mathematics

MDPI

Article

A Distribution-Free Neural Estimator for Mean Reversion, with Application to Energy Commodity Markets

Carlo Mari [1,*] **and Emiliano Mari** [2]

[1] DEIM—Department of Economics, Engineering, Society, Business Organization, University of Tuscia, 01100 Viterbo, Italy
[2] Sydus, 05018 Orvieto, Italy; emiliano.mari@sydus.it
* Correspondence: carlo.mari@unitus.it

Abstract

Accurate estimation of the mean-reversion speed α in the AR(1) process $X_{t+1} = (1-\alpha)X_t + \varepsilon_t$ is central to energy-commodity modelling. Classical estimators such as GARCH, jump-diffusion, and regime-switching produce model-conditioned estimates by embedding α within distributional assumptions, so that different model choices yield different $\hat{\alpha}$ values from the same series without a principled criterion to adjudicate. We propose a distribution-free neural estimator based on a Temporal Convolutional Network (TCN) trained on synthetic AR(1) series with Sinh-ArcSinh (SAS) innovations. *Distribution-free* here means that no parametric family is assumed for the innovation distribution at inference time: the estimator imposes no distributional hypothesis when processing a new series. The SAS family serves as a training vehicle—not a model for the real data—chosen for its ability to span a broad range of tail weights and asymmetry profiles. The theoretical foundation is spectral invariance: the Yule–Walker equations establish that the autocorrelation structure $\rho_k = (1-\alpha)^k$ depends on α alone, provided innovations are uncorrelated across lags—a condition satisfied not only by i.i.d. innovations but also by conditionally heteroscedastic processes such as GARCH. The TCN therefore generalises to volatility-clustering environments without modification, learning to extract α from temporal dependence alone, independently of the marginal innovation distribution and of the temporal variance structure. On held-out test series the estimator outperforms all classical competitors, with the advantage growing monotonically with non-Gaussianity. A robustness analysis on three out-of-distribution innovation families and on AR(1)-GARCH(1,1) processes empirically validates the spectral invariance guarantee across both marginal and temporal variance structure, including near-integrated GARCH processes where innovation kurtosis far exceeds the training range. The distribution-free $\hat{\alpha}$ enables a two-stage pipeline in which α and the innovation distribution are characterised independently—a decoupling structurally impossible in classical likelihood-based approaches. Once trained, the TCN acts as a universal mean-reversion estimator applicable to any price series without re-fitting. Applied to four energy markets—Italian natural gas (PSV price), Italian electricity (PUN price), US Henry Hub, and US PJM West Hub—spanning log-return kurtosis from near-Gaussian to strongly heavy-tailed, the TCN yields robust, distribution-free estimates of mean-reversion speed.

Keywords: mean reversion; temporal convolutional networks; distribution-free estimation; sinh-arcsinh distribution; AR(1) process; deep learning; energy markets

MSC: 37M10

Academic Editors: Yuping Song and Hanchao Wang

Received: 10 March 2026
Revised: 7 April 2026
Accepted: 9 April 2026
Published: 13 April 2026

1. Introduction

1.1. Mean Reversion in Energy Markets

Financial time series modelling is fundamental to a wide range of quantitative applications: risk measurement, derivative pricing, portfolio optimisation, and stress testing. The quality of model outputs depends critically on how well the assumed stochastic dynamics match the true data-generating process. Two properties are especially important—and especially hard to pin down from a finite sample: the degree of persistence in the price level, and the shape of the innovation distribution.

A distinctive and economically fundamental characteristic of energy commodity prices is their tendency toward mean reversion—the phenomenon whereby prices, after deviating from long-run equilibrium levels, exhibit a systematic tendency to return toward those levels over time [1–3]. The economic foundations of mean reversion are rooted in the physical and institutional structure of commodity production and consumption. For natural gas, production costs establish effective price floors: when prices fall significantly below marginal extraction and transportation costs, producers reduce output, constraining supply until prices recover. Conversely, price ceilings emerge from substitution effects and demand destruction: sustained high prices induce consumers to switch to alternative fuels or reduce consumption. These supply-demand adjustments create restoring forces that pull prices back toward equilibrium. Electric power markets exhibit even stronger mean-reverting dynamics due to the fundamental non-storability constraint [3]: electricity must be consumed instantaneously, so extreme spikes occur during demand peaks or supply shortfalls but prices rapidly return to baseload levels once the transient imbalance resolves. The half-life of mean reversion in power markets is typically measured in days rather than weeks or months.

Mathematically, mean reversion is modelled through Ornstein–Uhlenbeck processes or their discrete-time counterparts [4]. Denoting by X_t a stationary process (in applications, derived from raw price series by a stationarity-inducing preprocessing step; see Section 6), the discrete-time AR(1) dynamics read:

$$X_{t+1} = (1 - \alpha)\, X_t + \varepsilon_t, \tag{1}$$

where $\alpha > 0$ is the mean-reversion speed, $|1 - \alpha| < 1$ ensures stationarity, and ε_t is the stochastic innovation. The half-life of mean reversion,

$$\tau_{1/2} = \frac{\ln 2}{\alpha}, \tag{2}$$

measures the expected time for a deviation to decay by half.

The AR(1) model (1) and the estimator developed in this paper are formulated for a generic stationary process $\{X_t\}$ with uncorrelated innovations ε_t; the theory makes no reference to the specific interpretation of X_t. First differences $\Delta X_t = X_{t+1} - X_t$ are relevant to the distributional characterisation of ε_t, which is kept entirely separate from the estimation of α. In the energy market application of Section 6, X_t represents the detrended log-price and ΔX_t the corresponding log-return.

Accurately estimating α—equivalently, the half-life $\tau_{1/2}$—is not merely a statistical exercise: it is the foundation of quantitative risk management and derivative pricing in energy markets, and errors in $\hat{\alpha}$ propagate directly into all downstream calculations. In risk measurement, Value-at-Risk and Expected Shortfall at horizons of one week to one month are highly sensitive to α: underestimating α inflates multi-period tail risk estimates and leads to excessive capital requirements, while overestimating α understates persistence and underestimates downside exposure [5]. In derivative pricing, instruments written on

energy commodities—swing options, storage contracts, tolling agreements, and spread options—derive their value from the entire distribution of price paths; mispricing α produces substantial valuation errors for longer-dated contracts [4,5]. In scenario simulation and stress testing, Monte Carlo trajectories generated by iterating (1) are highly sensitive to α: an incorrect value produces scenarios with wrong persistence, distorting capital allocation, margin requirements, and hedging strategies [6,7].

Yet the estimation of α is complicated by the fact that energy price innovations are not Gaussian: they display fat tails, volatility clustering, and occasional price spikes. Log-returns are well documented to exhibit heavy tails—Pearson kurtosis substantially above the Gaussian benchmark of 3—driven by sporadic supply-demand imbalances, demand shocks, and the non-storability of electricity [1,2,8]. They also display significant asymmetry: non-zero skewness arises from the structural asymmetry of energy supply shocks, with positive and negative deviations from equilibrium having different magnitudes and frequencies. These distributional features of energy price innovations—heavy tails and structural asymmetry—expose a fundamental limitation of classical parametric approaches to estimating α, as discussed in the following section.

It is important to note that the AR(1) model (1) is adopted here as a *reduced-form statistical representation*, not as a structural economic model. In practice, large price movements may originate from rare structural shocks—geopolitical events, supply disruptions, extreme weather episodes, or regulatory interventions—that propagate through production networks and input–output linkages before affecting observed commodity prices [9,10]. Within the reduced-form AR(1) framework, such structural shocks are absorbed into the innovation term ε_t as large realisations; their origin and propagation mechanism are not modelled explicitly. This abstraction does not invalidate the statistical estimator, but it has an important implication for the interpretation of $\hat{\alpha}$: the estimated mean-reversion speed captures the *net* restoring force operating on observed prices, integrating over the full distribution of shock sizes and types encountered in the sample period.

1.2. Existing Approaches and Their Limitations

Classical approaches estimate α from a time series by embedding the AR(1) recursion (1) within a fully specified parametric model and maximising the log-likelihood (MLE). Three families are prominent in the energy finance literature.

AR(1)-GARCH(1,1) [11] augments the AR(1) mean equation with a conditional heteroscedastic variance equation. The MLE of α is obtained jointly with the GARCH variance parameters.

Jump-diffusion [12] models innovations as the superposition of a diffusion component and random jumps governed by a Poisson process. The likelihood sums over possible jump counts.

Regime-switching [13] allows the innovation variance to switch between a finite number of states governed by a latent Markov chain; α is estimated via the Hamilton filter and the expectation-maximisation (EM) algorithm.

Each model couples α with distributional parameters through the likelihood function. The MLE of α simultaneously pursues two objectives: fitting the autocorrelation structure of the series (primarily sensitive to α via the Yule–Walker relations) and fitting the marginal (unconditional, one-dimensional) distribution of the innovations (sensitive to distributional parameters, but also to α through the stationary moments of ΔX_t). When the model is correctly specified, both objectives are mutually consistent and the MLE yields a consistent estimator of the true α. When the model is misspecified, they conflict, and the likelihood resolves the conflict by shifting $\hat{\alpha}$ away from its true value.

A fundamental consequence is that $\hat{\alpha}$ is not a distribution-free estimate of mean-reversion speed: it is a *model-conditioned* quantity whose numerical value depends on the distributional assumptions embedded in each likelihood. Comparing $\hat{\alpha}_{\text{GARCH}}$ with $\hat{\alpha}_{\text{Jump-Diff}}$ is not a comparison of two estimates of the same underlying parameter; it is a comparison of two model-specific answers to the question "what value of α is most consistent with the data *given this particular distributional model*?" In the absence of knowledge of the true innovation distribution—which is precisely the situation faced in practice—there is no principled criterion to adjudicate between them. The disagreement among classical estimators is not a sampling artefact that would vanish with more data: it is a structural consequence of model dependence that persists asymptotically.

This model dependence is a property of the *inference procedure*, not of the data. The mean-reversion speed α is a structural parameter that governs the autocorrelation pattern of the observed series, independently of the innovation distribution. Classical estimators obscure this structural parameter because they require a specific distributional model to form the likelihood—and the estimated α is contaminated by whatever distributional assumptions the chosen model embeds. The goal is therefore to remove this contamination at the level of inference: to produce an estimator that operates without any distributional assumption on the real data, reading α solely from the temporal structure of the observed series.

Deep learning methods have recently been applied to energy price forecasting and modelling [14–16], demonstrating the potential of neural architectures for capturing the complex dynamics of energy markets. Their application to the estimation of structural dynamical parameters such as α has, however, remained limited [17]. Distribution-free estimation of α from a single time series, without any assumption on the innovation distribution at inference time, has therefore remained largely unexplored.

The central objective of this paper is to address this gap by introducing a distribution-free neural estimator that: (i) produces model-independent $\hat{\alpha}$ estimates relying solely on the autocorrelation structure of the series; (ii) enables a clean two-stage decomposition in which α and the innovation distribution are characterised independently; and (iii) once trained on synthetic data, functions as a universal mean-reversion instrument applicable to any market or time period without re-fitting.

1.3. A Distribution-Free Neural Approach

We propose a Temporal Convolutional Network (TCN) [18] as a distribution-free estimator of α: a neural architecture that processes the full sample path $\{X_1, \ldots, X_T\}$ and learns to extract α from the autocorrelation structure at all lags simultaneously, without imposing any distributional assumption. This approach belongs to the broader family of simulation-based (likelihood-free) estimators [19–23], in which neural networks perform parameter estimation from simulated data without requiring explicit likelihood evaluation. Trained on 2×10^5 synthetic AR(1) series with Sinh-ArcSinh (SAS) distributed innovations [24,25], the TCN learns to read the temporal dependence structure, not the marginal distribution of the observations.

We use *distribution-free* to mean that no parametric family is assumed for the innovation distribution of the real data at inference time: once trained, the TCN imposes no distributional hypothesis when applied to a new series. "Distribution-free" is therefore a property of the *inference step*. In our framework, the formal guarantee underlying this property is not an empirical observation but the Yule–Walker spectral invariance theorem: the lag-k autocorrelation of the stationary AR(1) process satisfies

$$\rho_k = (1 - \alpha)^k, \tag{3}$$

provided only that $\mathbb{E}[\varepsilon_t \varepsilon_{t-k}] = 0$ for all $k \neq 0$ and $\mathrm{Var}(\varepsilon_t) < \infty$; since ρ_k depends on α alone, any estimator that reads α exclusively from the autocorrelation pattern inherits this invariance at inference time, regardless of which training distribution generated the calibration series.

This *spectral invariance* holds for any stationary innovation process satisfying these weak conditions—it does not depend on the shape of the marginal distribution, on the presence of heavy tails, or on volatility clustering. In particular, the GARCH(p, q) family satisfies the zero-autocorrelation condition by construction [11]: GARCH innovations are uncorrelated across lags even though they are conditionally heteroscedastic. Spectral invariance therefore extends the distribution-free guarantee from the marginal innovation distribution to the temporal variance structure: any estimator that reads α exclusively from the autocorrelation pattern (3) is distribution-free not only with respect to the innovation shape but also with respect to volatility dynamics. This has a direct practical consequence: a TCN trained on synthetic AR(1) series with i.i.d. SAS innovations transfers to processes with GARCH-type volatility clustering without any modification. This theoretical prediction is empirically validated in Section 5: across a wide range of GARCH volatility persistence levels, TCN accuracy remains essentially unchanged, confirming that volatility clustering does not impair estimation based on the autocorrelation structure.

The choice of SAS as the training distribution does not constitute a distributional assumption on the true innovation process: SAS serves as a training vehicle, not a model for the real data. SAS is chosen because its two-parameter family spans a continuous range of innovation profiles—from near-Gaussian to strongly heavy-tailed and skewed—and admits kurtosis-uniform sampling without additional reparametrisation; the technical details are given in Section 2.2. By exposing the network to this broad diversity of innovation shapes, the training procedure ensures that the autocorrelation structure (3) is the only signal that remains stable across all training examples, compelling the network to base its estimates on temporal dependence alone rather than on distributional features that vary from instance to instance.

Crucially, the specific kurtosis range used during training has no bearing on the validity of the inference-time estimate: by spectral invariance (3), the autocorrelation function depends only on α regardless of the innovation distribution, so the TCN's learned mapping remains valid for any innovation kurtosis, whether inside or outside the training range. Empirical support for this is provided by the robustness analysis of Section 5: stable performance across Student-t, GED, and Gaussian mixture innovations—three families with fundamentally different tail structures, none seen during training—confirms that the network has learned autocorrelation-based features rather than SAS-specific artefacts. Out-of-distribution generalisation is therefore a structural prediction, not a fortuitous empirical finding.

1.4. Advantages over Existing Estimators

The classical Yule–Walker estimator $\hat{\alpha}_{\mathrm{YW}} = 1 - \hat{\rho}_1$ exploits precisely this property: it recovers α from the lag-1 sample autocorrelation alone, without any distributional assumption. Its limitation is that it discards the information in higher lags $\rho_2, \rho_3, \ldots$: for slow mean reversion ($\alpha \to 0$), the autocorrelation decay is shallow and the lag-1 estimate alone provides a weak signal for α, with a well-known finite-sample downward bias [26]. A natural extension replaces the single lag-1 estimate with a nonlinear least-squares fit of $\hat{\alpha}$ to the full empirical autocorrelation profile $(1 - \alpha)^k$; this multi-lag Yule–Walker estimator (ACF-YW) serves as the distribution-free baseline in the benchmark of Section 4.

It is natural to ask what the TCN adds over ACF-YW, a distribution-free estimator that already exploits the full autocorrelation profile. The answer has three components. First,

adaptive lag-weighting: ACF-YW fits $\hat{\alpha}$ to the empirical ACF with fixed uniform weights; the TCN weights all lags $\rho_1, \rho_2, \ldots, \rho_{T-1}$ through learned dilated convolutions, adapting the weighting to the signal-to-noise conditions of each series. For slow mean reversion (small α), where higher lags are most informative and uniform weighting is suboptimal, the adaptive gain is substantial—consistent with the TCN achieving its best performance precisely in the $\alpha \in [0.01, 0.11)$ regime (Section 3). Second, *finite-sample adaptation*: ACF-YW relies on an asymptotic nonlinear least-squares criterion; the TCN, trained entirely on finite-T series, learns the finite-sample distribution of the autocorrelation pattern implicitly, adapting its aggregation to the effective noise level rather than relying on asymptotic approximations. Third, *robustness under heavy tails*: for heavy-tailed innovations, the variance of sample autocorrelations grows with kurtosis, degrading the signal-to-noise ratio of the empirical ACF at all lags. By learning to adaptively down-weight noisy high-lag estimates, the TCN compensates for this increased noise, consistent with the empirical finding that TCN accuracy improves as kurtosis increases while ACF-YW accuracy remains flat. A plausible complementary mechanism is that the occasional large deviations from equilibrium under heavy tails provide locally clean geometric-decay episodes that are easier to read even from short stretches of the series; however, this remains a conjecture supported by empirical evidence rather than a formal result.

Section 4 evaluates the TCN against four competitors—ACF-YW, GARCH(1,1), Jump-Diffusion, and Regime-Switching—on synthetic AR(1) series with known α, using MAE of $\hat{\alpha}$ as the accuracy metric. The comparison is symmetric: each classical estimator is re-estimated from scratch on each individual test series via independent maximum-likelihood or EM optimisation, with no access to the true parameter values or the generating distribution. Against the ACF-YW baseline the TCN achieves an overall MAE of 0.00918 versus 0.01944; ACF-YW accuracy is flat across innovation kurtosis bins, confirming that the TCN's advantage stems from adaptive lag-weighting and finite-sample adaptation rather than multi-lag aggregation per se. Against the three classical parametric estimators the TCN outperforms GARCH(1,1) (MAE 0.01545), Jump-Diffusion (MAE 0.01017), and Regime-Switching (MAE 0.01975), with the margin widening monotonically with κ_ε, the Pearson kurtosis of ε_t, reflecting the growing distributional misspecification bias of likelihood-based methods under heavy-tailed innovations. The adaptive lag-weighting advantage also extends to temporal variance structure: across AR(1)-GARCH(1,1) processes ranging from i.i.d. Gaussian innovations to near-integrated volatility clustering, the TCN remains stable while ACF-YW degrades at high persistence levels, confirming that data-adaptive weighting is beneficial beyond distributional robustness.

Regarding model complexity: the estimation target is a scalar (α), but the input is a variable-length sequence of up to $T = 1826$ observations. The complexity of the TCN architecture is driven by the input dimension, not the output dimension—the seven-block dilated architecture with receptive field covering 97% of the input length is necessary to aggregate the decay pattern of correlations across all lags. A linear regressor on lag-1 autocorrelation (Yule–Walker) is far simpler but discards most of the available information. The TCN is therefore not over-engineered for a scalar output: it is appropriately sized for a sequence-to-scalar regression task where the relevant signal is distributed across the entire temporal structure of the input.

The estimated $\hat{\alpha}$ enables a natural two-stage pipeline: clean residuals $\hat{\varepsilon}_t$ are available for independent distributional analysis by any method, free from the coupling between α and distributional parameters that is the source of model dependence in classical estimators.

A further consequence of the train-on-synthetics design is that the trained TCN functions as a *universal mean-reversion estimator*: calibrated once on synthetic AR(1) dynamics, it applies to any price series—regardless of market, time period, or innovation

distribution—without re-fitting or re-optimisation. Unlike classical estimators, which must solve a fresh numerical optimisation problem for every new series, the TCN is a reusable analytical instrument that delivers $\hat{\alpha}$ with no dependence on the specific data used in deployment.

1.5. Application to Energy Commodity Markets

The methodology described above is developed generically—the TCN is trained exclusively on synthetic AR(1) series with no reference to any specific market—and is subsequently applied to four energy commodity markets: Italian natural gas (PSV price), Italian electricity (PUN price), US Henry Hub natural gas, and US PJM West Hub electricity. These markets span different storage characteristics, geographical regions, and innovation profiles—from near-Gaussian to strongly heavy-tailed—providing a demanding and diverse validation environment. At inference, real price series are preprocessed to yield a stationary residual conforming to the AR(1) structure of the synthetic training data (see Section 6); after standardisation, the TCN outputs $\hat{\alpha}$ in under one millisecond. The US Henry Hub case, with log-return kurtosis well beyond the nominal training range, illustrates the practical consequence of spectral invariance: the TCN's validity does not depend on the innovation kurtosis falling within any predetermined range.

1.6. Contributions

The main contributions of this paper are: (i) a distribution-free neural estimator of mean-reversion speed whose validity extends, by spectral invariance, from i.i.d. to GARCH-type innovations; (ii) a two-stage pipeline that decouples α estimation from innovation distribution characterisation; (iii) a universal train-once instrument requiring no re-fitting at deployment; (iv) a controlled benchmark against four classical estimators and a comprehensive out-of-distribution robustness analysis; and (v) application to four energy commodity markets with markedly different kurtosis profiles.

1.7. Paper Organisation

The remainder of the paper is organised as follows. Section 2 presents the methodology: the formal problem statement, the SAS innovation model, the synthetic data generation protocol, the TCN architecture, and the training procedure. Section 3 evaluates the TCN in isolation: training dynamics and test-set performance stratified by α bin and innovation kurtosis. Section 4 introduces the three classical parametric estimators and reports the kurtosis-stratified comparison on synthetic data. Section 5 analyses out-of-distribution generalisation to innovation families not seen during training, including extreme kurtosis and AR(1)-GARCH(1,1) processes with varying persistence, empirically validating the spectral invariance guarantee for temporal variance structure. Section 6 applies all estimators to four energy commodity markets. Section 7 draws conclusions and outlines directions for future research.

2. Methodology

2.1. Problem Formulation

Let $\{X_t\}_{t=1}^{T}$ be a stationary time series assumed to follow the AR(1) mean-reversion model

$$X_{t+1} = (1 - \alpha)\, X_t + \varepsilon_t, \tag{4}$$

where ε_t is a stationary innovation process satisfying $\mathbb{E}[\varepsilon_t\, \varepsilon_{t-k}] = 0$ for all $k \neq 0$ and $\mathrm{Var}(\varepsilon_t) < \infty$. This assumption is weaker than i.i.d.: it is satisfied by i.i.d. innovations, but also by conditionally heteroscedastic processes such as GARCH, where innovations are uncorrelated across lags but not identically distributed over time. The parameter of interest

is the mean-reversion speed α. It is well known from the Yule–Walker equations that the autocorrelation structure $\rho_k = (1-\alpha)^k$ holds regardless of $\mathbb{E}[\varepsilon_t]$, so zero-mean innovations are not required for the validity of the estimator.

Standardisation. Prior to any estimation, the observed series is standardised:

$$\tilde{X}_t = \frac{X_t - \bar{X}}{s_X}, \qquad \bar{X} = \frac{1}{T}\sum_{t=1}^{T} X_t, \quad s_X^2 = \frac{1}{T-1}\sum_{t=1}^{T}(X_t - \bar{X})^2. \tag{5}$$

Under the AR(1) model (4), the stationary variance is $\mathrm{Var}(X_t) = \sigma_\varepsilon^2/[1-(1-\alpha)^2]$. Dividing by $s_X \approx \sqrt{\mathrm{Var}(X_t)}$ removes the scale σ_ε from the input, so the standardised series $\tilde{X}_t$ carries autocorrelation information about α without any dependence on the innovation scale. Since standardisation is a location-scale transformation, it also preserves the skewness and kurtosis of the series, so $\tilde{X}_t$ retains the full distributional shape of X_t. The estimation task is therefore:

$$\hat{\alpha} = f_{\hat{\theta}}(\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_T), \tag{6}$$

where $f_{\hat{\theta}}$ is the trained neural estimator (architecture and training described in Sections 2.4–2.5).

2.2. The Sinh-ArcSinh Distribution

Training a distribution-free estimator requires exposure to a sufficiently wide variety of innovation distributions, so that no distributional feature is consistently predictive of α and the network is compelled to learn from the autocorrelation structure alone. We use the Sinh-ArcSinh (SAS) distribution introduced by Jones and Pewsey [24,25] as the *training vehicle*: it is a flexible family whose parameters allow kurtosis and skewness to be varied continuously and jointly over a broad range, providing the necessary training diversity. The choice of SAS is a computational convenience, not a modelling assumption about the true innovation process. By spectral invariance, the validity of the inference-time estimate does not depend on whether the true innovations belong to the SAS family, nor on whether their kurtosis falls within the nominal training range: it depends only on innovations being uncorrelated across lags.

2.2.1. Definition

If $Z \sim \mathcal{N}(0,1)$, the SAS random variable is

$$\varepsilon = \sinh\left(\frac{\operatorname{arcsinh}(Z) + \nu}{\tau}\right), \tag{7}$$

where $\nu \in \mathbb{R}$ controls asymmetry and $\tau > 0$ controls tail weight. The transformation (7) is a monotone bijection on $\mathbb{R}$, so the density is obtained via the change-of-variables formula. A positive scale parameter ℓ could multiply the right-hand side, but since every generated series is standardised to zero mean and unit variance before input to the TCN, any such factor cancels identically; we therefore set $\ell = 1$ without loss of generality. For $\nu \neq 0$, the innovation (7) has non-zero mean $\mathbb{E}[\varepsilon_t] = \sinh(\nu/\tau)\, M(1/\tau)$, where $M(c) = \frac{e^{1/4}}{2\sqrt{2\pi}}\left[K_{(c+1)/2}(\frac{1}{4}) + K_{(c-1)/2}(\frac{1}{4})\right]$ and $K_p(\cdot)$ is the modified Bessel function of the second kind of order p [24], which induces a non-zero stationary mean $\mathbb{E}[X_t] = \mathbb{E}[\varepsilon_t]/\alpha$ in the AR(1) process. Key special cases of the SAS family are: $\nu = 0$, $\tau = 1$ (Gaussian); $\tau < 1$ (heavier tails than Gaussian); $\tau > 1$ (lighter tails); $\nu \neq 0$ (asymmetric).

2.2.2. SAS Parameter Ranges

The choice of SAS over alternative flexible distributions (e.g., Normal Inverse Gaussian, Generalised Hyperbolic, skew-t) is motivated by a key practical property: in the symmetric case ($\nu = 0$), the Pearson kurtosis is a strictly monotone function of τ alone [24]:

$$\kappa_\varepsilon(\tau) = \frac{M(4/\tau) - 4M(2/\tau) + 3}{2\big[M(2/\tau) - 1\big]^2}. \tag{8}$$

The function $\kappa_\varepsilon(\tau)$ is strictly decreasing in τ, with $\kappa_\varepsilon = 3$ at $\tau = 1$ (Gaussian case) and $\kappa_\varepsilon \to \infty$ as $\tau \to 0^+$. This monotonicity enables direct numerical inversion: given a target κ_ε, Equation (8) yields τ exactly, without additional reparametrisation.

The training ranges are $\kappa_\varepsilon \sim \text{Uniform}(3, 17)$, spanning from the Gaussian baseline ($\kappa_\varepsilon = 3$, $\tau = 1$) to strongly leptokurtic innovations ($\kappa_\varepsilon = 17$, $\tau \approx 0.40$), and $\nu \sim \text{Uniform}(-0.5, +0.5)$. Since Equation (8) holds at $\nu = 0$, τ is first derived from the sampled target κ_ε, and ν is then drawn independently. The actual Pearson kurtosis at the sampled pair (ν, τ) therefore exceeds the target whenever $|\nu| > 0$, reaching ≈ 37 at the boundary $|\nu| = 0.5$, $\tau \approx 0.40$; the asymmetry range produces innovation skewness $\gamma_\varepsilon \in [-4.57, +4.57]$. The model is thus exposed to a substantially wider tail-behaviour and asymmetry spectrum than the nominal ranges suggest, ensuring that the training diversity is sufficient to prevent any distributional feature from being a reliable predictor of α.

We distinguish throughout between κ_ε, the Pearson kurtosis of the innovation ε_t, and the first-difference kurtosis $\kappa(\Delta X_t)$; the two are related through Equation (A6) in Appendix A. SAS is used as a *training vehicle*, not as an assumption on the true innovation distribution. The key practical advantage that motivates this choice is the monotone mapping $\tau \mapsto \kappa_\varepsilon(\tau)$ (Equation (8)), which enables κ_ε-uniform sampling by direct numerical inversion. Any alternative family rich enough to span the same range of kurtosis and skewness profiles would yield an equally valid estimator, since the distribution-free property is grounded in spectral invariance (3), not in the specific choice of SAS.

2.3. Synthetic Data Generation

Training and evaluation sets consist entirely of synthetic AR(1) series generated as follows.

1. **Parameter sampling.** For each series, draw independently: $\alpha \sim \text{Uniform}(0.01, 1.00)$, $\nu \sim \text{Uniform}(-0.50, +0.50)$, $\kappa_\varepsilon \sim \text{Uniform}(3, 17)$. Given κ_ε, the tail parameter τ is obtained by numerical inversion of Equation (8); ν is then applied to the resulting τ. This κ_ε-uniform sampling ensures uniform coverage of innovation tail-weight profiles across the training range.
2. **Series length.** The series length $T_i \sim \text{Uniform}(1200, 1826)$ covers the range of sample sizes typically available for daily energy price series, corresponding to approximately three to five years of calendar or business-day observations.
3. **Innovation generation.** Draw $T_i + T_{\text{burn}}$ independent SAS(ν, τ) samples using the representation (7), where $T_{\text{burn}} = 200$ is a burn-in buffer discarded at the next step.
4. **AR(1) recursion.** Apply the recursive filter (4) starting from $X_0 = 0$; discard the first T_{burn} steps to ensure approximate stationarity.
5. **Standardisation.** Apply Equation (5) to the retained T_i steps. For batch processing, sequences shorter than $T_{\max} = 1826$ are right-zero-padded; the network architecture ensures that only valid (non-padded) timesteps contribute to the output (see Section 2.4).

Figure 1 illustrates the complete estimation pipeline: from parameter sampling and synthetic series generation during training, to standardisation and TCN inference at test time. The training set uses *on-the-fly* generation (new parameter draws and noise realisations

each epoch), so that the model never sees the same series twice across epochs. The validation set ($N_{\text{val}} = 20{,}000$ series) and the test set ($N_{\text{test}} = 20{,}000$ series) are pre-generated once for reproducibility.

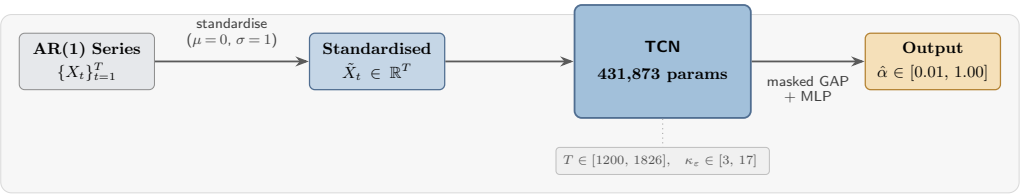

Figure 1. Estimation pipeline. A stationary AR(1) series $\{X_t\}_{t=1}^{T}$ is standardised to yield $\tilde{X}_t$, which is fed to the TCN; the output is the distribution-free mean-reversion speed estimate $\hat{\alpha} \in [0.01, 1.00]$. During training, $\{X_t\}$ is generated synthetically using SAS innovations with parameters sampled as described in Section 2.2; at inference on real data, $\{X_t\}$ is the preprocessed price series (see Section 6).

2.4. TCN Architecture

Choice of architecture. Estimating α requires reading the autocorrelation structure of the full series—a pattern that decays as $\rho_k = (1-\alpha)^k$ across potentially hundreds of lags. The chosen architecture must therefore process the entire time series while strictly respecting temporal causality (the output may only depend on past observations). Recurrent architectures (LSTM, GRU) address this requirement but process sequences step-by-step, are prone to vanishing gradients over long horizons, and are expensive to train in parallel. Transformer architectures introduce self-attention whose computational cost grows quadratically with sequence length and requires explicit positional encoding, adding complexity without benefit for a fixed-length stationary input. The Temporal Convolutional Network (TCN) [18,27] offers a natural alternative: it processes the full sequence in a single parallel pass using dilated causal convolutions stacked in residual blocks, achieves an exponentially growing receptive field with a modest number of layers, and has been shown to match or outperform recurrent models on a wide range of sequence modelling benchmarks. For our problem, the ability to engineer the receptive field to cover the full series length is a decisive practical advantage.

Architecture. The network consists of L stacked residual blocks. Each block applies two dilated causal convolutions with n_f output channels (feature maps), followed by ReLU non-linearities, spatial dropout, and a residual skip connection that adds the block's input directly to its output. A *causal* convolution ensures that the representation at time t depends only on inputs at times $\leq t$. *Dilation* exponentially expands the temporal span covered by each convolution without increasing the number of parameters: block i uses dilation factor $d_i = 2^i$ for $i = 0, 1, \ldots, L-1$, so successive blocks capture patterns at progressively longer time scales. The resulting receptive field (RF)—the number of past timesteps that influence the output—is:

$$\text{RF} = 1 + 2(w-1)\left(2^L - 1\right), \tag{9}$$

where w is the kernel size. With $L = 7$ blocks and $w = 8$, RF $= 1779$ steps, covering 97.4% of $T_{\max} = 1826$. This near-complete temporal coverage ensures the network can exploit autocorrelation patterns at every lag, including those associated with very slow mean reversion ($\alpha \to 0$). The residual connections facilitate gradient flow through the full stack and training stability.

After the final block, Masked Global Average Pooling aggregates the temporal representations across all valid (non-padded) timesteps into a fixed-size vector. LayerNorm—which normalises each sample independently over its feature dimension, with no cross-sample running statistics—is applied before a two-layer multi-layer percep-

tron (MLP) head that maps the pooled representation to the scalar estimate $\hat{\alpha} \in [0.01, 1.00]$. The bounded output range is enforced by a sigmoid activation rescaled to $[0.01, 1.00]$, matching the training distribution of α.

Table 1 summarises the architectural parameters.

Table 1. TCN architecture summary.

Component	Configuration
Residual blocks L	7
Kernel size w	8
Filters n_f	64
Dropout rate	0.05
Dilation schedule	$2^0, 2^1, \ldots, 2^6$
Receptive field	1779/1826 (97.4%)
Total parameters	431,873

Hyperparameter selection. The dominant design criterion is that the receptive field (9) must cover the vast majority of the input series, so that the network can access the full autocorrelation decay implied by small α values (long memory). For a fixed kernel size $w = 8$, Equation (9) gives RF = 883 (48%) for $L = 6$ and RF = 1779 (97.4%) for $L = 7$; the latter satisfies the coverage requirement with a single additional block. Increasing to $L = 8$ would raise RF to 3571—well beyond $T_{\max}$—at the cost of doubling the network depth, so $L = 7$ is the minimal depth achieving near-complete coverage. The kernel size $w = 8$ is chosen because smaller kernels (e.g., $w = 4$) would require a substantially larger L to achieve comparable receptive-field coverage, while $w = 8$ produces sufficient local context at each dilation level. The number of filters $n_f = 64$ balances model capacity with training throughput: with $L = 7$ blocks the resulting 431,873 parameters train comfortably on a single GPU (batch size 256, $T_{\max} = 1826$). The dropout rate $p = 0.05$ was found to prevent overfitting without degrading training speed. Figure 2 shows the TCN architecture.

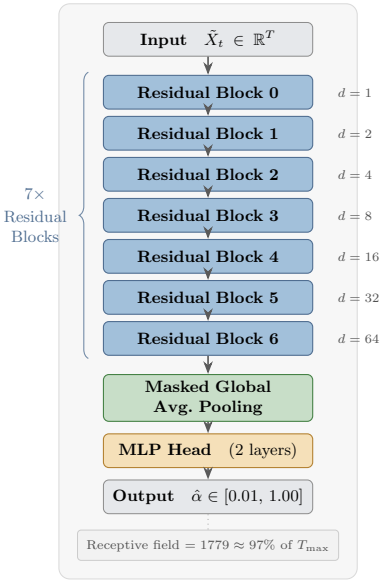

Figure 2. TCN architecture. The standardised input series passes through seven residual blocks with exponentially growing dilation factors $d = 2^0, 2^1, \ldots, 2^6$. Masked Global Average Pooling aggregates the temporal representations across all valid (non-padded) timesteps, and a two-layer MLP head maps the resulting vector to $\hat{\alpha} \in [0.01, 1.00]$. The total receptive field spans 1779 timesteps ($\approx$97% of $T_{\max} = 1826$).

2.5. Training Procedure

Loss function. We minimise the mean absolute error (MAE, L_1 loss):

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\left|\hat{\alpha}_n(\theta) - \alpha_n\right|, \tag{10}$$

where $\hat{\alpha}_n(\theta) = f_\theta(\tilde{X}_n)$ is the TCN prediction for series n (cf. Equation (6)), and α_n is the corresponding true mean-reversion speed. The trained parameters are $\hat{\theta} = \arg\min_\theta \mathcal{L}(\theta)$. MAE is preferred over MSE because it treats errors uniformly across the α range, avoiding the disproportionate penalisation of large errors at extreme α values.

Optimiser and learning-rate schedule. We use the Adam optimiser with initial learning rate $\eta_0 = 10^{-3}$ and a CosineAnnealingLR scheduler over 50 epochs with minimum learning rate $\eta_{\min} = 10^{-5}$. Gradient norms are clipped to 1.0 to prevent gradient explosion.

Data and compute. Training set: $N_{\text{train}} = 200{,}000$ series per epoch (on-the-fly generation); batch size: 256. Validation: $N_{\text{val}} = 20{,}000$ fixed series evaluated after each epoch. The best checkpoint is selected by minimum validation MAE. Training runs on a single NVIDIA RTX 3080 Ti GPU (12 GB VRAM) via Ray Train; each epoch takes approximately 90 s, for a total training time of approximately 75 min over 50 epochs.

Early stopping. Training is halted if the validation MAE does not improve for 10 consecutive epochs (patience = 10); the checkpoint achieving the lowest validation MAE is retained for evaluation.

Weight initialisation. Convolutional kernels are initialised with Kaiming normal initialisation [28], which is optimal for layers followed by ReLU activations. Linear layers in the projection head use Xavier uniform initialisation. All bias terms are initialised to zero, so the sigmoid head initially outputs $\frac{1}{2}$, which maps to $\frac{1}{2} \times 0.99 + 0.01 = 0.505 \approx \mathbb{E}[\alpha]$ for $\alpha \sim \text{Uniform}(0.01, 1)$, yielding a theoretically expected MAE of approximately 0.247 at epoch zero; after the first training epoch the model rapidly learns, dropping to a training MAE of 0.0410.

3. TCN Performance

3.1. Training Dynamics

Table 2 reports training and validation MAE at selected epochs together with the cosine-annealed learning rate. Training and validation MAE track each other closely throughout, with no sign of overfitting. The best checkpoint is selected at epoch 45, achieving a validation MAE of 0.01017. Panel (a) of Figure 3 illustrates the training convergence curve.

Table 2. Training dynamics. Train MAE and validation MAE at selected epochs. The learning rate follows the CosineAnnealingLR schedule ($\eta_0 = 10^{-3}$, $\eta_{\min} = 10^{-5}$, 50 epochs). Best checkpoint at epoch 45 (bold row), validation MAE = 0.01017.

Epoch	Train MAE	Val MAE	LR
1	0.0410	0.0176	9.9×10^{-4}
10	0.0132	0.0132	8.1×10^{-4}
22	0.0116	0.0111	6.0×10^{-4}
30	0.0111	0.0106	2.5×10^{-4}
45	**0.0104**	**0.0102**	$\mathbf{3.6 \times 10^{-5}}$
50	0.0103	0.0102	1.0×10^{-5}

3.2. Test-Set Performance

Table 3 reports the TCN test MAE stratified by α bin over $N_{\text{test}} = 20{,}000$ series. The overall test MAE is 0.01004, marginally below the validation MAE of 0.01017 ($\Delta = 0.00013$), confirming excellent generalisation with no overfitting. The median absolute error is 0.00759, the 90th percentile is 0.02200, and the 99th percentile is 0.04135.

Table 3. TCN test-set performance stratified by α bin ($N_{\text{test}} = 20{,}000$ series, $T_i \sim \text{Uniform}(1200, 1826)$, $\kappa_\varepsilon \sim \text{Uniform}(3, 17)$, $\nu \sim \text{Uniform}(-0.50, +0.50)$). The overall test MAE of 0.01004 is marginally below the validation MAE of 0.01017, confirming no overfitting.

Bin (α)	***N***	**MAE**
$[0.01,\ 0.11)$	2095	0.00484
$[0.11,\ 0.21)$	1934	0.00718
$[0.21,\ 0.31)$	2003	0.00877
$[0.31,\ 0.41)$	1976	0.00995
$[0.41,\ 0.51)$	2020	0.01061
$[0.51,\ 0.60)$	2033	0.01146
$[0.60,\ 0.70)$	2014	0.01134
$[0.70,\ 0.80)$	1995	0.01226
$[0.80,\ 0.90)$	1993	0.01285
$[0.90,\ 1.00)$	1937	0.01207
Overall	20,000	0.01004

The MAE profile across α bins follows a physically interpretable pattern. For $\alpha \in [0.01, 0.11)$ (slow mean reversion, long half-life, strong autocorrelation signal with $\rho_1 = 1 - \alpha$ close to 1), MAE $= 0.00484$. Error increases monotonically as α approaches 1, where the process approaches i.i.d. white noise and the autocorrelation signal weakens. The slight recovery at $\alpha \in [0.90, 1.00)$ reflects the output boundary at $\alpha = 1.00$: the sigmoid activation prevents the estimator from overshooting beyond the training range, truncating positive errors in this bin.

Table 4 stratifies the same test set by κ_ε and reveals a striking complementary pattern: TCN accuracy *improves* with increasing tail weight, from MAE $= 0.01474$ at $\kappa_\varepsilon \in [3, 5)$ to MAE $= 0.00765$ at $\kappa_\varepsilon \in [15, 17)$. This pattern is consistent with the conjecture that heavier-tailed innovations produce occasional large deviations that make the geometric decay pattern more locally detectable; in any case, the improvement is established empirically and is not an artefact of the training distribution.

Table 4. TCN test-set MAE stratified by κ_ε bin (same $N_{\text{test}} = 20{,}000$ series as Table 3). MAE decreases monotonically as tail weight increases.

Bin (κ_ε)	***N***	**MAE**
$[3,\ 5)$	2815	0.01474
$[5,\ 7)$	2853	0.01223
$[7,\ 9)$	2817	0.01033
$[9,\ 11)$	2879	0.00929
$[11,\ 13)$	2868	0.00874
$[13,\ 15)$	2873	0.00796
$[15,\ 17)$	2895	0.00765

4. Benchmark Comparison

4.1. AR(1)-GARCH(1,1)

The AR(1)-GARCH(1,1) model [11] specifies

$$X_{t+1} = (1-\alpha)\, X_t + \varepsilon_t, \tag{11}$$

$$\varepsilon_t = \sigma_t\, z_t, \quad z_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \tag{12}$$

$$\sigma_t^2 = \omega + a\, \varepsilon_{t-1}^2 + b\, \sigma_{t-1}^2, \tag{13}$$

with $\omega > 0$, $a, b \geq 0$, and $a + b < 1$ for stationarity. All parameters (α, ω, a, b) are estimated jointly by maximum likelihood using the Gaussian conditional density (12).

The limitation of this estimator for our purposes is that the likelihood function couples α with the GARCH variance parameters. When innovations are truly non-Gaussian, the MLE remains consistent for the pseudo-true parameter vector, which differs from the true α unless the model is correctly specified.

4.2. AR(1)-Jump-Diffusion

The jump-diffusion estimator augments the AR(1) model with a compound Poisson jump component [12,29,30]:

$$X_{t+1} = (1-\alpha)\, X_t + \varepsilon_t, \tag{14}$$

$$\varepsilon_t = \sigma\, z_t + \sum_{j=1}^{N_t} \eta_{t,j}, \quad z_t \sim \mathcal{N}(0,1), \quad N_t \sim \text{Poisson}(\lambda), \quad \eta_{t,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu_J, \sigma_J^2), \tag{15}$$

where z_t, N_t, and $\{\eta_{t,j}\}$ are mutually independent. The marginal density of ε_t is an infinite Gaussian mixture: conditional on $N_t = n$, $\varepsilon_t \sim \mathcal{N}(n\mu_J, \sigma^2 + n\sigma_J^2)$ with mixture weight $e^{-\lambda}\lambda^n/n!$; for estimation the sum is truncated at $N_{\text{tr}} = 10$ jump events per step. The parameter vector $(\alpha, \sigma, \lambda, \mu_J, \sigma_J)$ is estimated by maximising the resulting truncated-mixture log-likelihood. Although (15) generates heavier tails than the pure GARCH model, the Gaussian base component and the Gaussian jump size distribution still constrain the achievable kurtosis and asymmetry profiles, leading to distributional misspecification for highly non-Gaussian series.

4.3. AR(1)-Regime-Switching

The regime-switching estimator models the innovation variance as a function of a latent Markov state $s_t \in \{0, 1\}$ [13]:

$$X_{t+1} = (1-\alpha)\, X_t + \sigma_{s_t}\, z_t, \quad z_t \sim \mathcal{N}(0,1), \tag{16}$$

$$s_t \sim \text{Markov}(P), \tag{17}$$

where P is the 2×2 transition probability matrix and $\sigma_0 \neq \sigma_1$ allows heteroskedasticity across regimes. We impose a single mean-reversion coefficient α shared across regimes and allow regime-specific innovation variances $\sigma_0 \neq \sigma_1$. This specification is motivated by the following rationale: the regime structure captures the distributional heterogeneity of the innovations (low- and high-volatility states) without requiring the introduction of regime-specific mean-reversion speeds, which would require an aggregation step that obscures the interpretation of $\hat{\alpha}$. Estimation uses the Hamilton filter with the EM algorithm.

4.4. Multi-Lag Yule–Walker (ACF-YW)

To isolate the contribution of deep-architecture features from pure multi-lag aggregation, we include a distribution-free closed-form baseline: the multi-lag Yule–Walker

estimator (ACF-YW). It fits the geometric ACF decay $\rho_k = (1-\alpha)^k$ to the full empirical ACF profile by nonlinear least squares:

$$\hat{\alpha}_{\text{ACF-YW}} = \arg\min_{\alpha \in [0.01,\,1]} \sum_{k=1}^{K} \left(\hat{\rho}_k - (1-\alpha)^k\right)^2, \qquad K = \min(\lfloor T/4 \rfloor, 100), \tag{18}$$

where $\hat{\rho}_k$ is the sample autocorrelation at lag k. This one-dimensional bounded optimisation generalises the lag-1 Yule–Walker estimator. Like the standard Yule–Walker estimator, ACF-YW is distribution-free; unlike it, ACF-YW exploits all lags up to K. Including ACF-YW in the comparison makes it possible to distinguish multi-lag aggregation (which ACF-YW also performs) from the additional TCN advantages of finite-sample adaptation and adaptive lag-weighting of the ACF profile.

4.5. Comparison Results

Panel (b) of Figure 3 compares all five estimators across kurtosis bins.

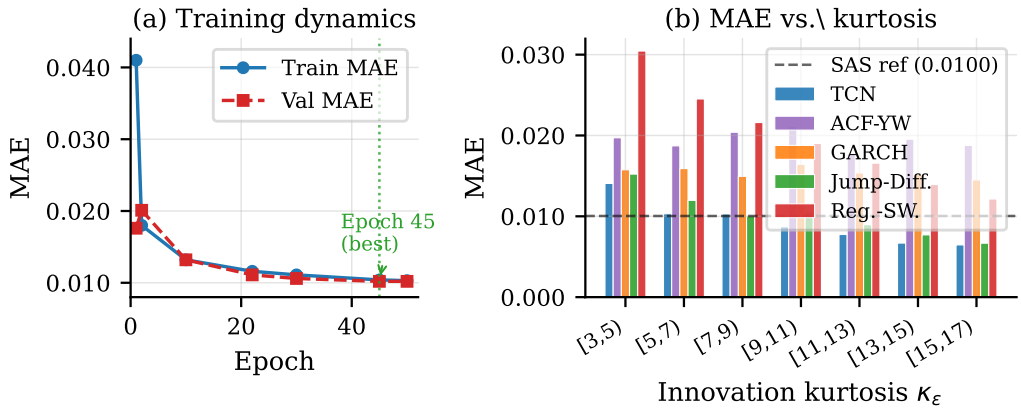

Figure 3. (**a**) Training and validation MAE over 50 epochs (κ_ε-uniform training). The best checkpoint is selected at epoch 45 (val MAE = 0.01017). (**b**) MAE versus κ_ε for all five estimators ($N = 200$ series per bin). The ACF-YW bar (purple) is essentially flat across all bins, while TCN (blue) improves monotonically. The dashed line marks the SAS in-distribution reference MAE (0.01004).

Table 5 reports the benchmark comparison stratified by κ_ε (bin labels computed at $\nu = 0$, see Section 2.2; N = 1400 series, 200 per bin, $\kappa_\varepsilon \sim \text{Uniform}(3, 17)$ within each bin, $\alpha \sim \text{Uniform}(0.01, 1.00)$, $\nu \sim \text{Uniform}(-0.50, +0.50)$). All five estimators are applied to the same set of synthetic series: the TCN produces $\hat{\alpha}$ in a single forward pass, while each classical estimator is re-estimated from scratch on each individual series via an independent maximum-likelihood or EM optimisation, with no access to the true parameter values or the generating distribution. The TCN achieves an overall MAE of 0.00918, outperforming ACF-YW (0.01944, >2× worse), GARCH(1,1) (0.01545, 68.3% worse), Jump-Diffusion (0.01017, 10.8% worse), and Regime-Switching (0.01975, 115.1% worse).

The kurtosis-stratified comparison reveals the mechanism underlying the TCN's advantage. For near-Gaussian innovations ($\kappa_\varepsilon \in [3, 5)$), all models perform similarly and the TCN (0.01407) already achieves the lowest MAE, though the margin over Jump-Diffusion (0.01522) and GARCH (0.01575) is modest: in this regime, classical estimators with Gaussian-family likelihoods are naturally competitive. As tail weight increases, GARCH's MAE remains nearly constant (≈ 0.015)—confirming its insensitivity to kurtosis—while both the TCN and Jump-Diffusion improve. This pattern is precisely what the κ_ε-uniform training protocol is designed to exploit: the TCN's accuracy continues to improve as tail weight increases—consistent with the conjectured signal-amplification mechanism discussed in Section 1.4—whereas classical estimators' likelihoods are increasingly misspecified.

Table 5. MAE comparison of TCN vs. ACF-YW and classical estimators stratified by κ_ε (bin labels computed at $\nu = 0$, see Section 2.2). $N = 200$ series per bin ($\alpha \sim \text{Uniform}(0.01, 1.00)$, $\nu \sim \text{Uniform}(-0.50, +0.50)$, $T = 1826$). Bold: best result per row. ACF-YW: multi-lag Yule–Walker estimator (see text).

Bin (κ_ε)	TCN	ACF-YW	GARCH(1,1)	Jump-Diff.	Regime-SW.
[3, 5)	**0.01407**	0.01972	0.01575	0.01522	0.03045
[5, 7)	**0.01031**	0.01872	0.01590	0.01198	0.02452
[7, 9)	**0.01029**	0.02040	0.01495	0.01005	0.02160
[9, 11)	**0.00871**	0.02116	0.01643	0.01057	0.01901
[11, 13)	**0.00775**	0.01775	0.01536	0.00896	0.01658
[13, 15)	**0.00669**	0.01954	0.01523	0.00771	0.01393
[15, 17)	**0.00646**	0.01878	0.01450	0.00668	0.01214
Overall	**0.00918**	0.01944	0.01545	0.01017	0.01975

The regime-switching specification adopted here—two Gaussian states, single shared α—is a parsimonious baseline [13]. More elaborate specifications could reduce estimation error but would compound model dependence: additional regimes require information criteria that embed distributional assumptions, and regime-specific AR coefficients require an aggregation convention to yield a single $\hat{\alpha}$. The TCN requires none of these choices.

On the design of the comparison. The comparison is symmetric in the following sense: at inference time, every estimator—including the TCN—observes only the test series and has no knowledge of the true α or of the generating distribution. The TCN is trained once on diverse synthetic data and then applied as a fixed analytical instrument; the classical estimators are refitted independently from scratch on each individual test series. Neither has an informational advantage over the other at the point of estimation. One might nonetheless ask whether comparing the TCN (trained on AR(1) series) against classical estimators on AR(1) test data places the classical models at an inherent disadvantage, since they estimate parameters for features (GARCH volatility dynamics, jump intensity, regime assignments) that are not present in pure AR(1) data. This concern does not hold in practice: classical MLE estimators are consistent—not biased by over-parameterisation—when the true model is a special case of the estimated one. Under i.i.d. innovations, GARCH MLE concentrates the GARCH coefficients near zero and converges to the OLS AR(1) estimate; similarly for Jump-Diffusion and Regime-Switching. The performance deficit of classical estimators therefore does not arise from having too many parameters, but from the way heavy-tailed observations are processed: under innovation kurtosis $\kappa_\varepsilon > 3$, the non-AR distributional components of each classical model partially absorb the large realisations—attributing them to conditional volatility, Poisson jumps, or regime transitions—thereby draining the AR coefficient and biasing $\hat{\alpha}$ downward. This distributional-absorption mechanism, and not a parameterisation penalty, is the source of the classical estimators' degradation documented in Table 5. The AR(1) setting is in this respect the *minimal* environment in which this mechanism manifests: in real data with genuine higher-order dynamics, additional sources of misspecification bias would compound the effect.

The ACF-YW estimator (18) is also distribution-free and already performs multi-lag aggregation. Its inclusion in Table 5 therefore allows us to isolate the three TCN advantages over ACF-YW identified in Section 1.4: adaptive lag-weighting, finite-sample adaptation, and robustness under heavy tails. The results are unambiguous. ACF-YW achieves an overall MAE of 0.01944—more than 2× the TCN's 0.00918—and, critically, its MAE is essentially *flat* across kurtosis bins, ranging only from 0.01775 to 0.02116 with no systematic trend. The TCN, by contrast, improves monotonically from 0.01407 at $\kappa_\varepsilon \in [3, 5)$ to 0.00646 at $\kappa_\varepsilon \in [15, 17)$—a 2.2× improvement over the same range. This divergence directly

demonstrates what ACF-YW cannot do: it weights all lags uniformly in the least-squares objective (18), so it cannot exploit any signal-amplification effect of heavy tails—the conjectured mechanism whereby large deviations facilitate detection of the geometric decay pattern. The TCN, trained on finite-T series across a wide range of $(\alpha, \kappa_\varepsilon)$ combinations, learns a data-adaptive weighting of the autocorrelation decay profile that concentrates on the most informative lags for the given sample size and tail weight. The gap between ACF-YW and TCN therefore quantifies the value of finite-sample adaptation and adaptive lag-weighting beyond what classical multi-lag aggregation already achieves.

As a complementary evaluation, Table 6 reports the overall MAE of the five estimators on 1000 series stratified by mean-reversion speed α (100 per bin, $\kappa_\varepsilon \sim \text{Uniform}(3, 17)$). The TCN and Jump-Diffusion achieve similar overall MAE, with the TCN's slight advantage concentrated at higher α values where the distributional misspecification of the jump model is most consequential.

Table 6. Overall MAE of the five estimators on α-stratified series (N = 1000, 100 per bin, $\alpha \sim \text{Uniform}(0.01, 1.00)$, $\kappa_\varepsilon \sim \text{Uniform}(3, 17)$, $\nu \sim \text{Uniform}(-0.50, +0.50)$, T = 1826). Bold: best result.

TCN	ACF-YW	GARCH(1,1)	Jump-Diff.	Regime-SW.
0.00944	0.01944	0.01517	0.00965	0.01907

Crucially, the differential MAE profiles in Table 5 imply that the three classical estimators systematically produce different $\hat{\alpha}$ values from the same series—not as sampling noise, but as the empirical manifestation of their model-conditioned nature. On synthetic data with known ground truth, these differences are quantifiable as differential biases. This is directly demonstrated on real data in Section 6, where the three estimators disagree without any principled way to adjudicate between them. The TCN, extracting α from autocorrelation structure alone, provides a distribution-free reference that is not subject to this ambiguity.

5. Robustness to Distribution Shift

A potential objection to the distribution-free claim is that the TCN, while making no distributional assumption at inference time, was trained on SAS innovations and might have learnt SAS-specific features. To address this concern, we evaluate the trained TCN on synthetic AR(1) series whose innovations follow three distribution families that were never seen during training:

- **Student-t**: power-law tails, kurtosis $\kappa_\varepsilon = 3 + 6/(d-4)$ for degrees of freedom $d > 4$.
- **Generalised Error Distribution (GED)**: exponential tails,

$$\kappa_\varepsilon = \frac{\Gamma(5/\beta)\,\Gamma(1/\beta)}{[\Gamma(3/\beta)]^2},$$

  where $\Gamma(\cdot)$ is the Euler gamma function and $\beta > 0$ is the shape parameter; $\beta = 2$ recovers the Gaussian, $\beta = 1$ the Laplace distribution.
- **Gaussian mixture**: $\eta \cdot \mathcal{N}(0, 1) + (1-\eta) \cdot \mathcal{N}(0, \sigma_2^2)$ with $\eta = 0.9$, producing jump-like heavy tails without power-law decay.

All three families are symmetric ($\gamma_\varepsilon = 0$), providing a direct comparison against the symmetric ($\nu = 0$) boundary of the SAS training distribution. For each family, we test kurtosis values $\kappa_\varepsilon \in \{6, 9, 15\}$ (within the training range $[3, 17]$) and $\kappa_\varepsilon = 20$ (out-of-distribution, OOD), yielding $4 \times 3 = 12$ base configurations (N = 1000 AR(1) series each, $\alpha \sim \text{Uniform}(0.01, 1.00)$, T = 1826, 12,000 series total). To stress-test performance at extreme kurtosis—as encountered in some energy markets (see Section 6)—we additionally evaluate

Student-t and GED at $\kappa_\varepsilon \in \{30, 45\}$ and Gaussian mixture at $\kappa_\varepsilon = 28$ (the family's kurtosis supremum with $\eta = 0.9$ is $3/(1-\eta) = 30$, approached asymptotically as $\sigma_2 \to \infty$ but never reached), adding five extreme out-of-distribution configurations (5000 additional series). In all cases, the distribution parameter is obtained by numerical inversion of the family's kurtosis formula: Student-t uses $d = 4 + 6/(\kappa_\varepsilon - 3)$; GED uses numerical inversion of the kurtosis formula in β; Gaussian mixture uses numerical inversion of the kurtosis formula in σ_2 with $\eta = 0.9$. Results are reported in Table 7.

Table 7. TCN and ACF-YW MAE on out-of-distribution innovation families (TCN trained on SAS only). $\kappa_\varepsilon \geq 20$ is outside the training range $\kappa_\varepsilon \in [3, 17]$ (marked OOD); rows with $\kappa_\varepsilon \in \{30, 45\}$ for Student-t/GED and $\kappa_\varepsilon = 28$ for Gaussian mixture are extreme OOD configurations. Reference: SAS test set MAE = 0.01004 (TCN). The parameter of each family is obtained by numerical inversion of the kurtosis formula.

Family	Parameter	κ_ε	In Training	TCN	ACF-YW
Student-t	$d = 6.000$	6	yes	0.01451	0.01969
Student-t	$d = 5.000$	9	yes	0.01397	0.01847
Student-t	$d = 4.500$	15	yes	0.01335	0.01956
Student-t	$d = 4.353$	20	OOD	0.01344	0.01932
Student-t	$d = 4.222$	30	OOD	0.01330	0.01958
Student-t	$d = 4.143$	45	OOD	0.01282	0.01933
GED	$\beta = 1.000$	6	yes	0.01231	0.01959
GED	$\beta = 0.778$	9	yes	0.01097	0.01918
GED	$\beta = 0.610$	15	yes	0.00983	0.01925
GED	$\beta = 0.544$	20	OOD	0.00844	0.01867
GED	$\beta = 0.472$	30	OOD	0.00810	0.01880
GED	$\beta = 0.417$	45	OOD	0.00787	0.01916
Gaussian mix	$\sigma_2 = 2.449$	6	yes	0.01374	0.01892
Gaussian mix	$\sigma_2 = 3.149$	9	yes	0.01319	0.01862
Gaussian mix	$\sigma_2 = 4.583$	15	yes	0.01045	0.01862
Gaussian mix	$\sigma_2 = 6.279$	20	OOD	0.00876	0.01942
Gaussian mix	$\sigma_2 = 15.997$	28	OOD	0.00803	0.01856

The theoretical framework of Section 1.3 leads to two precise predictions and one empirical regularity. The Yule–Walker invariance $\rho_k = (1-\alpha)^k$—invariant to distribution family and kurtosis—implies: (i) the TCN should generalise stably to any distribution family not seen during training, since the autocorrelation signal it reads is invariant across families; (ii) performance should not exhibit a discontinuous drop at the training boundary $\kappa_\varepsilon = 17$, because that boundary reflects the choice of the training range and not any structural discontinuity in the autocorrelation mechanism. A third, empirical regularity—not derived from spectral invariance but consistently observed in the benchmark results—is that (iii) accuracy tends to *improve* with increasing κ_ε, consistent with the conjectured signal-amplification mechanism of Section 1.4.

Table 7 yields three findings that confirm all three predictions. First, across all three families the TCN's MAE remains in the range $[0.00803, 0.01451]$, comparable to the reference SAS test MAE of 0.01004. No distribution family causes catastrophic degradation, confirming that the model has not overfit the SAS parametric form. Second, at $\kappa_\varepsilon = 15$ the GED achieves MAE = 0.00983, already below the overall SAS reference of 0.01004; Gaussian mixture achieves 0.01045, comparable to but marginally above the reference. This result is consistent with the empirical regularity observed in Table 4: heavier-tailed innovations, regardless of the distribution family, tend to produce accuracy gains for the TCN, consistent with the conjectured signal-amplification mechanism. Third, the OOD point ($\kappa_\varepsilon = 20$, outside the training boundary $\kappa_\varepsilon = 17$) does not produce a discontinuous

degradation: Student-t MAE at $\kappa_\varepsilon = 20$ (0.01344) is virtually identical to $\kappa_\varepsilon = 15$ (0.01335), and GED and Gaussian mixture actually improve further. Most strikingly, this pattern persists to extreme kurtosis: at $\kappa_\varepsilon = 45$—more than 2.6× beyond the training boundary $\kappa_\varepsilon = 17$—Student-t MAE is 0.01282, GED MAE is 0.00787, and Gaussian mixture at $\kappa_\varepsilon = 28$ achieves 0.00803, all below or comparable to the SAS reference (0.01004). These findings confirm that extreme innovation kurtosis does not cause performance collapse, and that any divergence between the TCN and classical estimators at high κ_ε reflects the structural mechanism identified in Section 4, not numerical instability. The OOD robustness follows from two complementary mechanisms acting in concert: Yule–Walker invariance guarantees that the autocorrelation structure $\rho_k = (1-\alpha)^k$ is mathematically identical at any κ_ε, so the signal the TCN reads is invariant across distribution families; and the conjectured signal-amplification effect—whereby rare large excursions under heavy tails make the mean-reversion pattern more locally detectable—is consistent with performance continuing to improve well beyond the training boundary.

The ACF-YW comparison in Table 7 reinforces the benchmark finding. Across all three OOD families and all kurtosis values, ACF-YW MAE is essentially flat in the range $[0.018, 0.021]$—virtually unchanged from its in-distribution performance and insensitive to both innovation family and kurtosis. The TCN, by contrast, improves with kurtosis within each family and achieves MAE as low as 0.00787 (GED, $\kappa_\varepsilon = 45$) and 0.00803 (Gaussian mixture, $\kappa_\varepsilon = 28$), more than 2× below ACF-YW. This pattern holds without exception across all 17 OOD configurations: the TCN's OOD advantage over ACF-YW is at least as large as its in-distribution advantage, confirming that the superiority comes from adaptive lag-weighting—a property that transfers to any innovation distribution by virtue of spectral invariance—and not from SAS-specific feature learning.

The monotone improvement of TCN accuracy with kurtosis is not an artefact of the MAE loss function: as a diagnostic check, we verified that the same monotone improvement is observed under the L2 (MSE) metric, ruling out the possibility that the trend is a consequence of the L_1 training objective.

A natural question is whether extending the training range beyond $\kappa_\varepsilon = 17$ would further improve performance at extreme tail weights. The spectral invariance result (Section 1.3) answers this question at the theoretical level: since the autocorrelation function $\rho_k = (1-\alpha)^k$ depends only on α regardless of κ_ε, the signal the TCN reads is mathematically identical at any tail weight. What matters is not that the training range covers any particular set of kurtosis values, but that it is broad enough to prevent any distributional feature from being a reliable predictor of α during training—thereby forcing the network to learn the autocorrelation pattern alone. The range $\kappa_\varepsilon \in [3, 17]$ achieves this. Moreover, the empirical regularity of improving accuracy with kurtosis—consistent with the conjectured signal-amplification mechanism—suggests that extending the training range would not be beneficial: if anything, performance tends to improve beyond the training boundary, as Tables 4 and 7 confirm. The present robustness analysis therefore does not reveal a limitation to be remedied by an extended training range; it provides empirical confirmation of what spectral invariance predicts theoretically.

A further robustness question concerns the *temporal structure* of the innovations rather than their marginal distribution. The spectral invariance result (Section 1.3) guarantees that the autocorrelation function $\rho_k = (1-\alpha)^k$ is preserved whenever $\mathbb{E}[\varepsilon_t \varepsilon_{t-k}] = 0$ for all $k \neq 0$—a condition satisfied by GARCH innovations. To verify this guarantee empirically, we evaluate the TCN on AR(1)-GARCH(1,1) series (see Equations (11)–(13) in Section 4.1) with $\omega = 1 - a - b$ (unconditional unit variance) and Gaussian innovations $z_t \sim \mathcal{N}(0,1)$. We vary the GARCH persistence $\rho = a + b$ across seven levels from 0 (i.i.d. Gaussian) to 0.99 (near-integrated GARCH), fixing $a = 0.1\rho$ and $b = 0.9\rho$. For each level, $N = 200$

AR(1)-GARCH series are generated with $\alpha \sim \text{Uniform}(0.01, 1.00)$ and $T = 1826$; the trained TCN is applied without any modification.

Table 8 reports the results for both the TCN and ACF-YW, applied to the same series. The TCN MAE remains in the range $[0.017, 0.022]$ across all persistence levels—a variation of at most 23% relative to the i.i.d. Gaussian baseline ($\rho = 0$, MAE $= 0.018$)—and shows no systematic trend with ρ. At $\rho = 0.99$, where the GARCH innovation kurtosis reaches $\kappa_\varepsilon \approx 200$ (far beyond the SAS training range $[3, 17]$), the TCN MAE is 0.022, virtually unchanged from the baseline. This confirms that volatility clustering—even in its most extreme near-integrated form—does not destabilise the estimator, in direct agreement with the spectral invariance guarantee: GARCH innovations satisfy the zero-autocorrelation condition by construction, so the autocorrelation pattern $\rho_k = (1 - \alpha)^k$ is undistorted regardless of the variance dynamics.

The ACF-YW estimator, which is also distribution-free and theoretically guaranteed by spectral invariance, behaves differently: its MAE is stable and comparable to the TCN for $\rho \leq 0.95$ (range $[0.020, 0.022]$), but degrades noticeably at high persistence (0.024–0.026 for $\rho \geq 0.97$). This divergence is consistent with the finite-sample mechanism identified in Section 1.4: at high GARCH persistence, conditional heteroscedasticity inflates the variance of sample autocorrelation estimates, increasing the noise in the equal-weight sum-of-squares objective of ACF-YW. The TCN's data-adaptive lag-weighting implicitly compensates for this heteroscedastic noise, maintaining stable accuracy where ACF-YW degrades. The result reinforces the finding of Table 5: the TCN's advantage over ACF-YW stems not from multi-lag aggregation (which ACF-YW already performs) but from adaptive weighting—a property that remains beneficial beyond the distributional robustness setting and extends to temporal variance structure as well.

Table 8. TCN and ACF-YW MAE on AR(1)-GARCH(1,1) series with varying persistence $\rho = a + b$ ($a = 0.1\rho$, $b = 0.9\rho$, $\omega = 1 - a - b$; $N = 200$ per level, $\alpha \sim \text{Uniform}(0.01, 1.00)$, $T = 1826$; both estimators applied to the same series). κ_ε: Pearson kurtosis of GARCH(1,1) innovations (Equations (11)–(13)) with Gaussian z_t. The finite-kurtosis condition $2a^2 < 1 - \rho^2$ is violated at $\rho^* = 1/\sqrt{1.02} \approx 0.990$, so κ_ε grows rapidly as $\rho \to \rho^*$ and the $\rho = 0.99$ row constitutes an extreme stress test. Neither estimator is modified or retrained.

$\rho = a + b$	κ_ε	TCN MAE	ACF-YW MAE
0.00	3.0	0.01776	0.02032
0.30	3.0	0.01882	0.01968
0.50	3.0	0.01834	0.02211
0.70	3.1	0.01717	0.02053
0.90	3.3	0.01931	0.02170
0.95	3.7	0.02044	0.02157
0.97	4.4	0.02002	0.02443
0.98	5.8	0.02165	0.02443
0.99	200.3	0.02181	0.02565

A separate robustness question concerns order misspecification: what does the TCN return when the true data-generating process is AR(p) with $p > 1$? In this case the autocorrelation structure does not follow the geometric decay $(1 - \alpha)^k$, and the TCN will fit the best AR(1) approximation to the observed autocorrelation pattern, returning the α whose geometric decay profile most closely matches the empirical ACF in the least-squares sense implicitly learned during training. The result is interpretable as the effective AR(1) mean-reversion speed of the series and remains free of distributional assumptions; however, the user should be aware that the estimate may not recover any individual AR(p) coefficient and that an AR(p) specification test is advisable before applying the estimator. For en-

ergy commodity prices, the AR(1) discrete-time representation of the Ornstein–Uhlenbeck process is the canonical specification in the literature [1,4,5], so this concern is largely academic in the application context of this paper; nonetheless it is a genuine limitation for applications where higher-order dynamics are suspected.

6. Real Data Application

Applying the trained estimator to real price data requires a preprocessing step that maps raw prices onto stationary residuals conforming to the AR(1) structure of the synthetic training data. Each raw daily price series $\{P_t\}$ is transformed as follows: the log-price $\log P_t$ is decomposed via LOESS smoothing [31] (bandwidth $h = 0.1$) into a slowly-varying trend $\hat{T}_t$ and a stationary residual $X_t = \log P_t - \hat{T}_t$. The bandwidth $h = 0.1$ specifies that each local polynomial fit uses the nearest 10% of observations—approximately 120–180 days depending on the market—defining a smoothing window of roughly six months that captures the long-term non-stationary component while preserving mean-reverting dynamics at shorter timescales. The residual X_t is then standardised (zero mean, unit variance) to yield $\tilde{X}_t$, which is passed to the TCN. This two-step chain is applied identically to all four markets.

The time-scale separation between the LOESS trend (smoothing window ≈ 180 days) and the mean-reversion half-lives (2–7 days across all four markets) ensures that the trend cannot absorb mean-reverting fluctuations. To verify empirically, we evaluated all four estimators at $h \in \{0.05, 0.10, 0.15, 0.20\}$, corresponding to local fitting windows of approximately 90 to 365 days. The TCN estimates vary by at most 0.005 (PSV), 0.009 (PUN), 0.012 (HH), and 0.008 (PJM) across this bandwidth range; the classical estimators display similar insensitivity. All qualitative conclusions are unchanged. The baseline $h = 0.1$ is retained as it strikes the best balance between trend capture and short-term signal preservation.

We apply the TCN estimator and the three classical benchmarks to four energy commodity markets: Italian natural gas (PSV price, Punto di Scambio Virtuale, quoted in €/MWh) and Italian electricity (PUN price, Prezzo Unico Nazionale, €/MWh) (calendar-daily prices, 2019–2023, $N = 1826$) and US Henry Hub natural gas ($/MMBtu) and PJM West Hub electricity ($/MWh) (business-daily EIA prices, 2019–2023, $N \approx 1225$–1252). Figure 4 shows the LOESS-detrended log-price series X_t for all four markets. Table 9 summarises the first four moments of the first differences ΔX_t; Table 10 reports the mean-reversion speed estimates $\hat{\alpha}$ and the implied half-lives for all estimators.

Table 9. Descriptive statistics of first differences $\Delta X_t = X_{t+1} - X_t$ of the LOESS-detrended log-prices ($T - 1$ differences). σ: standard deviation; γ: skewness; κ: kurtosis (not excess). * HH raw $\kappa = 44.8$; structural kurtosis ≈ 9.3 excluding the Winter Storm Uri spike sequence (February 2021).

Market	T	σ	γ	κ
PSV (IT gas price)	1826	0.0707	0.450	14.2
PUN (IT elec. price)	1826	0.1467	0.317	5.4
HH (US gas) *	1252	0.0768	−0.785	44.8
PJM (US power)	1225	0.1968	0.127	10.9

The four markets span a wide range of log-return kurtosis profiles: from near-Gaussian PUN ($\kappa = 5.4$) to the extreme HH ($\kappa = 44.8$), providing a structurally diverse real-data validation. The HH raw kurtosis is dominated by the Winter Storm Uri spike sequence (February 2021); excluding those observations yields a structural $\kappa \approx 9.3$ (see the HH discussion below for full interpretation).

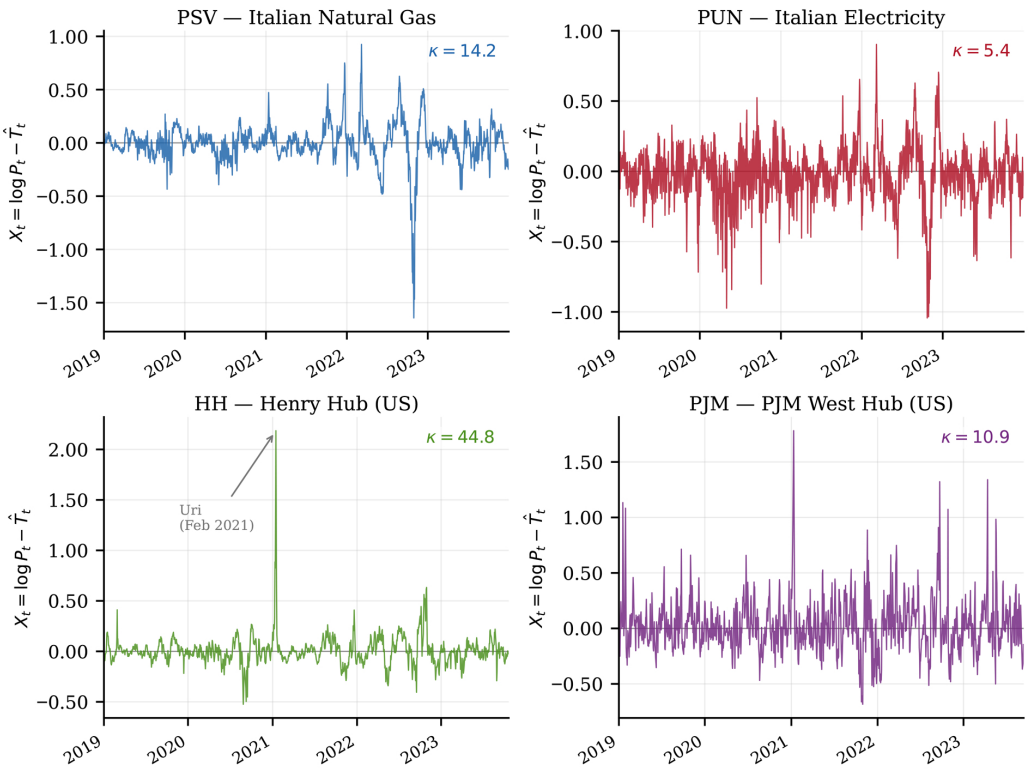

Figure 4. Detrended log-price series $X_t = \log P_t - \hat{T}_t$ for all four markets (LOESS bandwidth $h = 0.1$, period 2019–2023). Top row: Italian markets (PSV natural gas price, $\kappa = 14.2$; PUN electricity price, $\kappa = 5.4$). Bottom row: US markets (Henry Hub gas, $\kappa = 44.8$; PJM West Hub electricity, $\kappa = 10.9$). All series are approximately zero-mean and stationary, conforming to the AR(1) structure of the synthetic training data. The large spike in HH (February 2021) corresponds to Winter Storm Uri. Kurtosis computed on first differences ΔX_t.

Table 10. Mean-reversion speed estimates $\hat{\alpha}$ and implied half-life $\tau_{1/2} = \ln 2/\hat{\alpha}$ (trading days, in parentheses) for all four estimators applied to Italian and US energy markets (preprocessing: Section 6). κ: kurtosis of log-returns ΔX_t. * HH: raw $\kappa = 44.8$; structural $\kappa \approx 9.3$ excluding the Winter Storm Uri spike sequence (February 2021); see text for interpretation.

Estimator	**PSV ($\kappa = 14.2$) Gas (IT), $T = 1826$**	**PUN ($\kappa = 5.4$) Power (IT), $T = 1826$**	**HH * ($\kappa = 44.8$) Gas (US), $T = 1252$**	**PJM ($\kappa = 10.9$) Power (US), $T = 1225$**
TCN (ours)	0.0975 (7.1 d)	0.2857 (2.4 d)	0.1996 (3.5 d)	0.3540 (2.0 d)
GARCH(1,1)	0.0834 (8.3 d)	0.3007 (2.3 d)	0.1245 (5.6 d)	0.3585 (1.9 d)
Jump-diffusion	0.0421 (16.5 d)	0.2745 (2.5 d)	0.1114 (6.2 d)	0.3517 (2.0 d)
Regime-switching	0.0555 (12.5 d)	0.2479 (2.8 d)	0.1094 (6.3 d)	0.3189 (2.2 d)

The distribution-free nature of the TCN estimate enables a natural two-stage approach to the complete characterisation of AR(1) dynamics. In Stage 1, the TCN delivers $\hat{\alpha}$ from the autocorrelation structure of the observed series, without any assumption on the innovation distribution. In Stage 2, the residuals $\hat{\varepsilon}_t = X_{t+1} - (1 - \hat{\alpha})\, X_t$ are available for independent distributional analysis: their sample moments (variance, skewness, kurtosis) can be estimated and any parametric or non-parametric distribution fitted, free from the coupling to α that affects classical estimators. This decoupling is the fundamental structural advantage of the proposed approach: while GARCH, Jump-Diffusion, and Regime-Switching jointly optimise α and distributional parameters through a single likelihood—so that misspecification of the innovation model biases $\hat{\alpha}$—the TCN pipeline separates the two problems and ensures that Stage 2 distributional inference is not contaminated by an incorrect α.

The distributional characterisation of $\hat{\varepsilon}_t$ falls outside the scope of this paper and is left to any chosen method—parametric or non-parametric—without any modification to the Stage 1 pipeline.

Several features of Table 10 merit comment.

For the Italian electricity PUN price ($\kappa = 5.4$), all four estimators agree closely—$\hat{\alpha}$ ranges from 0.248 to 0.301 and implied half-lives span only 2.3–2.8 trading days—consistent with the near-Gaussian innovation distribution and the well-established rapid mean reversion of non-storable electricity.

PJM electricity (US, $\kappa = 10.9$) displays a similar pattern of inter-model agreement, with $\hat{\alpha}$ in $[0.319, 0.358]$ and half-lives of 1.9–2.2 days, confirming rapid mean reversion in a non-storable market.

For the Italian natural gas PSV price ($\kappa = 14.2$), model dependence becomes pronounced. The three classical estimators span $\hat{\alpha} \in [0.042, 0.083]$, corresponding to implied half-lives of 8.3 days (GARCH) versus 16.5 days (Jump-Diffusion)—a factor-of-two discrepancy arising solely from the different distributional assumptions embedded in each likelihood. The TCN estimate $\hat{\alpha} = 0.0975$ (half-life 7.1 days) is model-free and not subject to this ambiguity; it implies that Italian natural gas prices revert to their long-run equilibrium level in approximately one trading week. The economic consequences of this disagreement are material. In risk management, a 10-day Value-at-Risk computed from a Monte Carlo path generator based on the Jump-Diffusion estimate ($\hat{\alpha} = 0.042$, half-life 16.5 days) would produce substantially wider price distributions than one based on the TCN estimate ($\hat{\alpha} = 0.098$, half-life 7.1 days), since slower mean reversion implies greater multi-day drift away from equilibrium. In derivative pricing, swing options and storage contracts on natural gas are highly sensitive to the decay speed of price deviations: a half-life of 16.5 days versus 7.1 days translates directly into a different time value of optionality, affecting the fair value of contracts with weekly or bi-weekly exercise windows [4,5]. The factor-of-two spread across classical estimators means that two practitioners using different but equally defensible models would arrive at materially different contract valuations—a model risk that the TCN, providing a single distribution-free estimate, eliminates.

The most striking case is HH Henry Hub (US, raw $\kappa = 44.8$). The three classical estimators agree narrowly with each other ($\hat{\alpha} \approx 0.11$, inter-model spread $\max \hat{\alpha} - \min \hat{\alpha} = 0.015$) but diverge markedly from the TCN ($\hat{\alpha} = 0.200$). This pattern is diagnostic: the classical models are not disagreeing with each other (which would indicate random sampling noise) but all agree in underestimating α relative to the TCN. The mechanism is the same identified in the kurtosis-stratified synthetic benchmark (Section 4): with $\kappa = 44.8$ dominated by the Winter Storm Uri spike sequence (February 2021), the extreme observations are absorbed by the non-AR components of each classical model—as conditional volatility in GARCH, as Poisson jumps in Jump-Diffusion, and as regime transitions in Regime-Switching—thereby draining the AR coefficient and biasing $\hat{\alpha}$ downward. The TCN, estimating α from the autocorrelation structure alone, is not subject to this redistribution of variance across model components.

We emphasise that the TCN estimate $\hat{\alpha} = 0.200$ is computed on the *full* HH series without removing or adjusting any observations: the Winter Storm Uri spike sequence is not treated as an anomaly to be filtered out. The mention of the structural kurtosis ($\kappa \approx 9.3$ excluding the spike sequence) is provided solely as contextual information for the reader to understand the origin of the extreme raw kurtosis; it has no bearing on the estimation. By spectral invariance, the autocorrelation structure $\rho_k = (1 - \alpha)^k$ depends only on α regardless of innovation kurtosis, so the raw $\kappa = 44.8$ does not impair the TCN estimate, as confirmed by the OOD robustness analysis up to $\kappa_\varepsilon = 45$ (Section 5).

The interpretation that the classical estimators exhibit downward bias on HH is not circular: it is consistent with the distributional-absorption mechanism identified in Section 4, which provides a plausible explanation for why classical estimators tend to underestimate α under heavy-tailed innovations. The direction of the discrepancy on real data is attributed to the same mechanism—not to the TCN being used as a ground truth. We acknowledge that on real data neither estimate is provably correct, and that non-stationarities not captured by the AR(1) model could in principle affect the TCN estimate differently from classical ones. The TCN provides a distribution-free reference, not an absolute ground truth.

The divergences observed in the PSV and HH cases are direct empirical manifestations of the model-conditioned nature of classical $\hat{\alpha}$ identified in Section 1.2. Both cases share the same underlying cause—heavy-tailed innovations causing each classical model to redistribute variance from the AR component to its own distributional mechanism—but present complementary signatures: in PSV, the three classical estimators diverge from each other (spread $= 0.041$), reflecting the fact that different distributional mechanisms absorb the heavy tails differently; in HH, they converge on a common underestimate (spread $= 0.015$) but all diverge from the TCN, revealing a shared systematic bias. The TCN, conditioning $\hat{\alpha}$ on the autocorrelation structure alone, is immune to this redistribution in both cases.

7. Conclusions

This paper has introduced a distribution-free neural estimator of mean-reversion speed based on a Temporal Convolutional Network trained entirely on synthetic AR(1) series. The key design choice is kurtosis-uniform innovation sampling, which prevents distributional bias and allows the network to learn the autocorrelation decay profile $\rho_k = (1-\alpha)^k$ as a stable signal across the full range of tail shapes encountered in practice. On held-out test data the estimator performs best for slow mean-reverting processes, where the autocorrelation signal persists across many lags and multi-lag aggregation is most valuable, and—consistent with the conjectured signal-amplification mechanism discussed in Section 1.4—accuracy improves rather than degrades as innovation tail weight increases.

The controlled benchmark comparison confirms the central theoretical prediction. Classical parametric estimators are competitive in the near-Gaussian regime; as innovation tail weight increases their accuracy degrades while the TCN's improves, so the performance gap widens monotonically with kurtosis. The multi-lag Yule–Walker baseline (ACF-YW), which already exploits the full autocorrelation profile but lacks finite-sample adaptation, shows flat accuracy across all kurtosis bins—isolating adaptive lag-weighting and finite-sample learning as the dominant sources of the TCN's identification gain over purely autocorrelation-based approaches. The distribution-free claim is validated on out-of-distribution innovation families with no performance degradation at the training boundary, and a complementary GARCH robustness experiment empirically confirms the spectral invariance guarantee for the temporal variance structure: TCN accuracy is stable across all tested GARCH persistence levels, including near-integrated configurations far outside the SAS training range, while ACF-YW degrades under strong conditional heteroscedasticity—a further demonstration that the TCN advantage stems from adaptive weighting, not merely from multi-lag aggregation.

Applied to four energy commodity markets spanning Italian and US gas and power prices, the TCN yields physically plausible mean-reversion estimates consistent with the known dynamics of these markets. The empirical results reveal a consistent pattern: classical parametric estimators produce mutually inconsistent $\hat{\alpha}$ values from the same series, with discrepancies that grow with innovation tail weight. The PSV case illustrates the model-dependence most starkly—the three classical estimators disagree by a factor of

two in implied half-life, driven by heavy-tailed log-returns being absorbed differently by each parametric likelihood. The Henry Hub case presents a complementary signature: the three classical estimators converge on a common estimate but all diverge systematically from the TCN, consistent with extreme price events being redistributed from the AR component into the non-AR model components rather than reflected in $\hat{\alpha}$.

The practical implications are significant. A fundamental operational advantage of the TCN is its *train-once, deploy-forever* nature: the model is trained offline once on synthetic data and subsequently applied to any new time series in under one millisecond, with no further optimisation, whereas classical estimators must solve a new numerical maximisation for every series with no convergence guarantee under heavy tails. The one-time training cost is approximately 75 min. Classical estimator run times on a single 1826-point series are approximately 1–2 s (GARCH MLE), 25–30 s (Jump-Diffusion MLE), and 50–60 s (Regime-Switching EM). The break-even number of series—at which the amortised training cost equals the total classical estimation cost—is approximately 3000 series for GARCH (4500 s/1.5 s), 160 series for Jump-Diffusion, and 80 series for Regime-Switching. In a typical portfolio application with hundreds of instruments, rolling estimation windows, or Monte Carlo bootstrap scenarios, the break-even is reached immediately, and each additional inference costs under one millisecond.

A single distribution-free $\hat{\alpha}$ can moreover be used consistently across risk models, derivative pricing engines, and portfolio optimisers without introducing distributional assumptions that would compromise internal consistency—in sharp contrast with the factor-of-two half-life discrepancies that classical estimators generate for heavy-tailed markets, which are large enough to materially affect Value-at-Risk calculations, swing-option prices, and portfolio rebalancing frequencies. The train-once architecture positions the TCN as a *universal mean-reversion estimator*: just as a calibrated measuring instrument is applied to any specimen without recalibration, the trained TCN measures $\hat{\alpha}$ in any price series without market-specific re-fitting. The distribution-free $\hat{\alpha}$ further enables the two-stage pipeline introduced in Section 6: once α is fixed by Stage 1, the residuals $\hat{\varepsilon}_t$ are available for independent distributional modelling in Stage 2, free from the coupling that contaminates classical likelihood-based approaches and enabling a modular treatment of mean-reversion dynamics and innovation distribution.

Beyond the numerical results, the analysis contributes a conceptual clarification with broader implications for energy market modelling. Classical parametric estimators of mean-reversion speed do not estimate a model-free property of the data: they estimate a model-conditioned quantity, whose value is determined jointly by the autocorrelation structure of the series and by the distributional assumptions embedded in the likelihood. Two analysts applying different classical estimators to the same price series will obtain different $\hat{\alpha}$ values not because of sampling variability, but because they are implicitly answering different questions. In the absence of knowledge of the true innovation distribution, there is no principled criterion to adjudicate between them, and any apparent agreement is coincidental. The TCN, conditioning $\hat{\alpha}$ solely on the autocorrelation structure of the series, provides an estimate that is genuinely comparable across markets and models—a prerequisite for any meaningful cross-market comparison of mean-reversion dynamics or for using $\hat{\alpha}$ as an input to models downstream.

Several directions for future research are warranted. First, the assumption of a time-invariant α is a genuine limitation. In practice, energy markets may undergo structural breaks or regime changes—associated with geopolitical crises, regulatory shifts, or prolonged supply disruptions—during which the effective mean-reversion speed may differ substantially from the full-sample estimate. A direct practical response to this concern is rolling estimation: since the TCN delivers $\hat{\alpha}$ in under one millisecond, it can be applied to

overlapping sub-windows of the price series at negligible cost, producing a time series of local estimates $\hat{\alpha}_t$ that reveals changes in mean-reversion dynamics over time. Extending the TCN architecture to explicitly model time-varying α—for instance, by training on piecewise-stationary AR(1) series with regime-switching α—is a natural methodological extension. Second, the framework could be extended to account for structural shock components of the form $X_{t+1} = (1-\alpha)X_t + \beta S_t + \varepsilon_t$, where S_t is an observed or latent shock variable, connecting the statistical estimator with the structural literature on commodity price dynamics [9]. Third, the methodology could be extended to multivariate settings where mean-reversion speeds and cross-commodity correlations are estimated jointly—relevant for spread-option pricing and co-integrated commodity portfolios [32,33]. Fourth, the principle underlying the proposed approach—training a neural estimator on synthetic data with sufficient distributional variation to isolate the target parameter's signature in the temporal structure of the series—could be extended beyond α to other dynamic parameters, such as jump intensity or volatility of volatility, yielding a family of distribution-free neural estimators for energy market calibration.

Author Contributions: Conceptualisation, C.M. and E.M.; methodology, C.M. and E.M.; software, E.M.; validation, C.M. and E.M.; formal analysis, C.M. and E.M.; investigation, C.M. and E.M.; data curation, E.M.; writing—original draft preparation, C.M. and E.M.; writing—review and editing, C.M. and E.M.; visualisation, E.M.; supervision, C.M.; and project administration, C.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The code and synthetic data generation scripts are publicly available at https://github.com/DeepCE/MeanNeural. Original Italian energy market data (PSV, PUN) are available from GME (Gestore dei Mercati Energetici, https://www.mercatoelettrico.org, accessed on 21 May 2025). US energy market data (Henry Hub natural gas and PJM West Hub electricity) are available from the U.S. Energy Information Administration (EIA, https://www.eia.gov, accessed on 8 January 2026).

Conflicts of Interest: Author Emiliano Mari was employed by the company Sydus. The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Moments of AR(1) First Differences

Let $\{X_t\}$ be the stationary AR(1) process in Equation (1) with i.i.d. innovations ε_t having variance σ_ε^2, skewness γ_ε, and Pearson kurtosis κ_ε. The first differences $\Delta X_t = X_{t+1} - X_t = -\alpha X_t + \varepsilon_t$ are a linear combination of X_t (stationary) and ε_t (innovation), which are independent by the AR(1) structure.

Appendix A.1. Cumulant Formula

Since X_t and ε_t are independent, cumulant additivity applies throughout [34]: cumulants of sums of independent random variables add, and scale as $\mathcal{K}_r(cZ) = c^r \mathcal{K}_r(Z)$. Applied to the AR(1) recursion $X_{t+1} = (1-\alpha)X_t + \varepsilon_t$, cumulant additivity gives $\mathcal{K}_r(X_{t+1}) = (1-\alpha)^r \mathcal{K}_r(X_t) + \mathcal{K}_r(\varepsilon_t)$; invoking stationarity $\mathcal{K}_r(X_{t+1}) = \mathcal{K}_r(X_t)$ and solving gives

$$\mathcal{K}_r(X_t) = \frac{\mathcal{K}_r(\varepsilon_t)}{1-(1-\alpha)^r}, \qquad r \geq 1. \tag{A1}$$

Applying the same principle to $\Delta X_t = -\alpha X_t + \varepsilon_t$ gives $\mathcal{K}_r(\Delta X_t) = (-\alpha)^r \mathcal{K}_r(X_t) + \mathcal{K}_r(\varepsilon_t)$, which combined with (A1) gives

$$\mathcal{K}_r(\Delta X_t) = \mathcal{K}_r(\varepsilon_t) \cdot \frac{(-\alpha)^r + 1 - (1-\alpha)^r}{1-(1-\alpha)^r}, \qquad r \geq 1. \tag{A2}$$

Appendix A.2. Mean

Setting $r = 1$ in (A1) gives $\mathcal{K}_1(X_t) = \mathbb{E}[X_t] = \mathbb{E}[\varepsilon_t]/\alpha$, i.e.,

$$\mathbb{E}[X_t] = \frac{\mathbb{E}[\varepsilon_t]}{\alpha}, \tag{A3}$$

which reduces to zero when $\mathbb{E}[\varepsilon_t] = 0$. Setting $r = 1$ in (A2) gives $\mathbb{E}[\Delta X_t] = 0$ for any $\alpha \in (0, 2)$, confirming that first differences ΔX_t are zero-mean regardless of $\mathbb{E}[\varepsilon_t]$.

Appendix A.3. Moment Formulas

Evaluating (A2) for $r = 2, 3, 4$ and converting to standard moments yields:

$$\sigma(\Delta X_t) = \sigma_\varepsilon \sqrt{\frac{2}{2-\alpha}}, \tag{A4}$$

$$\gamma(\Delta X_t) = \gamma_\varepsilon \cdot \frac{3(1-\alpha)(2-\alpha)^{3/2}}{2\sqrt{2}\,(\alpha^2 - 3\alpha + 3)}, \tag{A5}$$

$$\kappa(\Delta X_t) = 3 + (\kappa_\varepsilon - 3) \cdot \frac{(2-\alpha)(2\alpha^2 - 3\alpha + 2)}{2(\alpha^2 - 2\alpha + 2)}. \tag{A6}$$

As $\alpha \to 0$ all three scaling factors tend to unity, confirming that first differences ΔX_t coincide with innovations in the absence of mean reversion. For $\alpha > 0$, the kurtosis scaling factor in (A6) is strictly less than one, so $\kappa(\Delta X_t) < \kappa_\varepsilon$: first-difference kurtosis *understates* innovation kurtosis. Similarly, the skewness scaling factor in (A5) is less than one for $\alpha > 0$ and vanishes at $\alpha = 1$, where $\Delta X_t = \varepsilon_t - \varepsilon_{t-1}$ has zero skewness regardless of γ_ε.

Scope of validity. Equations (A4)–(A6) are exact under the i.i.d. innovation assumption stated at the opening of this appendix. The derivation relies on cumulant additivity, which requires independence between X_t and ε_t; this holds when innovations are i.i.d. but fails for conditionally heteroscedastic processes such as GARCH, where σ_t^2 depends on past values of ε_{t-k} and hence on past X_{t-k}. For GARCH innovations, the variance formula (A4) remains exact (it requires only zero cross-covariance $\mathrm{Cov}(X_t, \varepsilon_t) = 0$, which holds by the AR(1) structure), while the skewness (A5) and kurtosis (A6) formulas become first-order approximations whose accuracy depends on the magnitude of the GARCH persistence. In the context of this paper the formulas are applied in Table A1 using the TCN estimate $\hat{\alpha}$ to invert observed log-return moments into implied innovation moments; since the TCN is trained on i.i.d. SAS innovations, this inversion is used purely for interpretive purposes and the approximation error is inconsequential for the estimation results.

Appendix A.4. Implied Innovation Moments for the Four Markets

Table A1 reports, for each market, the log-return moments from Table 9 alongside the implied innovation moments obtained by inverting Equations (A4)–(A6) using the TCN estimate $\hat{\alpha}$ as a representative value.

Table A1. Log-return moments (from Table 9) and implied innovation moments obtained by inverting Equations (A4)–(A6) using the TCN estimates $\hat{\alpha}$. The implied innovation kurtosis κ_ε falls within the nominal training range for all markets except Henry Hub, whose $\kappa_\varepsilon \approx 54$ reflects the Winter Storm Uri spike; by spectral invariance this has no bearing on the validity of the TCN estimate, as confirmed by the robustness analysis of Section 5.

Market	Log-Return Moments ΔX_t				Implied Innovation Moments ε_t		
	$\hat{\alpha}$	σ	γ	κ	σ_ε	γ_ε	κ_ε
PSV (IT gas price)	0.0975	0.0707	0.450	14.2	0.0690	0.487	15.4
PUN (IT elec. price)	0.2857	0.1467	0.317	5.4	0.1358	0.415	6.2
HH (US gas)	0.1996	0.0768	−0.785	44.8	0.0729	−0.934	54.5
PJM (US power)	0.3540	0.1968	0.127	10.9	0.1785	0.181	14.4

References

1. Weron, R. *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach*; Wiley: Chichester, UK, 2006.
2. Weron, R. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *Int. J. Forecast.* **2014**, *30*, 1030–1081. [CrossRef]
3. De Sanctis, A.; Mari, C. Modeling spikes in electricity markets using excitable dynamics. *Phys. A Stat. Mech. Its Appl.* **2007**, *384*, 547–560.
4. Leung, T.; Lu, K.W. Monte Carlo simulation for trading under a Lévy-driven mean-reverting framework. *Appl. Math. Financ.* **2024**, *30*, 207–230. [CrossRef]
5. Schwartz, E.S. The stochastic behavior of commodity prices: Implications for valuation and hedging. *J. Financ.* **1997**, *52*, 923–973. [CrossRef]
6. Lucheroni, C.; Mari, C. Internal hedging of intermittent renewable power generation and optimal portfolio selection. *Ann. Oper. Res.* **2021**, *299*, 873–893. [CrossRef]
7. Mari, C. Stochastic NPV based vs stochastic LCOE based power portfolio selection under uncertainty. *Energies* **2020**, *13*, 3677. [CrossRef]
8. Geman, H. *Commodities and Commodity Derivatives: Modelling and Pricing for Agriculturals, Metals and Energy*; Wiley: Chichester, UK, 2005.
9. Acemoglu, D.; Carvalho, V.M.; Ozdaglar, A.; Tahbaz-Salehi, A. The network origins of aggregate fluctuations. *Econometrica* **2012**, *80*, 1977–2016. [CrossRef]
10. Gabaix, X. The granular origins of aggregate fluctuations. *Econometrica* **2011**, *79*, 733–772. [CrossRef]
11. Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *J. Econom.* **1986**, *31*, 307–327. [CrossRef]
12. Merton, R.C. Option pricing when underlying stock returns are discontinuous. *J. Financ. Econ.* **1976**, *3*, 125–144. [CrossRef]
13. Hamilton, J.D. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica* **1989**, *57*, 357–384. [CrossRef]
14. Lago, J.; Marcjasz, G.; De Schutter, B.; Weron, R. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Appl. Energy* **2021**, *293*, 116983. [CrossRef]
15. Marcjasz, G.; Narajewski, M.; Weron, R.; Ziel, F. Distributional neural networks for electricity price forecasting. *Energy Econ.* **2023**, *125*, 106843. [CrossRef]
16. Mari, C.; Mari, E. Deep learning based regime-switching models of energy commodity prices. *Energy Syst.* **2023**, *14*, 913–934. [CrossRef]
17. Fein-Ashley, J. A comparison of traditional and deep learning methods for parameter estimation of the Ornstein-Uhlenbeck process. *arXiv* **2024**, arXiv:2404.11526. [CrossRef]
18. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271. [CrossRef]
19. Cranmer, K.; Brehmer, J.; Louppe, G. The frontier of simulation-based inference. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 30055–30062. [CrossRef]
20. Lenzi, A.; Bessac, J.; Rudi, J.; Stein, M.L. Neural networks for parameter estimation in intractable models. *Comput. Stat. Data Anal.* **2023**, *185*, 107762. [CrossRef]
21. Sainsbury-Dale, M.; Zammit-Mangion, A.; Huser, R. Likelihood-free parameter estimation with neural Bayes estimators. *Am. Stat.* **2024**, *78*, 1–14. [CrossRef]
22. Zammit-Mangion, A.; Sainsbury-Dale, M.; Huser, R. Neural methods for amortized inference. *Annu. Rev. Stat. Its Appl.* **2025**, *12*, 311–335. [CrossRef]

23. Gloeckler, M.; Toyota, S.; Fukumizu, K.; Macke, J.H. Compositional simulation-based inference for time series. *arXiv* **2025**, arXiv:2411.02728.
24. Jones, M.C.; Pewsey, A. Sinh-arcsinh distributions. *Biometrika* **2009**, *96*, 761–780. [CrossRef]
25. Jones, M.C.; Pewsey, A. The eschewed sinh-arcsinh *t* distribution. *Stat. Probab. Lett.* **2025**, *228*, 110560. [CrossRef]
26. Kendall, M.G. Note on bias in the estimation of autocorrelation. *Biometrika* **1954**, *41*, 403–404. [CrossRef]
27. Lara-Benitez, P.; Carranza-Garcia, M.; Luna-Romera, J.M.; Riquelme, J.C. Temporal convolutional networks applied to energy-related time series forecasting. *Appl. Sci.* **2020**, *10*, 2322. [CrossRef]
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 20 December 2015; pp. 1026–1034.
29. Mari, C.; Mari, E. Gaussian clustering and jump-diffusion models of electricity prices: A deep-learning analysis. *Decis. Econ. Financ.* **2021**, *44*, 1039–1062. [CrossRef]
30. Mari, C.; Baldassari, C. Ensemble methods for jump-diffusion models of power prices. *Energies* **2021**, *14*, 4404. [CrossRef]
31. Cleveland, W.S. Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.* **1979**, *74*, 829–836. [CrossRef]
32. Benth, F.E.; Koekebakker, S. Pricing of forwards and other derivatives in cointegrated commodity markets. *Energy Econ.* **2015**, *52*, 104–117. [CrossRef]
33. Farkas, W.; Gourier, E.; Huitema, R.; Necula, C. A two-factor cointegrated commodity price model with an application to spread option pricing. *J. Bank. Financ.* **2017**, *77*, 249–268. [CrossRef]
34. Brockwell, P.J.; Davis, R.A. *Time Series: Theory and Methods*, 2nd ed.; Springer: New York, NY, USA, 1991.