The selection step chooses the lag with the largest absolute score. It sets

$$\ell_{ij}^{\star}(t) = \arg \max_{1 \le \ell \le \ell_{\max}} \big|s_{ij}^{(\ell)}(t)\big|, \qquad M_t(i, j) = s_{ij}^{(\ell_{ij}^{\star}(t))}(t).$$

The convention sets $M_t(i, i) = 0$. When shifting yields insufficient overlap, the algorithm writes the entry as NaN and the downstream pipeline ignores it. Consequently, the matrix M_t is directed. When $M_t(i, j) > 0$, asset i *leads* asset j with same-sign co-movement. When $M_t(i, j) < 0$, asset i leads asset j with opposite-sign co-movement.

The final paragraph reports hyperparameters and ranges. The implementation uses a window length of $L = \mathbf{30}$ business days, an update frequency of $f = \mathbf{1}$ business day, and a maximum forward lag of $\ell_{\max} = \mathbf{7}$. Because per-window normalisation enforces $\|\widehat{\Phi}_m(\cdot)\| = 1$, each inner product lies in $[-1, 1]$. This bound implies $s_{ij}^{(\ell)}(t) \in [-2, 2]$ and thus $M_t(i, j) \in [-2, 2]$. As a robustness check, the analysis can clip M_t to $[-1, 1]$ by setting $M_t \leftarrow \frac{1}{2} M_t$. This clipping reduces the impact of extreme windows.

3.4 Strategy

We construct portfolios from the rolling lead–lag matrices $\{M_t\}$ described earlier. All decisions at update date t use only in-window information and are implemented on the next trading day t^+ to avoid look-ahead.

3.4.1 Baseline Lead–Lag Hedge

```
Algorithm 3: Baseline lead–lag hedge (followers-long / leaders-short)
Input  : Rolling matrices $\{M_t\}$; daily returns $\{r_{i,t}\}$ on a unified calendar
Params: Tail quantile $q \in (0, 0.5)$; min names $k$; update spacing $f$=1 (bdays)
Output: Daily strategy returns $\{R_{t^+}\}$ executed at the next trading day $t^+$
for each update date t (every f business days) do
    Compute per-asset score $s_t(i) \leftarrow \frac{1}{N-1} \sum_{j \neq i} M_t(i, j)$
    Define leaders $\mathcal{L}_t \leftarrow \{i : s_t(i) \geq Q_{1-q}(s_t)\}$ and followers
     $\mathcal{F}_t \leftarrow \{i : s_t(i) \leq Q_q(s_t)\}$
    If $\min(|\mathcal{L}_t|, |\mathcal{F}_t|) < k$, skip this update
    On $t^+$ set equal weights: $w_{t^+}(i) = 1/|\mathcal{F}_t|$ for $i \in \mathcal{F}_t$, $w_{t^+}(i) = -1/|\mathcal{L}_t|$ for $i \in \mathcal{L}_t$,
     else 0
    Realize return $R_{t^+} \leftarrow \sum_i w_{t^+}(i)\, r_{i,t^+}$    (optionally rescale to a target gross)
```

For each update date t, define a per-asset score by the row mean of the directed matrix

$$s_t(i) \;=\; \frac{1}{N-1} \sum_{\substack{j=1 \\ j \neq i}}^{N} M_t(i, j), \qquad i = 1, \ldots, N,$$

so that large positive $s_t(i)$ indicates that i tends to *lead* many others (a “leader”), while large negative values indicate a “follower”.

Fix a tail quantile $q \in (0, 0.5)$ and a minimum leg size $k \in \mathbb{N}$. Let

$$\mathcal{L}_t = \{\, i : \; s_t(i) \geq Q_{1-q}(s_t) \,\}, \qquad \mathcal{F}_t = \{\, i : \; s_t(i) \leq Q_q(s_t) \,\}.$$

If $\min\{|\mathcal{L}_t|, |\mathcal{F}_t|\} < k$, we skip trading at t. Otherwise, on t^+ we hold an equal-weight long–short hedge that is *long followers* and *short leaders*:

$$w_{t^+}(i) = \begin{cases} \dfrac{1}{|\mathcal{F}_t|}, & i \in \mathcal{F}_t, \\ -\dfrac{1}{|\mathcal{L}_t|}, & i \in \mathcal{L}_t, \\ 0, & \text{otherwise.} \end{cases}$$

The one-day portfolio return is

$$R_{t^+} \;=\; \sum_{i=1}^{N} w_{t^+}(i)\, r_{i,t^+} \;=\; \underbrace{\frac{1}{|\mathcal{F}_t|} \sum_{i \in \mathcal{F}_t} r_{i,t^+}}\_\text{followers} \;-\; \underbrace{\frac{1}{|\mathcal{L}_t|} \sum_{i \in \mathcal{L}_t} r_{i,t^+}}\_\text{leaders.}$$

Rebalancing follows the matrix update schedule (every f business days; $f = 1$ in our baseline).

3.4.2 Regime–Aware Lead–Lag Overlay

Algorithm 4: Regime–aware overlay (anchor BTC, preserve hedge)

```
Input   : Rolling matrices $\{M_t\}$; daily returns $\{r_{i,t}\}$; regime labels $z_t \in \{0,1,2\}$
          (bear, neutral, bull)
Params: Same $q, k, f$ as baseline; anchor $a = $ BTC; orientation=row
          $\Rightarrow$ $\mathrm{rel}_t(i)=M_t(a,i)$; neutral scale $\alpha$=0.5; apply-to-leaders=on;
          preserve-baseline-hedge=on
Output: Daily overlay returns $\{R^{\mathrm{overlay}}_{t+}\}$
for each update date $t$ do
    Compute $s_t(i)$ and sets $\mathcal{L}_t, \mathcal{F}_t$ as in Alg. 3; if either $< k$, skip
    For tradable $i \neq a$, compute anchor relation $\mathrm{rel}_t(i) \leftarrow M_t(a,i)$ (or $M_t(i,a)$ under
      column convention)
    Partition by sign:
    $\mathcal{F}^+_t = \{i \in \mathcal{F}_t : \mathrm{rel}_t(i) > 0\}$,
    $\mathcal{F}^-_t = \{i \in \mathcal{F}_t : \mathrm{rel}_t(i) < 0\}$,
    $\mathcal{L}^+_t = \{i \in \mathcal{L}_t : \mathrm{rel}_t(i) > 0\}$,
    $\mathcal{L}^-_t = \{i \in \mathcal{L}_t : \mathrm{rel}_t(i) < 0\}$
    Form baskets by regime:
    Bull ($z_t$=2): $\mathcal{B}^{\mathrm{long}} \leftarrow \mathcal{F}^+_t \cup \mathcal{L}^+_t$,   $\mathcal{B}^{\mathrm{short}} \leftarrow \mathcal{F}^-_t \cup \mathcal{L}^-_t$
    Bear ($z_t$=0): swap long/short
    Neutral ($z_t$=1): $\mathcal{B}^{\mathrm{long}} \leftarrow \mathcal{F}_t$, $\mathcal{B}^{\mathrm{short}} \leftarrow \mathcal{L}_t$
    Preserve hedge (on): $\mathcal{B}^{\mathrm{short}} \leftarrow \mathcal{B}^{\mathrm{short}} \cup \mathcal{L}_t$
    If $\min(|\mathcal{B}^{\mathrm{long}}|, |\mathcal{B}^{\mathrm{short}}|) < k$, skip this update
    On $t^+$ set equal weights within each leg and compute
    $R^{\mathrm{overlay}}_{t+} \leftarrow \overline{r}_{\mathcal{B}^{\mathrm{long}},t^+} - \overline{r}_{\mathcal{B}^{\mathrm{short}},t^+}$;   if $z_t$=1, set $R^{\mathrm{overlay}}_{t+} \leftarrow \alpha\, R^{\mathrm{overlay}}_{t+}$
```

We augment the baseline with a *regime detection* signal that modulates which names are long/short based on their *signed* lead–lag relation with a benchmark anchor (Bitcoin, denoted BTC). Let $z_t \in \{0,1,2\}$ encode the detected market state at date t (bear, neutral, bull). Let the *anchor relation* for asset i be

$$\mathrm{rel}_t(i) \;=\; M_t(\mathrm{BTC},\, i) \quad \text{or} \quad \mathrm{rel}_t(i) \;=\; M_t(i,\, \mathrm{BTC}),$$

depending on whether rows or columns encode “anchor leads asset” in the chosen convention. Intuitively, $\mathrm{rel}_t(i) > 0$ means i tends to move in the *same* direction after the anchor (co-moving follower), while $\mathrm{rel}_t(i) < 0$ indicates an *opposite*-sign relation.

Using the same leader/follower sets $\mathcal{L}_t, \mathcal{F}_t$ as in the baseline, we split each set by the sign of $\mathrm{rel}_t(i)$ and assign long/short according to the regime:

$$(\text{bull } z_t{=}2) : \begin{cases} \text{long } \{i \in \mathcal{F}_t : \mathrm{rel}_t(i) > 0\} \ \cup \ \big(\text{optionally } \{i \in \mathcal{L}_t : \mathrm{rel}_t(i) > 0\}\big), \\ \text{short } \{i \in \mathcal{F}_t : \mathrm{rel}_t(i) < 0\} \ \cup \ \big(\text{optionally } \{i \in \mathcal{L}_t : \mathrm{rel}_t(i) < 0\}\big), \end{cases}$$

$$(\text{bear } z_t{=}0) : \ \text{flip the above long/short assignments.}$$

We *fix* the neutral policy to **Scale** with coefficient $\alpha = \mathbf{0.5}$: whenever $z_t = 1$ we down-weight the regime overlay by

$$R^{\text{overlay}}_{t^+} \ \leftarrow \ \alpha \, R^{\text{overlay}}_{t^+} \quad \text{with } \alpha = 0.5,$$

while $z_t \in \{0, 2\}$ uses full weight.

Positions are set on t^+; if either the long or short basket has fewer than $k = 2$ names after sign-splitting (and optional strength filtering), we skip trading at t. All weights are equal within each basket; exposure can be normalized to target a fixed gross if desired.

Our overlay preserves the baseline economic intuition. The leaders act first. The followers adjust after a delay. In addition, the overlay conditions on the market state through an anchor asset.

Chapter 4

Results and Evaluation

We use 2021-01-01–2024-06-30 as the data-availability window. The *effective* evaluation start date t_0 is configuration-dependent due to warm-up requirements in both components. For each configuration we begin evaluation at its first tradable day t_0 (signals on t, execution on t^+). Annualisation uses factor 365, costs are reported separately, and the risk-free rate is set to zero.

4.1 Headline Performance

Table 4.1 reports headline performance of the lead–lag hedge. The overlay keeps that hedge and adds regime awareness. The risk–free rate is zero. The overlay delivers a stronger risk–return trade–off. Cumulative return rises from **59.53%** to **132.78%**. CAGR increases from **16.28%** to **31.38%** (see Figure 4.1). Meanwhile, annualised volatility falls from **38.09%** to **20.06%**. As a result, Sharpe improves from **0.58** to **1.46**, and Sortino from **0.90** to **2.46**. Drawdowns also ease: the maximum drawdown tightens from **–35.32%** to **–14.15%**. Importantly, time in market is identical at **43%**. So the gains come from better selection and positioning, not from higher exposure. Final NAV rises from **1.5953** to **2.3278**. In short, the overlay earns more with materially less risk at the same participation rate.

Figure 4.2 plots the rolling 6-month Sharpe. The overlay spends more time above 1 and shortens negative spells around 2022, consistent with its faster exit and recovery.

Table 4.2 details the drawdown. The overlay cuts the peak–to–trough loss from **–35.32%** to **–14.15%**. It also halves time under water from **495** to **238** days. The baseline's worst episode is long and persistent (**2021-09-10** to **2023-01-17**; trough on **2021-12-03**) and spans the 2022 downturn. The overlay's worst spell starts later (**2021-11-24**), bottoms quickly (**2022-01-05**), and ends by mid-**2022** (**2022-07-19**). In other words, it absorbs

Table 4.1: Headline performance.

Metric	Baseline	Overlay (preserves hedge)
Cumulative Return (%)	59.53	132.78
CAGR (%)	16.28	31.38
Volatility (ann., %)	38.09	20.06
Sharpe	0.58	1.46
Sortino	0.90	2.46
Max Drawdown (%)	-35.32	-14.15
Time in Market (%)	43.0	43.0
Final NAV	1.5953	2.3278
Active Days (approx.)	549	549

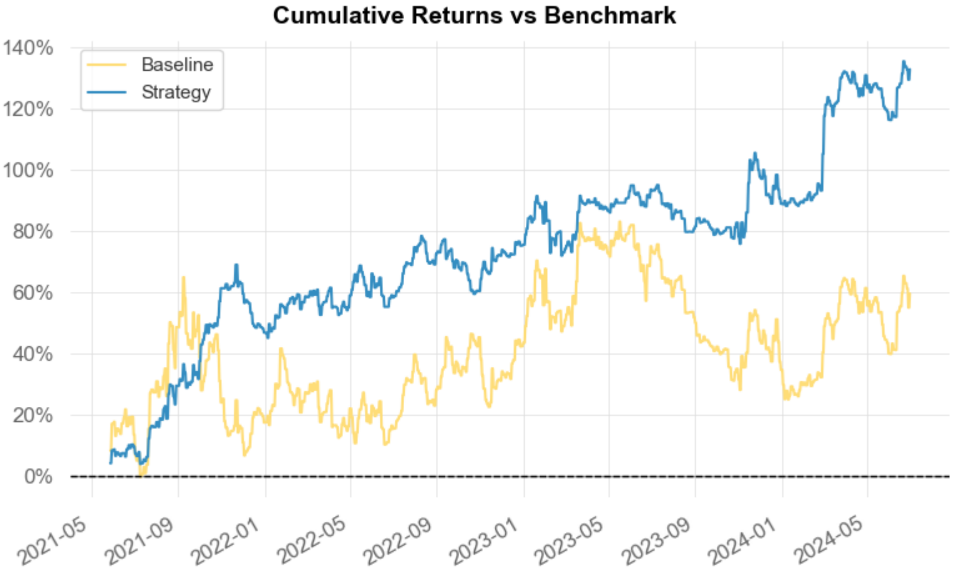

Figure 4.1: Cumulative NAV: baseline vs. regime overlay.

Table 4.2: Drawdown details.

Metric	Baseline	Overlay
Max DD (%)	-35.32	-14.15
Max DD Date	2021-12-03	2022-01-05
Max DD Period Start	2021-09-10	2021-11-24
Max DD Period End	2023-01-17	2022-07-19
Longest DD Days	495	238

the early-2022 selloff but exits and recovers faster. Net effect: a softer left tail and a shorter recovery horizon, consistent with the higher risk-adjusted performance.

Table 4.3 compares year-end returns and the multiplier (Strategy/Baseline). The overlay wins in **three of four** years—**2021**, **2023**, and **2024 (YTD)**—with multipliers of **2.87**, **-1.44**, and **1.26**. The negative multiplier in 2023 signals a sign flip (baseline −5.42% vs. strategy +7.79%). In **2022**, the multiplier is **0.91**, a mild shortfall. Overall, the overlay adds value when the baseline struggles and still provides an edge in rising markets.

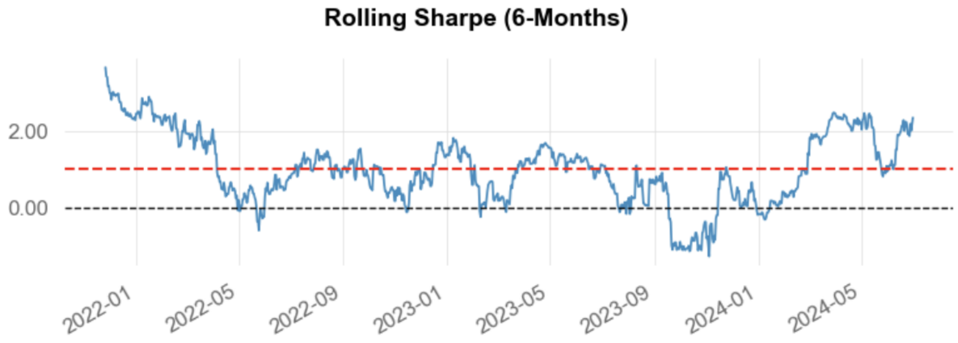

Figure 4.2: Rolling 6-month Sharpe for the baseline and the Overlay.

Table 4.3: EOY returns vs. baseline. Multiplier = Strategy / Baseline; negative means opposite signs.

Year	Baseline	Strategy	Multiplier	Won
2021	16.28%	46.77%	2.87	+
2022	21.22%	19.36%	0.91	−
2023	−5.42%	7.79%	−1.44	+
2024	18.54%	23.27%	1.26	+

Table 4.4: Top-5 by Sharpe compared with baseline strategy.

Strategy (params)	Sharpe	CAGR	Vol (ann)	MaxDD	Calmar	Alpha (ann)	WinRate
$n\_steps, n\_paths = 5, 20$	1.460	0.314	0.201	-0.141	2.161	0.212	0.215
$n\_steps, n\_paths = 5, 16$	1.006	0.217	0.218	-0.207	1.046	0.126	0.216
$n\_steps, n\_paths = 15, 12$	0.892	0.150	0.173	-0.173	0.863	0.112	0.155
$n\_steps, n\_paths = 15, 16$	0.853	0.154	0.188	-0.155	0.994	0.100	0.157
$n\_steps, n\_paths = 5, 8$	0.816	0.161	0.210	-0.192	0.841	0.095	0.214
Baseline	0.580	0.163	0.380	-0.353	0.440	0.000	0.219

4.2 Strategy selection and performance

First, we evaluate candidate parameterisations in a *strict walk-forward* setup, train only on past windows and label only the current group. Then we rank each configuration by Sharpe. For context, we also report the Baseline. Throughout, we use a unified daily calendar with a crypto annualisation factor of 365. We align returns from the *signal day* to the *next trading day*. Rates are in decimals (e.g., 0.20 = 20%). See Table 4.4.

Short windows with more paths perform best. In particular, $n_{\text{steps}} = 5$ with $n_{\text{paths}} \in \{16, 20\}$ leads. The top configuration, $n_{\text{steps}} = 5, n_{\text{paths}} = 20$, reaches a Sharpe of 1.44 with annualised volatility of 0.20, maximum drawdown −0.14, and Calmar 2.16. By contrast, the Baseline records a Sharpe of 0.56 and a Calmar of 0.44. Meanwhile, mid-range windows—such as $n_{\text{steps}} = 15$ with $n_{\text{paths}} \in \{12, 16\}$—also offer a solid return-risk trade-off.

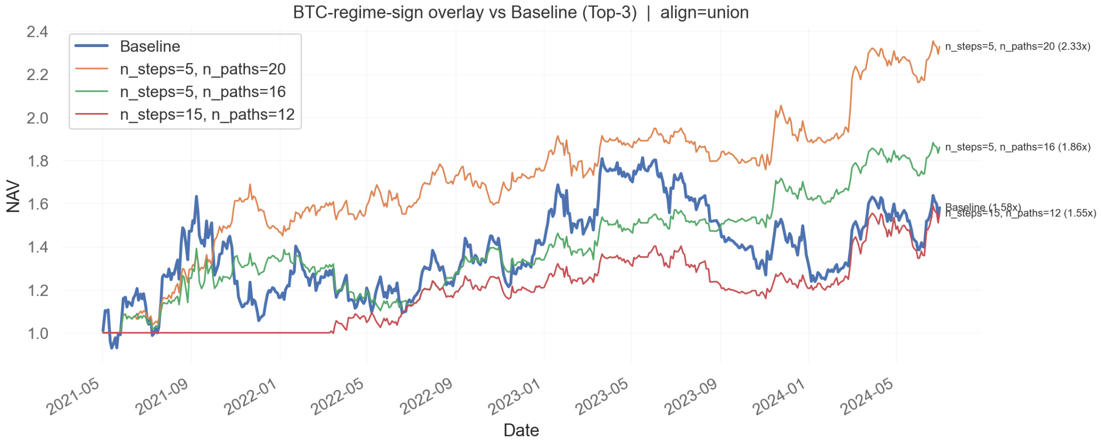

FIGURE 4.3: Top-3 NAV vs. Baseline (union-aligned). Values annotated at the last date indicate final NAV multiples.

TABLE 4.5: Significance vs. zero: NW t on daily mean excess, and 95% block-bootstrap CIs. Rates in decimals (e.g., 0.20 = 20%).

Strategy	NW t (mean)	AnnRet (pt)	AnnRet [2.5%,97.5%]	Sharpe (pt)	Sharpe [2.5%,97.5%]
Baseline	1.29	0.24	[−0.18, 0.90]	0.58	[−0.35, 1.80]
Overlay (5,20)	2.67	0.33	[0.08, 0.64]	1.46	[0.48, 2.48]
Overlay (5,16)	2.01	0.24	[0.01, 0.54]	1.00	[0.13, 2.10]
Overlay (15,12)	1.58	0.15	[−0.05, 0.40]	0.89	[−0.20, 1.96]
Overlay (15,16)	1.54	0.16	[−0.05, 0.43]	0.85	[−0.18, 1.88]
Overlay (5,8)	1.53	0.17	[−0.07, 0.47]	0.81	[−0.25, 1.90]

We assess significance against zero using Newey–West (HAC, Bartlett kernel; data-driven lag) t-statistics on daily excess returns (See Appendix B.6.7). We then build 95% moving-block bootstrap confidence intervals (block = 10 days, $B = 2000$) for annualized return and Sharpe on the unified calendar. Table 4.5 reports the Baseline and the top five overlays by Sharpe. As shown, *Overlay (5,20)* is significant versus zero; both confidence intervals lie strictly above zero. *Overlay (5,16)* is marginally significant. By contrast, the Baseline and the remaining overlays are not significant at the 5% level, with intervals overlapping zero.

Several other settings (e.g., $n_{\text{steps}} \in \{8, 10\}$) are mediocre or negative in this sample and are omitted. In practice, implementation should also account for switching frequency and transaction costs (see the switch-rate diagnostics), since these frictions affect net returns and capacity. Overall, the walk-forward state signal, when combined with our overlay, improves risk-adjusted performance, with "shorter steps + more paths" as the most effective pattern in this dataset.

Table 4.6: Baseline transaction-cost sensitivity (per side, applied to target-weight changes).

Cost (bps/side)	Sharpe	AnnRet (%)	Vol (%)	MaxDD (%)
5	0.323	5.07	40.98	-42.36
10	-0.015	-8.56	40.99	-58.57
25	-1.026	-39.77	41.15	-86.30

Table 4.7: Transaction-cost sensitivity. Costs are per side, applied to target-weight changes.

Strategy (params)	Sharpe	AnnRet (%)	Vol (%)	MaxDD (%)
Cost = 5 bps/side				
$n_{\text{steps}} = 5,\ n_{\text{paths}} = 20$	1.052	26.82	25.70	-22.98
$n_{\text{steps}} = 5,\ n_{\text{paths}} = 16$	0.628	14.24	26.86	-34.98
$n_{\text{steps}} = 15,\ n_{\text{paths}} = 12$	0.475	9.45	26.12	-31.18
$n_{\text{steps}} = 15,\ n_{\text{paths}} = 16$	0.561	11.88	25.96	-27.17
$n_{\text{steps}} = 5,\ n_{\text{paths}} = 8$	0.485	9.81	26.50	-26.90
Cost = 10 bps/side				
$n_{\text{steps}} = 5,\ n_{\text{paths}} = 20$	0.672	15.03	25.68	-32.08
$n_{\text{steps}} = 5,\ n_{\text{paths}} = 16$	0.289	4.28	26.82	-39.22
$n_{\text{steps}} = 15,\ n_{\text{paths}} = 12$	0.122	-0.19	26.13	-36.51
$n_{\text{steps}} = 15,\ n_{\text{paths}} = 16$	0.197	1.78	25.96	-33.31
$n_{\text{steps}} = 5,\ n_{\text{paths}} = 8$	0.103	-0.76	26.50	-34.76
Cost = 25 bps/side				
$n_{\text{steps}} = 5,\ n_{\text{paths}} = 20$	-0.466	-14.21	25.79	-65.20
$n_{\text{steps}} = 5,\ n_{\text{paths}} = 16$	-0.732	-20.73	26.84	-63.57
$n_{\text{steps}} = 15,\ n_{\text{paths}} = 12$	-0.931	-24.35	26.28	-68.04
$n_{\text{steps}} = 15,\ n_{\text{paths}} = 16$	-0.892	-23.39	26.07	-66.77
$n_{\text{steps}} = 5,\ n_{\text{paths}} = 8$	-1.038	-26.80	26.62	-68.02

4.3 Transaction-cost modeling and impact

We model transaction costs in a simple, transparent way that matches our portfolio construction and evaluation. First, on each trading day t we form target portfolio weights w_t (long equal-weight followers; short equal-weight leaders). Next, we proxy trading volume with the absolute change in target weights.

$$\text{turnover}_t = \sum_i |w_{t,i} - w_{t-1,i}|.$$

We then apply a per-side fee of c basis points to this turnover, which yields the daily cost

$$\text{cost}_t = (c \times 10^{-4})\ \text{turnover}_t.$$

Consequently, we compute net returns in *simple-return* space as $r_t^{\text{net}} = r_t^{\text{gross}} - \text{cost}_t$. We charge for the first trade (initial portfolio formation) and use the *target-diff* mechanic.

Costs erode performance roughly in line with annualised turnover. For the Baseline

(Table 4.6), sharpe drops from 0.58 at 0 bps to 0.32 at 5 bps, and turns negative by 10 bps. Therefore, the break-even lies between 5 and 10 bps under the *target-diff* mechanic. By contrast, the best overlay configurations (Table 4.7) suffer a milder hit thanks to lower average daily turnover ($\sim$ 1.16–1.30 vs. Baseline 1.77). For example, $n_{\text{steps}} = 5, n_{\text{paths}} = 20$ remains attractive at 5–10 bps (Sharpe 1.05 → 0.67), yet performance compresses sharply beyond 25 bps. As expected, volatility is largely unchanged across bps grids, while maximum drawdowns deepen as net returns are shaved each day. At high frictions, all configurations become economically unviable, and NAVs converge toward unity.

Chapter 5

Discussion

This chapter interprets the evidence rather than re-reporting it. We explain what the regime-aware overlay changes relative to the lead–lag hedge, why those changes plausibly work, and how robust and implementable they look. We then state the main limitations and sketch next steps. Conventions and headline metrics appear in Section 4.1; we refer to them without repeating details.

5.1 Interpreting the findings

The overlay improves return per unit risk while *holding participation constant* (Table 4.1). Therefore, the gain comes from selection and timing, not leverage. Figure 4.1 shows a smoother equity curve; Table 4.2 shows a softer left tail and faster recovery. Figure 4.2 adds the dynamic view: the overlay spends more time above a 6-month Sharpe of 1 and shortens the negative pockets around 2022. In calendar time (Table 4.3), the overlay helps most when the baseline is misaligned or when regime shifts are sharp; when the baseline is already onside, marginal gains shrink.

Why does this happen? The baseline is a fixed long-followers/short-leaders hedge. The overlay routes that hedge through state labels linked to BTC path geometry. In favourable states it tilts toward assets that propagate the anchor; in adverse states it flips the sign while *preserving* the hedge. Gross exposure stays similar, but composition rotates with the state. This reduces whipsaws and trims losses early when conditions turn.

5.2 Sensitivity and implementability

Across strict walk-forward runs, short windows with more paths dominate (Table 4.4). In particular, $n_{\text{steps}}=5$ with $n_{\text{paths}} \in \{16, 20\}$ balances sensitivity and stability; mid-length settings improve on the baseline but with lower Sharpe. This pattern is consistent with a bias–variance trade-off: long windows blur turning points, while too few paths raise variance in the distance estimates.

Statistical checks support a cautious reading. Against zero, the top configuration is significant and its block-bootstrap confidence intervals exclude zero; the next is marginal. However, after White's Reality Check (WRC) and Hansen's SPA (SPA), we cannot reject no outperformance relative to the baseline (overall $p \approx 0.66$; see Table A.2). Hence we emphasise absolute improvement and drawdown control, not a guaranteed beat of the baseline.

Finally, implementability hinges on turnover. Costs erode performance roughly in proportion to annualised turnover. The baseline breaks even between 5–10 bps/side (Table 4.6). The best overlay remains attractive at 5–10 bps because average turnover is lower, but performance compresses beyond 25 bps (Table 4.7). Simple throttles help in practice: skip small basket changes, increase rebalance spacing in high-noise states, and cap participation relative to liquidity.

5.3 Limitations and next steps

Scope comes first. The sample is daily and limited to 2021-01-01–2024-06-30 with a crypto-centric anchor; external validity to other regimes or asset classes is not guaranteed. Second, the single-anchor design risks omitted-state bias if leadership rotates outside BTC; multi-anchor or hierarchical states are a natural extension. Third, several hyperparameters are fixed (kernel scale, order, regularisation). Mis-scaling can over- or under-smooth geometry, and unbiased MMD raises variance in small training windows.

Label construction also matters. Voting over the last k groups reduces jitter but adds lag; close calls near turning points can flip. Lead–lag compression to a row-mean discards pairwise uncertainty and can over-reward near-ties in lag choice. Data handling imposes further constraints: winsorisation trims extremes; funding/borrow and market impact are simplified; reported headline metrics are gross of costs unless stated.

Finally, inference carries model-selection risk. Hyperparameters were chosen plausibly rather than via nested, walk-forward cross-validation. Multiple testing remains a threat despite our compact grid; WRC/SPA results underscore that caution. Next steps follow directly: extend to *multi-anchor* regimes; add *uncertainty-aware* scaling that gates gross exposure by state confidence; incorporate execution-aware controls (impact, latency, borrow/funding); and formalise selection with step-down SPA or a model-confidence set.

Chapter 6

Conclusion

This dissertation studied how *path-wise* information extracted by signature methods can detect market regimes in a strict walk–forward setting and improve a signature-based lead–lag hedge. We started from daily BTC data from 2021-01-01 to2024-06-30 and built backward-looking regime labels with a signature–kernel MMD on rolling path groups and combined them with a rolling signature lead–lag matrix across assets. We then overlaid the baseline hedge (long followers / short leaders) with state-conditioned basket selection keyed to the BTC anchor, while preserving the hedge structure.

The overlay raises return per unit risk at the same market participation. Cumulative return increases from 59.53% to 132.78%. Sharpe rises from 0.58 to 1.46 and Sortino from 0.90 to 2.46, while annualised volatility falls from 38.09% to 20.06% (Table 4.1). Drawdowns are shallower and shorter: the maximum improves from –35.32% to –14.15%, and underwater time halves from 495 to 238 days (Table 4.2). Year-by-year results show three wins out of four (2021, 2023, 2024 YTD) and a mild shortfall in 2022 (Table 4.3). Across strict walk–forward runs, shorter windows with more paths perform best; $n_{\text{steps}}{=}5$ with $n_{\text{paths}} \in \{16, 20\}$ dominates neighbouring settings (Table 4.4). Statistical checks are consistent with a cautious reading: the top configuration is significant versus zero and its block-bootstrap CIs exclude zero (Table 4.5), whereas we do not reject no outperformance relative to the baseline (Table A.2). Costs matter but do not erase the edge at realistic frictions: the baseline breaks even around 5–10 bps/side; the top overlay remains attractive at 5–10 bps and degrades beyond 25 bps (Tables 4.6–4.7).

The mechanism is relatively simple. The overlay does not lever up, extend holding time, or chase beta. It *routes* the same hedge through regime information. In bull states it favours names that co-move with the anchor; in bear states it flips signs; in neutral states it scales. Gross exposure stays similar, but composition rotates with the state. This reduces whipsaws, trims left-tail losses, and speeds recovery.

Methodologically and empirically, the work contributes four elements: (i) a strict walk–forward regime detector based on signature–kernel MMD with medoid mapping to bull/neutral/bear labels; (ii) a signature-based lead–lag matrix that captures directional, order-sensitive dependence and a transparent baseline hedge; (iii) a *regime-aware overlay* that preserves the hedge while gating baskets by an anchor-signed relation; and (iv) a reproducible evaluation protocol with unified calendar alignment, signal-to-execution timing, turnover-based costs, and diagnostics for drawdowns, yearly performance, and parameter sensitivity.

The conclusions are bounded. The sample is daily and crypto-centric over 2021-01-01–2024-06-30; external validity to other periods or asset classes is not guaranteed. A single BTC anchor can miss leadership outside crypto. Several hyperparameters are fixed (kernel scale, order, regularisation), and unbiased MMD raises variance in small training windows. Rolling votes reduce label jitter but add lag near turning points. Row-mean leadership compresses pairwise uncertainty, and lag maximisation can overfit short windows. Frictions—fees, funding/borrow, impact—are modelled parsimoniously; net results depend on venue microstructure. Finally, model selection risk remains despite a compact grid; the WRC/SPA outcome underlines that caution.

Promising extensions follow naturally. Multi-anchor or hierarchical regimes could generalise the design beyond BTC. Probabilistic (soft) states would allow confidence-weighted scaling of exposure and trades. Cost-aware construction—turnover penalties, participation caps, and state-dependent rebalance spacing—could harden net performance. Kernel approximations and streaming clustering would improve scalability. Broader horizons (pre-2021 history, intraday data) would test temporal and frequency robustness.

In summary, path-wise regime information can be combined with directional lead–lag structure to produce a *regime-aware hedge* that is more resilient and capital efficient than its unconditional counterpart. The approach is modular: path features, regime mapping, and portfolio rules can be swapped independently. This makes the framework a practical template for bringing modern geometric time-series tools into risk-managed trading.

Appendix A

Supplementary Material

This appendix collects supplementary tables and plots.

Table A.1: Investable universe (72 assets): full names and tickers. The anchor asset is BTC (No. 61).

No.	Currency	Ticker	No.	Currency	Ticker	No.	Currency	Ticker
1	Oasis Network	`ROSE`	25	Tezos	`XTZ`	49	BNB	`BNB`
2	Ankr	`ANKR`	26	Loopring	`LRC`	50	Uniswap	`UNI`
3	VeChain	`VET`	27	Harmony	`ONE`	51	Stacks	`STX`
4	NEAR Protocol	`NEAR`	28	Solar	`SXP`	52	THORChain	`RUNE`
5	Ethereum	`ETH`	29	Kava	`KAVA`	53	Theta Network	`THETA`
6	EOS	`EOS`	30	Axie Infinity	`AXS`	54	Holo	`HOT`
7	BakeryToken	`BAKE`	31	Cardano	`ADA`	55	1inch	`1INCH`
8	The Graph	`GRT`	32	Solana	`SOL`	56	Fetch.ai	`FET`
9	Reef	`REEF`	33	Ontology	`ONT`	57	Kusama	`KSM`
10	Injective	`INJ`	34	Ethereum Classic	`ETC`	58	Smooth Love Potion	`SLP`
11	Filecoin	`FIL`	35	Decentraland	`MANA`	59	Curve DAO Token	`CRV`
12	Polygon	`MATIC`	36	Synthetix	`SNX`	60	IoTeX	`IOTX`
13	Bitcoin Cash	`BCH`	37	Zcash	`ZEC`	61	Bitcoin	`BTC`
14	IOST	`IOST`	38	Conflux	`CFX`	62	Avalanche	`AVAX`
15	Chromia	`CHR`	39	Yearn Finance	`YFI`	63	Enjin Coin	`ENJ`
16	MultiversX	`EGLD`	40	Waves	`WAVES`	64	PancakeSwap	`CAKE`
17	Hedera	`HBAR`	41	Litecoin	`LTC`	65	XRP	`XRP`
18	Zilliqa	`ZIL`	42	Chiliz	`CHZ`	66	TRON	`TRX`
19	Algorand	`ALGO`	43	Stellar	`XLM`	67	Cosmos	`ATOM`
20	Dent	`DENT`	44	COTI	`COTI`	68	Aave	`AAVE`
21	Dash	`DASH`	45	Polkadot	`DOT`	69	Dogecoin	`DOGE`
22	My Neighbor Alice	`ALICE`	46	OMG Network	`OMG`	70	Neo	`NEO`
23	IOTA	`IOTA`	47	SushiSwap	`SUSHI`	71	The Sandbox	`SAND`
24	Chainlink	`LINK`	48	Fantom	`FTM`	72	Qtum	`QTUM`

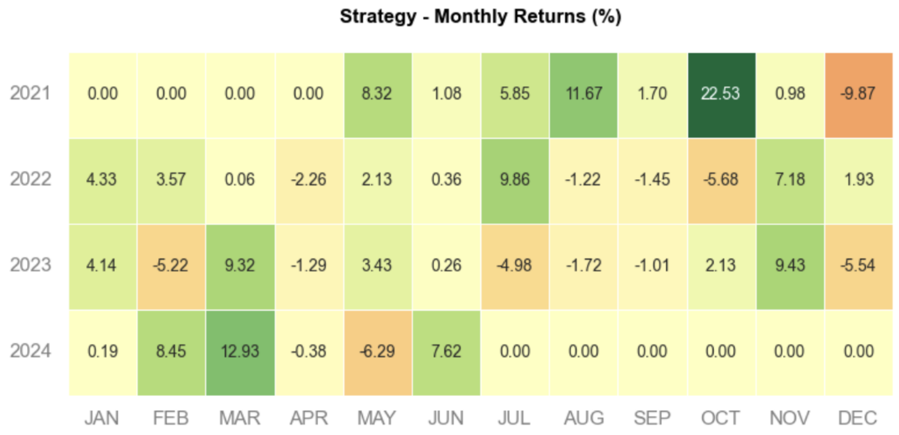

Figure A.1: Distribution of monthly returns for the Baseline and the Strategy. Bars use common bins; the smooth lines are kernel density estimates. The Strategy shifts the mass to the right and trims the left tail relative to the Baseline.

Table A.2: White's Reality Check (WRC) and Hansen's SPA (overall $p = 0.6645$). Mean differences are daily (overlay − baseline).

Strategy ($n_{\text{steps}}, n_{\text{paths}}$)	WRC mean_diff	WRC (bp/day)	SPA mean_diff	SPA std	Rank (WRC)
(5, 20)	0.000033	0.33	0.000033	0.015737	1
(5, 16)	-0.000140	-1.40	-0.000140	0.013985	2
(5, 8)	-0.000298	-2.98	-0.000298	0.015934	3
(15,16)	-0.000340	-3.40	-0.000340	0.017153	4
(15,12)	-0.000366	-3.66	-0.000366	0.018691	5

Notes: Positive mean_diff favours the overlay. "bp/day" = 10,000× mean_diff. SPA std is the standard deviation of the daily difference series.

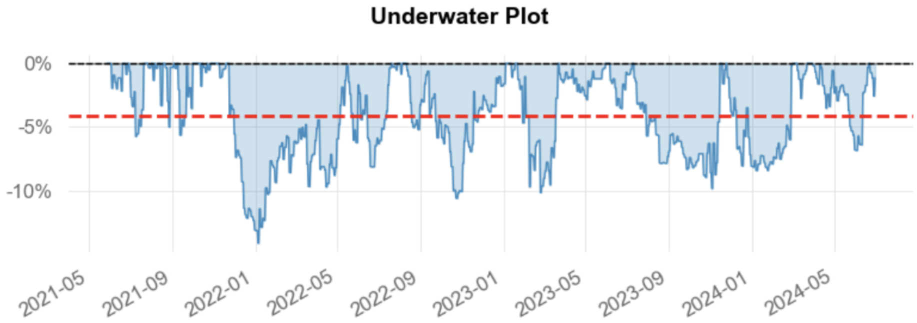

FIGURE A.2: Underwater plot (peak-to-trough drawdown over time) for the Strategy. The dashed line marks −5%. The series shows clustered losses in early 2022 and faster recoveries thereafter, consistent with the overlay's regime-aware routing.

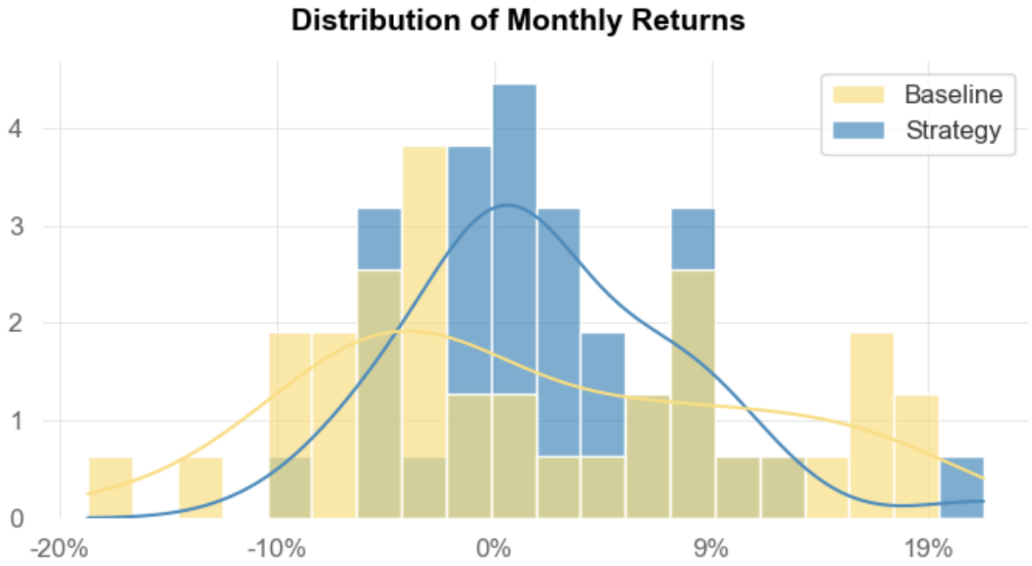

FIGURE A.3: Monthly returns (%) for the Strategy by calendar month. The heatmap highlights strong months in 2021Q4 and 2024Q1, as well as mixed outcomes during the 2022 drawdown.

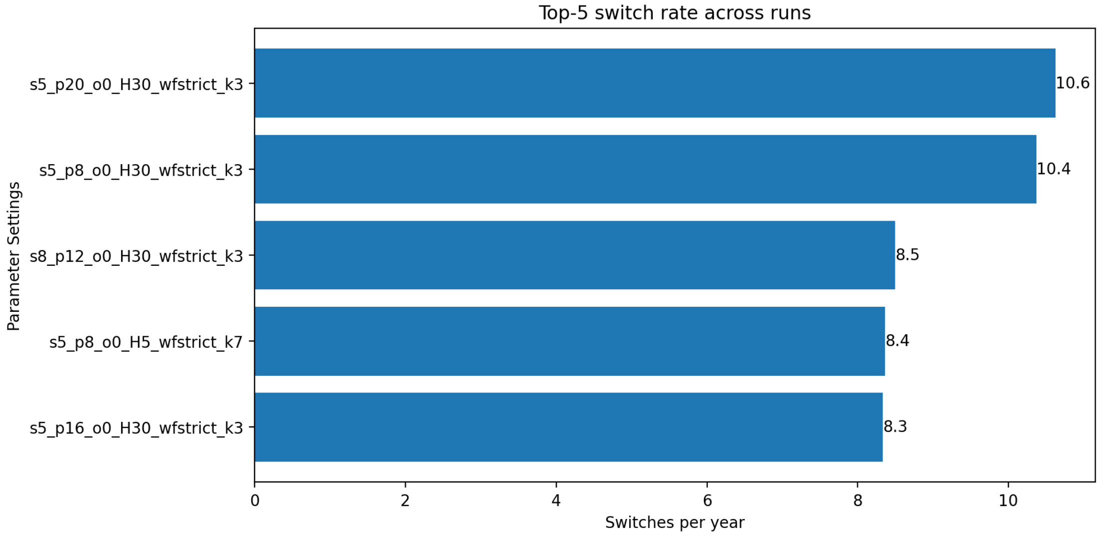

FIGURE A.4: Switch-rate diagnostics for the top-5 parameterisations. Bars show the average number of basket switches per calendar year (signal on t, execution on t^+). Higher switch rates imply higher turnover and stronger cost sensitivity.

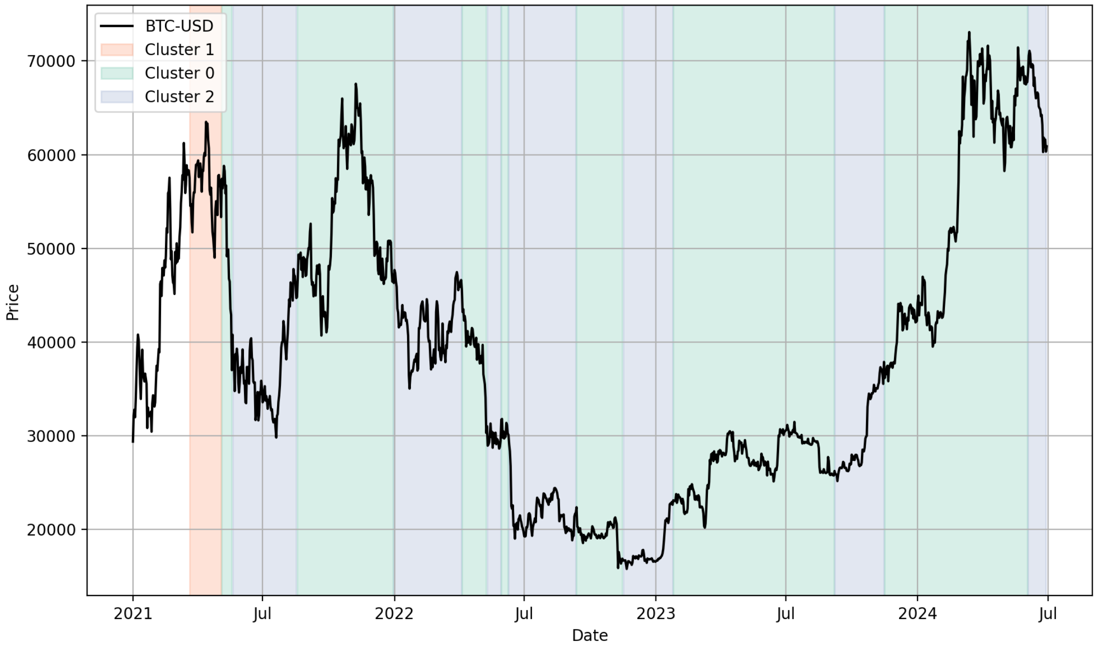

FIGURE A.5: BTC price with walk–forward regime shading (n_{steps}=5, n_{paths}=16; H=30, k=5, o=0).

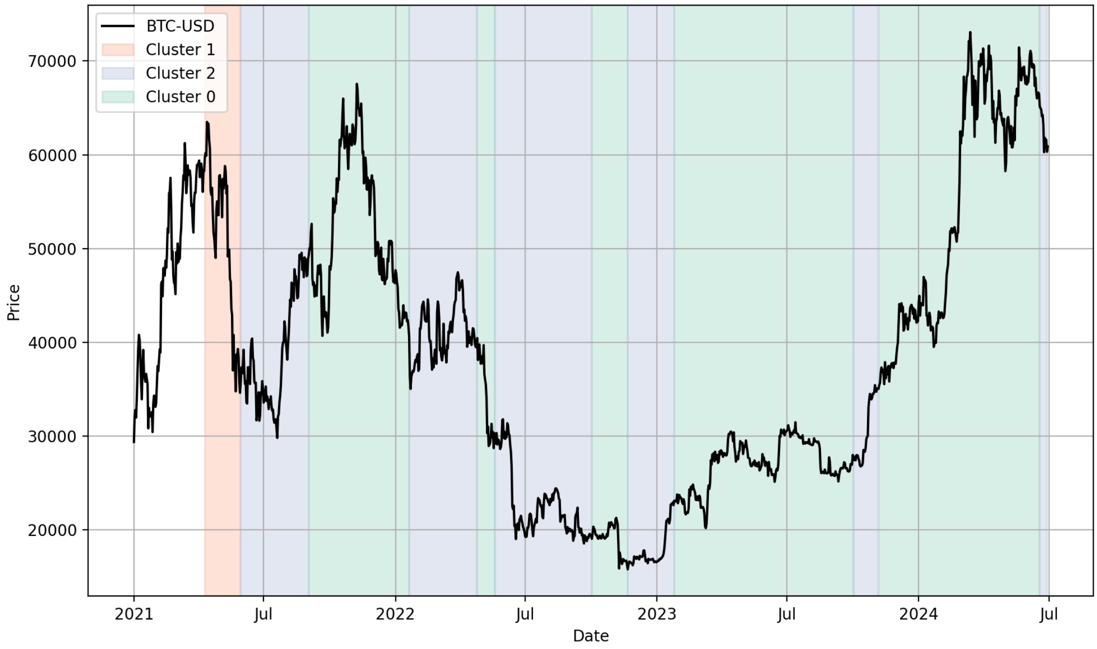

FIGURE A.6: BTC price with walk–forward regime shading (n_{steps}=5, n_{paths}=20; H=30, k=5, o=0). Increasing the number of paths stabilises the state sequence and shortens spurious flips.

Appendix B

Theoretical Appendix

B.1 Signature Features for Time Series

B.1.1 Definition of Path Signature

Consider a continuous path $X : [a, b] \to \mathbb{R}^d$; for example, a multivariate time series. The *signature* of X over $[a, b]$, written $S(X)_{a,b}$, is the (formal) infinite collection of iterated integrals on that interval. Formally, the signature lives in the tensor algebra $T((\mathbb{R}^d)) = \{1, \mathbb{R}^d, (\mathbb{R}^d)^{\otimes 2}, \dots\}$ and consists of:

- **Level 0:** a scalar 1 (by convention).
- **Level 1:** component-wise increments $\int_a^b dX_t^{(i)}$ for $i = 1, \dots, d$.
- **Level 2:** double iterated integrals $\int_a^b \int_a^{t_2} dX_{t_1}^{(i)} \otimes dX_{t_2}^{(j)}$.
- **Higher levels:** k-fold iterated integrals for $k = 3, 4, \dots$

In abbreviated form,

$$S(X)_{a,b} = \Big(1,\ \int_a^b dX_t,\ \int_a^b \int_a^{t_2} dX_{t_1} \otimes dX_{t_2},\ \dots\Big).$$

These integrals encode the shape of the path. Many readers view the signature as a "Taylor series for paths". It provides a principled expansion that captures ordered interactions in sequential data.

B.1.2 Truncation and Dimensionality

The full signature is infinite. In practice, we truncate at level m to obtain a finite vector. The truncated signature $S^{(m)}(X)_{a,b}$ lies in $\bigoplus_{k=0}^{m} (\mathbb{R}^d)^{\otimes k}$. The dimension grows as $1+d+d^2+\cdots+d^m$. Therefore, m trades information against computation. Low orders (for example, $m = 2$ or 3) often capture useful structure while keeping cost manageable.

B.1.3 Properties: Uniqueness and Chen's Identity

For broad classes of paths, the untruncated signature is essentially injective up to negligible equivalences. This motivates its use as a near-universal descriptor. A key algebraic rule is *Chen's identity.* If X on $[a, c]$ is split at b, then

$$S(X)_{a,c} \;=\; S\big(X^{(1)}\big)_{a,b} \otimes S\big(X^{(2)}\big)_{b,c}.$$

Signatures are invariant under time reparametrisation. Iterated integrals also satisfy shuffle relations. These properties enable efficient computation and clean algebra.

B.1.4 Motivation for Use in Time Series Modelling

Signatures provide high-capacity, order-sensitive features. First-order terms capture cumulative trends. Second-order terms capture pairwise lead–lag effects. Higher orders encode more complex interactions. Truncation then yields a compact and algebraically convenient representation for learning.

B.2 Normalisation and Scaling

B.2.1 Motivation in High Dimensions

High-dimensional features often sit on different scales. Without normalisation, distance-based methods can be dominated by large-scale coordinates. Moreover, gradient-based training may converge slowly. Normalisation improves numerical conditioning and interpretability by putting features on comparable footing.

B.2.2 Standard Normalisation Approaches

- ***z*-score normalisation:** $X' = (X - \mu_X)/\sigma_X$. After z-score normalisation, features have mean 0 and unit variance.
- **Min–max scaling:** $X' = \big(X - \min(X)\big)/\big(\max(X) - \min(X)\big)$ to map values to $[0, 1]$.
- **Rank/quantile normalisation:** map values to uniform or Gaussian quantiles. This is robust to outliers.
- **Robust scaling:** $(X - \mathrm{median}(X))/\mathrm{IQR}(X)$ to mitigate extreme values.

B.2.3 Impacts on Learning and Clustering

Normalisation speeds and stabilises gradient-based optimisation. It acts as a form of preconditioning. It also yields more sensible distance geometry for clustering. It is standard before PCA so that components reflect structure rather than scale.

B.3 Maximum Mean Discrepancy (MMD)

B.3.1 Kernel Mean Embeddings

Let an RKHS have kernel k and feature map ϕ. A distribution P embeds as $\mu_P := \mathbb{E}_{X\sim P}[\phi(X)]$. Hence, for any RKHS function f, we have $\mathbb{E}[f(X)] = \langle f, \mu_P \rangle$.

B.3.2 Definition of MMD

The MMD between P and Q is the RKHS distance between their mean embeddings:

$$\mathrm{MMD}_k(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}}.$$

Equivalently,

$$\mathrm{MMD}_k^2(P, Q) = \mathbb{E}_{X,X'} k(X, X') + \mathbb{E}_{Y,Y'} k(Y, Y') - 2\,\mathbb{E}_{X,Y} k(X, Y).$$

With characteristic kernels (for example, Gaussian), $\mathrm{MMD}_k(P, Q) = 0$ if and only if $P = Q$.

B.3.3 Two-Sample Testing with MMD

Suppose samples $\{x_i\}_{i=1}^n \sim P$ and $\{y_j\}_{j=1}^m \sim Q$. Unbiased quadratic-time estimators of MMD^2 enable consistent two-sample tests. One may calibrate by permutation or via asymptotic approximations.

B.3.4 Characteristic and Universal Kernels

A kernel is *characteristic* if $P \mapsto \mu_P$ is injective. Then $\mathrm{MMD}_k(P, Q) = 0$ implies $P = Q$. Gaussian and Laplace kernels are key examples. Universality implies characteristicness on compact domains. Non-characteristic kernels may collapse distinct distributions to the same mean embedding.

B.3.5 Interpretations and Theoretical Properties

MMD is both a feature-space norm and an integral probability metric. In an RKHS, the maximising witness function aligns with $\mu_P - \mu_Q$. Empirical MMD concentrates to its population value under standard conditions. Random features and related tricks can reduce computational cost. The metric is differentiable and widely used in modern learning tasks.

B.3.6 Summary

MMD provides a flexible way to compare distributions via kernels. With characteristic kernels, it detects any distributional difference in the limit. It supports hypothesis testing and representation learning with clear theoretical guarantees.

B.4 Lead–Lag Analysis

B.4.1 Temporal Ordering and Lead–Lag Relationships

A *lead–lag relationship* is directional. Series A *leads* series B if past values of A predict future values of B better than the reverse. The idea depends on temporal order, not just contemporaneous correlation.

B.4.2 Measuring Lead–Lag Dependencies

Cross-correlation offers a basic tool. Peaks at positive lags suggest "A leads B". Negative-lag peaks suggest the opposite. However, linear shifts may miss non-linear phase relations. Pathwise measures help here. The signed Lévy area between (A_t, B_t) captures direction. The sign indicates who leads. The magnitude reflects strength. These measures detect non-linear and phase-shifted behaviour beyond simple correlation.

B.4.3 Construction of Lead–Lag Matrices

With N series, we summarise all pairwise relations in an $N \times N$ matrix L. Entry L_{ij} quantifies how much i leads j. Usually $L_{ij} = -L_{ji}$ and $L_{ii} = 0$. We may build L from cross-correlation peaks, from estimated lags, or from normalised Lévy areas. Row means yield simple leadership scores. Large positive rows indicate leaders. Large negative rows indicate followers. We may also view L as a directed, weighted graph and then search for communities with coherent directionality.

B.4.4 Statistical Significance of Lead–Lag Effects

We should test for significance. Permutation or time-shift tests create null distributions. If an observed L_{ij} is extreme under the null, we treat it as significant. Frequency-domain methods also help. Phase leads in specific bands can reveal structured directionality.

B.4.5 Summary

Lead–lag analysis uncovers directed temporal dependence. Matrices or networks summarise the system. Scores and clustering then reveal leaders, followers, and group structure. The approach aids studies of causality and information flow in many domains.

B.5 Clustering Based on Distance Matrices

B.5.1 Pairwise Distance Matrices

Some clustering methods work only with pairwise dissimilarities. Given n objects, we compute an $n \times n$ distance matrix D with entries $D_{ij} = \mathrm{dist}(X_i, X_j)$. The distance may be any suitable metric or dissimilarity. Algorithms then use D directly. Coordinates are optional. Classical agglomerative hierarchical clustering can be phrased entirely in terms of D. We compute D, start with singletons, merge the closest clusters, update inter-cluster distances by a chosen linkage, and continue to one cluster.

B.5.2 Types of Distance Metrics

The metric matters.

- **Euclidean distance:** for vectors in $\mathbb{R}^d$, $D_{ij} = \|x_i - x_j\|_2$. It is intuitive, but may suffer from concentration in very high dimensions. Manhattan and other Minkowski variants are alternatives.
- **Kernel-induced distance:** if k is a positive-definite kernel,

$$D_{ij} = \sqrt{k(x_i, x_i) + k(x_j, x_j) - 2k(x_i, x_j)}\,,$$

  which equals the Euclidean distance between feature maps. This supports flexible, implicit embeddings.
- **Distributional distances (e.g. MMD):** when points are distributions or complex objects, we can define $D_{ij} = \mathrm{MMD}(P_i, P_j)$. Other choices include edit distance for strings, DTW for time series, and graph distances for networks.

B.5.3 Clustering Algorithms Using Distance Matrices

Two prominent approaches rely on distance or similarity matrices. They are hierarchical clustering and spectral clustering.

B.5.3.1 Hierarchical Clustering

Hierarchical clustering may be agglomerative (bottom–up) or divisive (top–down). In the agglomerative case, we begin with the full distance matrix. Each point is its own cluster. At each step, we merge the two closest clusters according to the current matrix. The cluster–cluster distance depends on the linkage choice. Common options are single, complete, and average linkage.

After each merge, we update the matrix. We remove the two cluster rows and columns. We add a row and column for the merged cluster. We compute its distances to others via the chosen linkage. This iterative scheme builds a dendrogram from n singletons to one all-encompassing cluster.

The matrix is central. It drives the first merge and all later merges. The output is a multi-scale structure. We cut the tree at a chosen level to obtain a flat partition with any desired number of clusters.

The method is deterministic given D and the linkage rule. It is useful when we expect nested groups or when the number of clusters is unknown. It also works with any valid distance. For instance, we can cluster texts with Jaccard distance or sequences with edit distance by supplying the appropriate matrix.

B.5.3.2 Spectral Clustering

Spectral clustering treats the task as graph partitioning. We start from an affinity (similarity) matrix A with entries A_{ij}. We may obtain A from distances using a kernel transform, for example

$$A_{ij} = \exp\Big(- \frac{\beta\, D_{ij}^2}{\sigma^2} \Big),$$

with scale parameters β and σ. This gives large weights to nearby points and near-zero weights to distant points. Normalisation often keeps affinities in a reasonable range.

We then build a weighted graph with nodes as points and edges weighted by A_{ij}. Next, we form the graph Laplacian $L = D_{\text{deg}} - A$, where $D_{\text{deg},ii} = \sum_j A_{ij}$. We compute the first k eigenvectors of L (or of a normalised Laplacian). These eigenvectors embed the data into a k-dimensional spectral space. The embedding reflects connectivity: highly connected points receive similar spectral coordinates.

Finally, we cluster the spectral coordinates, typically with k-means, to obtain k clusters. Spectral clustering thus uses distances indirectly through A and the Laplacian. It performs well on non-convex structures and manifolds where hyperplane methods struggle. Conditioning matters in practice. The similarity matrix should be well behaved and not extremely skewed. Suitable kernel scales a nd normalisation help. Spectral methods usually require k in advance, though heuristics exist to estimate it.

Other algorithms also accept precomputed dissimilarities. Affinity Propagation operates on similarities and finds exemplar points. DBSCAN defines density via an ϵ-neighbourhood, which depends on distances. Many libraries allow a precomputed distance matrix when Euclidean geometry is not appropriate.

B.6 Metrics and Evaluation Protocol

This section defines the metrics and the evaluation protocol used in Chapter 4. We keep conventions explicit and formulas short.

B.6.1 Calendar, alignment, and returns

We use the data window 2021-01-01–2024-06-30. The effective start date t_0 is configuration-dependent due to warm-up. We generate signals on day t and execute on the next tradable day t^+. The daily *simple* portfolio return realised on t^+ is

$$r_t = w_{t-1}^\top R_t,$$

where w_{t-1} are the weights set at the close of day $t-1$ (after applying the signal from $t-1$) and R_t are asset simple returns on t. Unless noted, the risk-free rate is 0. Rates are decimals in computation and may be shown as percentages in tables.

When comparing multiple configurations, we either report on each strategy's own window (per-strategy t_0) or on the intersection-aligned window (common t_0 = max across the set). We state which one we use alongside each result.

B.6.2 NAV, compounding, and annualisation

We track net asset value (NAV) by compounding simple returns:

$$\mathrm{NAV}_t = \prod_{u=1}^{t}(1+r_u), \qquad \text{Cumulative Return} = \mathrm{NAV}_T - 1.$$

The annualised arithmetic mean return is $\mathrm{AnnRet} = 365 \cdot \bar{r}$, where $\bar{r}$ is the sample mean of daily returns. The compound growth rate is

$$\mathrm{CAGR} = \mathrm{NAV}_T^{365/T} - 1.$$

Annualised volatility is

$$\sigma_{\mathrm{ann}} = \sqrt{365}\,\mathrm{sd}(r_t).$$

B.6.3 Headline risk–return ratios

With risk-free 0, the Sharpe ratio is

$$\mathrm{Sharpe} = \frac{\bar{r}}{\mathrm{sd}(r_t)}\sqrt{365}.$$

The Sortino ratio uses downside deviation with target 0:

$$\sigma_- = \sqrt{\tfrac{1}{T}\sum_{t=1}^{T} \min(r_t, 0)^2}, \qquad \mathrm{Sortino} = \frac{\bar{r}}{\sigma_-}\sqrt{365}.$$

The Calmar ratio is

$$\mathrm{Calmar} = \frac{\mathrm{CAGR}}{|\mathrm{MaxDD}|}.$$

B.6.4 Drawdowns and time statistics

We define running peak $P_t = \max_{u \le t} \mathrm{NAV}_u$ and drawdown $DD_t = \mathrm{NAV}_t / P_t - 1$. The maximum drawdown is $\mathrm{MaxDD} = \min_t DD_t$. We record the peak date, the trough date, and the recovery date (first $u > t$ with $\mathrm{NAV}_u \ge P_t$). The *longest drawdown* counts days between peak and full recovery. Time in market is the fraction of days with non-zero gross exposure:

$$\mathrm{TiM} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\{\|w_t\|_1 > 0\}.$$

Active days report the count of tradable days used for each configuration.

B.6.5 Alpha, win rate, and rolling metrics

We estimate alpha versus a reference series (e.g. the Baseline) by

$$r_t^{\mathrm{str}} = \alpha_{\mathrm{daily}} + \beta\, r_t^{\mathrm{ref}} + \varepsilon_t, \qquad \text{Alpha (ann)} = 365 \cdot \hat{\alpha}_{\mathrm{daily}}.$$

The win rate is $\frac{1}{T}\sum_t \mathbb{1}\{r_t > 0\}$. Rolling statistics use a fixed window (default: 126 trading days). Rolling Sharpe is computed from windowed $\bar{r}$ and $\mathrm{sd}(r_t)$ and annualised with 365.

B.6.6 Transaction costs and turnover

We model costs on target changes. Daily turnover is

$$\text{turnover}_t = \sum_i |w_{t,i} - w_{t-1,i}|.$$

With per-side fee c (in basis points), the cost is

$$\text{cost}_t = (c \times 10^{-4}) \cdot \text{turnover}_t.$$

Net simple return is $r_t^{\text{net}} = r_t^{\text{gross}} - \text{cost}_t$. We charge the first trade. The default is *target-diff*; a *full-rebalance* variant (drift offset back to equal weights) produces higher costs.

B.6.7 Newey–West t-test for the mean of daily excess returns

We test whether the mean daily *excess* return is zero. Let $\{r_t\}_{t=1}^T$ denote daily excess returns (gross or net of costs as stated). The null is $H_0 : \mu = \mathbb{E}[r_t] = 0$ against a two-sided alternative. We allow for heteroskedasticity and autocorrelation of unknown form.

Estimator and test statistic. The sample mean is $\bar{r} = \frac{1}{T}\sum_{t=1}^T r_t$. The Newey–West (HAC) variance of $\bar{r}$ with Bartlett weights and lag length L is

$$\widehat{\mathrm{Var}}(\bar{r}) = \frac{1}{T}\left(\hat{\gamma}_0 + 2\sum_{k=1}^{L} w_k\,\hat{\gamma}_k\right), \qquad w_k = 1 - \frac{k}{L+1} \quad (k = 1, \ldots, L),$$

where the sample autocovariances are

$$\hat{\gamma}_k = \frac{1}{T}\sum_{t=k+1}^{T} (r_t - \bar{r})\,(r_{t-k} - \bar{r}), \qquad k = 0, 1, \ldots, L.$$

The Newey–West t-statistic for $H_0 : \mu = 0$ is

$$\hat{t}_{\text{NW}} = \frac{\bar{r}}{\sqrt{\widehat{\mathrm{Var}}(\bar{r})}}.$$

Under H_0 and standard regularity conditions, $\hat{t}_{\mathrm{NW}} \xrightarrow{d} \mathcal{N}(0,1)$, so we use normal critical values (two-sided unless stated).

Data-driven lag length. We choose the bandwidth L by a data-driven rule. One convenient choice is

$$L \;=\; \left\lfloor 4\left(\frac{T}{100}\right)^{2/9} \right\rfloor,$$

though other automatic rules are admissible. Larger L captures longer serial dependence but increases variance; the rule balances this trade-off.

We centre and annualise point estimates separately as needed; the t-statistic itself is computed on daily returns. If the risk-free rate is non-zero, we form $r_t = R_t - R_t^f$ before applying the test. One can also report a HAC confidence interval for the mean as

$$\bar{r} \;\pm\; z_{1-\alpha/2}\,\sqrt{\widehat{\mathrm{Var}}(\bar{r})},$$

whilst keeping moving-block bootstrap intervals for annualised return and Sharpe as a complementary, serial-dependence–aware check.

B.6.8 Bootstrap confidence intervals

We form 95% confidence intervals for AnnRet and Sharpe via a moving-block bootstrap (MBB).

1. Choose block length $b = 10$ and draws $B = 2000$.
2. Sample $\lceil T/b \rceil$ overlapping blocks with replacement; concatenate and trim to length T.
3. Compute the statistic (AnnRet or Sharpe) on the resample using the same annualisation factor 365.
4. Take the 2.5th and 97.5th percentiles across the B replicates.

B.7 Model-selection bias corrections: WRC and SPA

We compare many candidate rules. This creates selection bias. WRC and SPA adjust for it.

Setup. Let $d_t^{(j)}$ be the daily performance difference of strategy j versus the benchmark. We test whether any strategy has positive expected performance relative to the benchmark.

WRC (White's Reality Check). It builds a null where the best-looking strategy is driven by noise. We compute the maximal statistic over j, then use a stationary bootstrap on $\{d_t^{(j)}\}$ to get the null distribution. The p-value controls data snooping across the whole family.

SPA (Hansen). SPA refines the critical values using a stepwise adjustment. It reduces the penalty on clearly poor rules, improving power while still accounting for multiple testing.

Interpretation. A large overall p means we cannot reject "no outperformance versus the benchmark". Per-strategy mean differences still describe economics (e.g., bp/day), but claims of superiority must respect the overall test.

Bibliography

Andrew Ang and Geert Bekaert. International asset allocation with regime shifts. *The Review of Financial Studies*, 15(4):1137–1187, 06 2002. ISSN 0893-9454.

Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert. Lead-lag detection and network clustering for multivariate time series with an application to the us equity market, 2022.

Chris Brooks, Alistair G. Rew, and Stuart Ritson. A trading strategy based on the lead–lag relationship between the spot index and futures contract for the ftse 100. *International Journal of Forecasting*, 17(1):31–44, 2001. ISSN 0169-2070.

Kuo-Tsai Chen. Iterated integrals and exponential homomorphisms. *Proceedings of the London Mathematical Society*, s3-4(1):502–512, 1954.

Rongbo Chen, Mingxuan Sun, Kunpeng Xu, Jean-Marc Patenaude, and Shengrui Wang. Clustering-based cross-sectional regime identification for financial market forecasting. In Christine Strauss, Alfredo Cuzzocrea, Gabriele Kotsis, A. Min Tjoa, and Ismail Khalil, editors, *Database and Expert Systems Applications*, pages 3–16, Cham, 2022. Springer International Publishing. ISBN 978-3-031-12426-6.

Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning, 2025.

Jim Gatheral, Thibault Jaisson, and Mathieu Rosenbaum. Volatility is rough. *Quantitative Finance*, 18(6):933–949, 2018.

James D. Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57(2):357–384, 1989.

James D Hamilton and Raul Susmel. Autoregressive conditional heteroskedasticity and changes in regime. *Journal of Econometrics*, 64(1):307–333, 1994. ISSN 0304-4076.

Blanka Horváth, Zacharia Issa, and Aitor Muguruza. Clustering market regimes using the wasserstein distance. *arXiv preprint arXiv:2110.11848*, 2021.

Blanka Horváth, Zacharia Issa, and Aitor Muguruza. Clustering market regimes using the wasserstein distance. *Journal of Computational Finance*, 28(1):1–39, 2024.

Nicolas Huth and Frédéric Abergel. High frequency lead/lag relationships - empirical facts, 2012.

Zacharia Issa and Blanka Horvath. Non-parametric online market regime detection and regime clustering for multidimensional and path-dependent data structures, 2023.

Andrew Lo. The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. 10 2004.

Yutong Lu, Ning Zhang, Gesine Reinert, and Mihai Cucuringu. A tug of war across the market: overnight-vs-daytime lead-lag networks and clustering-based portfolio strategies. SSRN Electronic Journal, June 2025. Available at SSRN: `https://ssrn.com/abstract=5371952` or `http://dx.doi.org/10.2139/ssrn.5371952`.

Terry J. Lyons. Differential equations driven by rough signals. *Revista Matemática Iberoamericana*, 14(2):215–310, 1998.

Terry J. Lyons and Zhongmin Qian. *System Control and Rough Paths*. Oxford Mathematical Monographs. Oxford University Press, 2002. ISBN 978-0198506485.

David S. Matteson and Nicholas A. James. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109(505):334–345, 2014.

Hao Meng, Hai-Chuan Xu, Wei-Xing Zhou, and Didier Sornette. Symmetric thermal optimal path and time-dependent lead-lag relationship: novel statistical tests and application to uk and us real-estate and monetary policies. *Quantitative Finance*, 17 (6):959–977, June 2017.

Tobias J. Moskowitz, Yao Hua Ooi, and Lasse Heje Pedersen. Time series momentum. *Journal of Financial Economics*, 104(2):228–250, 2012. ISSN 0304-405X. Special Issue on Investor Sentiment.

Cristopher Salvi, Maud Lemercier, Chong Liu, Blanka Hovarth, Theodoros Damoulas, and Terry Lyons. Higher order kernel mean embeddings to capture filtrations of stochastic processes. *arXiv preprint arXiv:2109.03582*, 2021.

Thilo A. Schmitt, Desislava Chetalova, Rudi Schäfer, and Thomas Guhr. Non-stationarity in financial time series: Generic features and tail behavior. *Europhysics Letters*, 103(5):58003, sep 2013.