Preprints.org

Article

Not peer-reviewed version

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

Carlo Mari * and Emiliano Mari

Posted Date: 13 March 2026

doi: 10.20944/preprints202603.0996.v1

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

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

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, Viterbo, 01100, Italy
[2] Sydus, Orvieto, 05018, Italy
* 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 estimator based on a Temporal Convolutional Network (TCN) trained on synthetic AR(1) series with Sinh-ArcSinh innovations of varying tail weight and asymmetry. The SAS family serves as a training vehicle—not a distributional hypothesis—chosen for its ability to span innovation profiles from near-Gaussian to strongly leptokurtic and skewed through its tail-weight and asymmetry parameters. Because the autocorrelation structure $\rho_k = (1-\alpha)^k$ is invariant to the marginal innovation distribution (Yule-Walker invariance), the TCN learns to extract α from temporal dependence alone, independently of distributional assumptions. On held-out test series the estimator outperforms all three classical estimators across the training innovation kurtosis range, with the advantage growing monotonically with non-Gaussianity. A robustness analysis on three out-of-distribution innovation families confirms stable or improved performance well beyond the training boundary. 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, model-free estimates not conditioned on any distributional hypothesis.

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

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 (or mean reversion) 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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

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. Energy commodities typically exhibit half-lives ranging from several days (electricity) to several weeks (natural gas).

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. Throughout this paper, X_t denotes the detrended log-price and $\Delta X_t = X_{t+1} - X_t$ its first differences, referred to as log-returns. 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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

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) [9] 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 [10] 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 [11] 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-maximization (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 the log-returns $\Delta X_t = X_{t+1} - X_t$). 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.

Deep learning methods have recently been applied to energy price forecasting and modelling [12–14], 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 [15]. Distribution-free estimation of α from a single time series, without any assumption on the innovation distribution, 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

The key insight motivating our approach is that α determines the autocorrelation structure of $\{X_t\}$, which is invariant to the marginal distribution of ε_t. Specifically, the lag-k autocorrelation of the stationary AR(1) process satisfies $\rho_k = (1-\alpha)^k$, depending on α alone and invariant to both the scale and the shape of the innovation distribution. Any estimator that reads α exclusively from this autocorrelation structure is therefore distribution-free by construction.

The classical Yule-Walker estimator $\hat{\alpha}_{\text{YW}} = 1 - \hat{\rho}_1$ exploits precisely this property: it recovers α from the lag-1 sample autocorrelation alone, without any distributional assumption, and is consistent under any ergodic stationary innovation distribution with finite variance. Its limitation is that it discards the information carried by higher lags $\rho_2, \rho_3, \ldots$. For slow mean reversion ($\alpha \to 0$), all autocorrelations $\rho_k = (1-\alpha)^k$ remain large and the decay is shallow; the lag-1 autocorrelation alone,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

close to 1, provides a weak and imprecise signal for α. An estimator that aggregates autocorrelation information across all lags would gain efficiency precisely in this regime.

We implement this generalisation as a Temporal Convolutional Network (TCN) [16]: 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. Trained on 2×10^5 synthetic AR(1) series with Sinh-ArcSinh (SAS) distributed innovations [17,18], the TCN learns to read the temporal dependence structure, not the marginal distribution of the observations. The SAS family is parameterised by an asymmetry parameter ν and a tail-weight parameter τ; special cases include the Gaussian ($\nu = 0$, $\tau = 1$), heavy-tailed distributions ($\tau < 1$), and skewed distributions ($\nu \neq 0$). By sampling $\nu \in [-0.5, +0.5]$ and $\kappa_\varepsilon \in [3, 17]$ during training, the synthetic dataset covers innovation profiles ranging from near-Gaussian to strongly heavy-tailed (κ_ε up to ≈ 37 at $|\nu| = 0.5$) and skewed ($\gamma_\varepsilon \in [-4.57, +4.57]$), spanning the full spectrum of tail behaviours and asymmetry profiles encountered in energy commodity markets.

The choice of SAS as the training distribution does not constitute a distributional assumption on the true innovation process: SAS is a training vehicle chosen for its practical convenience—the monotone mapping $\tau \mapsto \kappa_\varepsilon(\tau)$ enables κ_ε-uniform sampling via direct numerical inversion without additional reparametrisation (Section 2.2). The training range is deliberately broad: by presenting the network with a wide diversity of innovation shapes, it ensures that the autocorrelation structure $\rho_k = (1-\alpha)^k$ 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. Any parametric family rich enough to span this range would achieve the same effect. More broadly, the train-on-synthetics approach is an instance of simulation-based inference [19–23], a paradigm in which neural networks perform parameter estimation from simulated data without requiring explicit likelihood evaluation. The distribution-free claim therefore applies *at inference*: the trained TCN imposes no distributional assumption on new series and targets the true α, not a model-conditioned surrogate. Empirical support 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.

This has a direct consequence for out-of-distribution generalisation. Since the network has learned to read $\rho_k = (1-\alpha)^k$, and Yule-Walker invariance guarantees that this signal is identical for any innovation distribution at the same α, the TCN encounters the same fundamental structure at test time regardless of which distribution generates the innovations—whether within or beyond the training range. Out-of-distribution generalisation is therefore not a fortuitous empirical finding but a structural prediction that follows necessarily from what the network was forced to learn during training.

This structural guarantee is further reinforced by a complementary effect. Under heavier-tailed innovations (larger κ_ε), individual realisations X_t deviate more widely from zero, producing more pronounced returns to the mean. These larger excursions amplify the autocorrelation *signal strength* at finite sample size T: the contrast between excursion and reversion is sharper and detectable over a wider range of lags, making the mean-reversion dynamics easier for the estimator to resolve. For a fixed α, increasing κ_ε raises the unconditional kurtosis of $\{X_t\}$ without altering its autocorrelation sequence, yet the larger deviations that accompany heavier tails reduce the effective noise relative to the autocorrelation signal. The TCN therefore does not merely survive high-kurtosis inputs: its accuracy tends to *improve* as κ_ε increases, both within the training range and beyond it—a result consistent with the signal-amplification interpretation proposed above, and one that provides empirical support for the hypothesis that the network has learned to read autocorrelation structure rather than distributional features.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

The estimated $\hat{\alpha}$ enables a natural two-stage pipeline: clean residuals $\hat{\varepsilon}_t = X_{t+1} - (1 - \hat{\alpha})\, X_t$ 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.

1.4. 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, providing a demanding and diverse validation environment. Log-return kurtosis ranges from near-Gaussian PUN ($\kappa \approx 5.4$) to strongly heavy-tailed Henry Hub ($\kappa \approx 44.8$), and skewness from -0.79 (Henry Hub, US gas) to $+0.45$ (PSV, Italian gas), confirming that Gaussian assumptions are inadequate on both the tail-weight and asymmetry dimensions.

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 extreme log-return kurtosis (≈ 44.8, far beyond the training range), provides a demanding real-world instance of this out-of-distribution generalisation.

1.5. Contributions

The main contributions of this paper are:

1. **Distribution-free neural estimator.** We introduce the first neural estimator of mean-reversion speed that makes no assumption about the marginal distribution of price innovations, relying instead on the autocorrelation structure that α imprints on the observed series.
2. **Two-stage estimation framework.** The proposed approach naturally decomposes the characterisation of AR(1) dynamics into two independent stages: Stage 1 estimates α model-free via the TCN; Stage 2 characterises the innovation distribution from the clean residuals $\hat{\varepsilon}_t$ by any chosen method. This decoupling—impossible in classical likelihood-based estimators, which jointly optimise α and distributional parameters—eliminates the model dependence that is the central limitation of parametric approaches.
3. **Universal train-once estimator.** Once trained on synthetic data, the TCN requires no re-fitting when applied to new series: it functions as a universal mean-reversion instrument that maps any AR(1)-like time series to $\hat{\alpha}$ without dependence on the specific market or time period at hand. Classical parametric estimators must solve a fresh numerical optimisation for every application; the TCN does not.
4. **SAS-based synthetic training data.** We demonstrate how the Sinh-ArcSinh distribution provides a parsimonious and flexible training environment that covers the full range of fat-tail and skewness profiles observed in energy markets, enabling robust out-of-distribution generalisation.
5. **Benchmark evaluation on synthetic data.** We conduct a comprehensive comparison of the TCN against GARCH(1,1), jump-diffusion, and regime-switching estimators on synthetic series with known α, providing a controlled assessment of accuracy and model dependence.
6. **Evidence of model-conditioned estimation.** Using the same real data sets, we show that classical parametric estimators produce systematically different $\hat{\alpha}$ values, confirming that each estimate is

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

conditioned on the respective distributional model and that their disagreement is a structural artefact, not sampling variability.

7. **Application to four energy markets.** We apply the TCN estimator to daily prices for Italian natural gas (PSV price) and electricity (PUN price), US Henry Hub (gas), and US PJM West Hub (power), delivering robust, model-free $\hat{\alpha}$ estimates across markets with markedly different kurtosis profiles ($\kappa \in [5.4, 44.8]$). The Henry Hub case, with its extreme kurtosis, further validates the distribution-free robustness results.

1.6. 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 up to $\kappa = 45$. 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

$$X_{t+1} = (1 - \alpha)\, X_t + \varepsilon_t, \tag{3}$$

with ε_t i.i.d. with finite moments. The parameter of interest is the mean-reversion speed α.

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

$$\tilde{X}_t = \frac{X_t - \bar{X}}{s_X}, \qquad \bar{X} = \frac{1}{T}\sum_{t=1}^{T} X_t, \quad s_X^2 = \frac{1}{T-1}\sum_{t=1}^{T}(X_t - \bar{X})^2. \tag{4}$$

Under the AR(1) model (3), 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:

$$\hat{\alpha} = f_{\hat{\theta}}(\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_T), \tag{5}$$

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 wide and representative variety of innovation distributions during training. We use the Sinh-ArcSinh (SAS) distribution introduced by Jones and Pewsey [17,18], which provides a flexible family capable of modelling the full range of tail behaviours and asymmetry profiles observed in energy markets.

2.2.1. Definition

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

$$\varepsilon = \sinh\left(\frac{\operatorname{arcsinh}(Z) + \nu}{\tau}\right), \tag{6}$$

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

where $\nu \in \mathbb{R}$ controls asymmetry and $\tau > 0$ controls tail weight. The transformation (6) 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 (6) 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 [17], which induces a non-zero stationary mean $\mathbb{E}[X_t] = \mathbb{E}[\varepsilon_t]/\alpha$ in the AR(1) process. The autocorrelation structure $\rho_k = (1-\alpha)^k$ is independent of $\mathbb{E}[\varepsilon_t]$: it follows from the Yule-Walker equations, since $\mathrm{Cov}(X_t, X_{t+k}) = (1-\alpha)^k \mathrm{Var}(X_t)$ regardless of the innovation mean. The non-zero mean $\mathbb{E}[X_t]$ is removed by the series standardisation (4), which preserves ρ_k identically as a linear transformation.

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 [17]:

$$\kappa_\varepsilon(\tau) = \frac{M(4/\tau) - 4M(2/\tau) + 3}{2\big[M(2/\tau) - 1\big]^2}. \tag{7}$$

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 (7) yields τ exactly, without additional reparametrisation.

The training ranges are $\kappa_\varepsilon \sim \mathrm{Uniform}(3, 17)$, spanning from the Gaussian baseline ($\kappa_\varepsilon = 3$, $\tau = 1$) to strongly leptokurtic innovations ($\kappa_\varepsilon = 17$, $\tau \approx 0.40$), and $\nu \sim \mathrm{Uniform}(-0.5, +0.5)$. Since Equation (7) 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, providing comprehensive coverage of the profiles encountered in energy markets.

Throughout this paper κ_ε denotes the Pearson kurtosis of the innovation ε_t, distinct from the log-return kurtosis $\kappa(\Delta X_t)$; the two are related through Equation (A6) in Appendix A. We emphasise that SAS is used as a *training vehicle*, not as an assumption on the true innovation distribution; the rationale for this distinction and its implications for the distribution-free claim are developed in Section 1.3.

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: $\alpha \sim \mathrm{Uniform}(0.01, 1.00)$, $\nu \sim \mathrm{Uniform}(-0.50, +0.50)$, $\kappa_\varepsilon \sim \mathrm{Uniform}(3, 17)$. Given κ_ε, the tail parameter τ is obtained by numerical inversion of Equation (7); 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 $T_i \sim \mathrm{Uniform}(1200, 1826)$ 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 $\mathrm{SAS}(\nu, \tau)$ samples using the representation (6), where $T_{\text{burn}} = 200$ is a burn-in buffer discarded at the next step.
4. **AR(1) recursion.** Apply the recursive filter (3) starting from $X_0 = 0$; discard the first T_{burn} steps to ensure approximate stationarity.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

5. **Standardisation.** Apply Equation (4) 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.

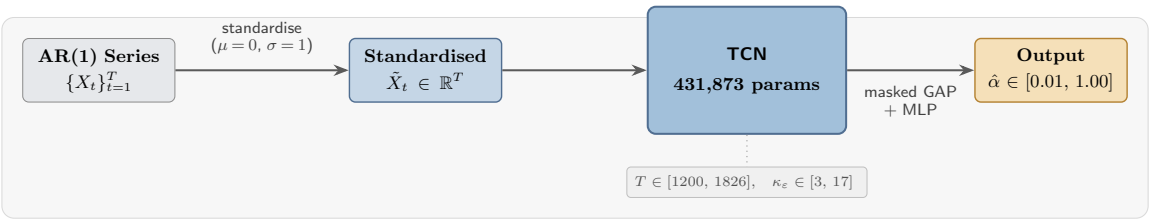

Figure 1. Estimation pipeline. A stationary AR(1) series $\{X_t\}_{t=1}^{T}$ is standardised to yield $\tilde{X}_t$, which is fed to the TCN; the output is the distribution-free mean-reversion speed estimate $\hat{\alpha} \in [0.01, 1.00]$. During training, $\{X_t\}$ is generated synthetically using 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) [16,24] 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:

$$\text{RF} = 1 + 2(w-1)\left(2^L - 1\right), \tag{8}$$

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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

sample independently over its feature dimension, with no cross-sample running statistics—is applied before a two-layer multi-layer perceptron (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 (8) 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 (8) 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.

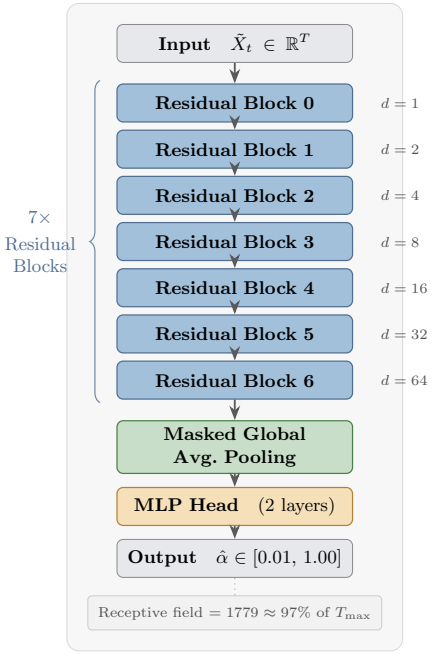

Figure 2. TCN architecture. The standardised input series passes through seven residual blocks with exponentially growing dilation factors $d = 2^0, 2^1, \ldots, 2^6$. Masked Global Average Pooling aggregates the temporal representations across all valid (non-padded) timesteps, and a two-layer MLP head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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

2.5. Training Procedure

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

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{n=1}^{N} \left| \hat{\alpha}_n(\theta) - \alpha_n \right|, \tag{9}$$

where $\hat{\alpha}_n(\theta) = f_\theta(\tilde{X}_n)$ is the TCN prediction for series n (cf. Equation (5)), 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 seconds, for a total training time of approximately 75 minutes 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 [25], 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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

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** (bold) is marginally below the validation MAE of 0.01017, confirming no overfitting.

Bin (α)	N	MAE
$[0.01,\ 0.11)$	2,095	0.00484
$[0.11,\ 0.21)$	1,934	0.00718
$[0.21,\ 0.31)$	2,003	0.00877
$[0.31,\ 0.41)$	1,976	0.00995
$[0.41,\ 0.51)$	2,020	0.01061
$[0.51,\ 0.60)$	2,033	0.01146
$[0.60,\ 0.70)$	2,014	0.01134
$[0.70,\ 0.80)$	1,995	0.01226
$[0.80,\ 0.90)$	1,993	0.01285
$[0.90,\ 1.00)$	1,937	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 upper boundary constraint.

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 physically interpretable: heavier-tailed innovations produce more pronounced spikes after standardisation, making the AR(1) autocorrelation structure more visible to the TCN; the improvement in κ_ε accuracy thus reflects increasing signal-to-noise rather than any 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, reflecting the stronger autocorrelation signal in heavy-tailed innovations.

Bin (κ_ε)	N	MAE
$[3,\ 5)$	2,815	0.01474
$[5,\ 7)$	2,853	0.01223
$[7,\ 9)$	2,817	0.01033
$[9,\ 11)$	2,879	0.00929
$[11,\ 13)$	2,868	0.00874
$[13,\ 15)$	2,873	0.00796
$[15,\ 17)$	2,895	0.00765

4. Benchmark Comparison

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

The AR(1)-GARCH(1,1) model [9] specifies

$$X_{t+1} = (1 - \alpha)\, X_t + \varepsilon_t, \tag{10}$$

$$\varepsilon_t = \sigma_t\, z_t, \quad z_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1), \tag{11}$$

$$\sigma_t^2 = \omega + a\, \varepsilon_{t-1}^2 + b\, \sigma_{t-1}^2, \tag{12}$$

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 (11).

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

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 [10,26,27]:

$$X_{t+1} = (1-\alpha)\, X_t + \varepsilon_t, \tag{13}$$

$$\varepsilon_t = \sigma\, z_t + \sum_{j=1}^{N_t} \eta_{t,j}, \quad z_t \sim \mathcal{N}(0,1), \quad N_t \sim \text{Poisson}(\lambda), \quad \eta_{t,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu_J, \sigma_J^2), \tag{14}$$

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 (14) 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 $s_t \in \{0, 1\}$ [11]:

$$X_{t+1} = (1-\alpha)\, X_t + \sigma_{s_t}\, z_t, \quad z_t \sim \mathcal{N}(0,1), \tag{15}$$

$$s_t \sim \text{Markov}(P), \tag{16}$$

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. Comparison Results

Panel (b) of Figure 3 compares the TCN with the three classical estimators.

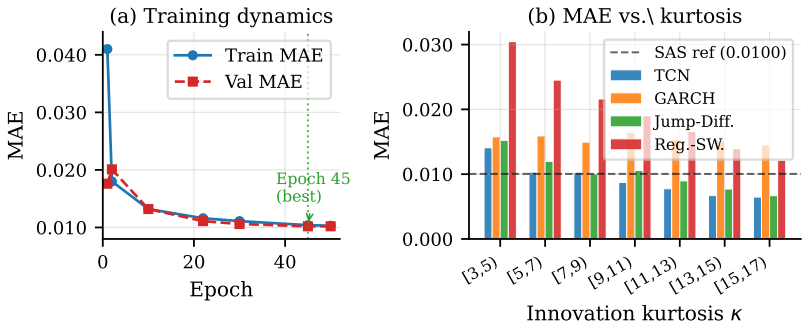

Figure 3. (a) Training and validation MAE over 50 epochs (κ_ε-uniform training). The best checkpoint is selected at epoch 45 (val MAE = 0.01017). (b) TCN MAE versus κ_ε compared to three classical estimators ($N = 200$ series per bin). The dashed line marks the SAS in-distribution reference MAE (0.01004).

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

Table 5 reports the benchmark comparison stratified by κ_ε (bin labels computed at $\nu = 0$, see Section 2.2; $N = 1{,}400$ 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 four estimators are applied to the same set of synthetic series: the TCN produces $\hat{\alpha}$ in a single forward pass, while each classical estimator performs an independent maximum-likelihood or EM optimisation on the same series. The TCN achieves an overall MAE of **0.00918**, outperforming GARCH(1,1) (0.01545, 68.3% worse), Jump-Diffusion (0.01017, 10.8% worse), and Regime-Switching (0.01975, 115.1% worse).

Table 5. MAE comparison of TCN vs. classical estimators stratified by κ_ε (bin labels computed at $\nu = 0$, see Section 2.2). $N = 200$ series per bin ($\alpha \sim \text{Uniform}(0.01, 1.00)$, $\nu \sim \text{Uniform}(-0.50, +0.50)$, $T = 1{,}826$). Bold: best result per row.

Bin (κ_ε)	TCN	GARCH(1,1)	Jump-Diff.	Regime-SW.
[3, 5)	0.014 07	0.015 75	0.015 22	0.030 45
[5, 7)	0.010 31	0.015 90	0.011 98	0.024 52
[7, 9)	0.010 29	0.014 95	0.010 05	0.021 60
[9, 11)	0.008 71	0.016 43	0.010 57	0.019 01
[11, 13)	0.007 75	0.015 36	0.008 96	0.016 58
[13, 15)	0.006 69	0.015 23	0.007 71	0.013 93
[15, 17)	0.006 46	0.014 50	0.006 68	0.012 14
Overall	0.009 18	0.015 45	0.010 17	0.019 75

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 network learns to leverage the autocorrelation signal that heavy tails amplify, whereas classical estimators' likelihoods are increasingly misspecified.

The regime-switching specification adopted here—two states, Gaussian within-regime innovations, single mean-reversion coefficient shared across regimes—is adopted as a parsimonious baseline for the comparison [11]. More elaborate specifications (additional regimes, regime-specific AR coefficients, non-Gaussian within-regime innovations) could in principle reduce estimation error, but at the cost of substantially increased calibration complexity: the EM algorithm is sensitive to initialisation, the number of regimes becomes a model-selection problem requiring information criteria that themselves embed distributional assumptions, and regime-specific AR coefficients require an aggregation convention to yield a single $\hat{\alpha}$, introducing further model dependence. The TCN requires none of these choices.

As a complementary evaluation, Table 6 reports the overall MAE of the four estimators on 1,000 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 four estimators on α-stratified series ($N = 1{,}000$, 100 per bin, $\alpha \sim \text{Uniform}(0.01, 1.00)$, $\kappa_\varepsilon \sim \text{Uniform}(3, 17)$, $\nu \sim \text{Uniform}(-0.50, +0.50)$, $T = 1{,}826$). Bold: best result.

TCN	GARCH(1,1)	Jump-Diff.	Regime-SW.
0.009 44	0.015 17	0.009 65	0.019 07

Crucially, the three classical estimators produce *different* $\hat{\alpha}$ values from the same series. These differences are not sampling noise: they are the empirical manifestation of the model-conditioned nature of $\hat{\alpha}$. On synthetic data with known ground truth, they are quantifiable as differential biases. On

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

real data—where the true α is unknown—the three estimators will continue to disagree without any principled way to adjudicate between them, since each $\hat{\alpha}$ answers a different question conditioned on a different distributional model. 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 ($N = 1{,}000$ AR(1) series each, $\alpha \sim \text{Uniform}(0.01, 1.00)$, $T = 1{,}826$, 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 (5,000 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 MAE on out-of-distribution innovation families (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. The parameter of each family is obtained by numerical inversion of the kurtosis formula.

Family	Parameter	κ_ε	In training	MAE
Student-t	$d = 6.000$	6	yes	0.014 51
Student-t	$d = 5.000$	9	yes	0.013 97
Student-t	$d = 4.500$	15	yes	0.013 35
Student-t	$d = 4.353$	20	OOD	0.013 44
Student-t	$d = 4.222$	30	OOD	0.013 30
Student-t	$d = 4.143$	45	OOD	0.012 82
GED	$\beta = 1.000$	6	yes	0.012 31
GED	$\beta = 0.778$	9	yes	0.010 97
GED	$\beta = 0.610$	15	yes	0.009 83
GED	$\beta = 0.544$	20	OOD	0.008 44
GED	$\beta = 0.472$	30	OOD	0.008 10
GED	$\beta = 0.417$	45	OOD	0.007 87
Gaussian mix	$\sigma_2 = 2.449$	6	yes	0.013 74
Gaussian mix	$\sigma_2 = 3.149$	9	yes	0.013 19
Gaussian mix	$\sigma_2 = 4.583$	15	yes	0.010 45
Gaussian mix	$\sigma_2 = 6.279$	20	OOD	0.008 76
Gaussian mix	$\sigma_2 = 15.997$	28	OOD	0.008 03

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

The theoretical framework of Section 1.3 leads to three precise predictions. 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; (iii) accuracy should tend to *improve* with increasing κ_ε at fixed α, due to the signal-amplification effect of heavy tails identified above.

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 follows from the same mechanism observed in Table 4: heavier-tailed innovations, regardless of the distribution family, create more pronounced autocorrelation structures that are easier for the TCN to exploit. 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 signal-amplification effect ensures that the rare large excursions produced by heavy-tailed innovations make the mean-reversion pattern more, not less, detectable—explaining why performance continues to improve well beyond the training boundary.

The Student-t family produces consistently higher MAE than GED and Gaussian mixture at the same κ_ε value. This can be attributed to the structural difference between power-law tails (Student-t, which produce occasional extreme values with algebraically decaying frequency) and exponential or discrete-mixture tails (GED, Gaussian mix, which generate tail events more predictably scaled to σ). The latter two families produce autocorrelation patterns more structurally similar to the SAS training distribution, leading to better cross-family transfer.

A natural question is whether extending the training range beyond $\kappa_\varepsilon = 17$ would further improve performance at extreme tail weights. We argue that the current choice is preferable on three grounds. First, demonstrating stable OOD generalisation to $\kappa_\varepsilon = 45$ is scientifically stronger than an in-distribution claim: training on $\kappa_\varepsilon \in [3, 45]$ would absorb this result into the baseline, eliminating its evidential value as a robustness test. Second, κ_ε-uniform sampling over $[3, 45]$ would allocate the majority of training capacity to extreme configurations ($\kappa_\varepsilon > 17$), where series increasingly resemble jump processes rather than stationary AR(1) realisations, potentially degrading performance in the practically relevant range. Third, the extreme log-return kurtosis observed in some energy markets (discussed in Section 6) is largely driven by isolated spike events rather than by the stationary innovation distribution, so the structural innovation kurtosis κ_ε of those markets falls well within the training range $[3, 17]$. Fourth, and most fundamentally, the signal-amplification mechanism identified throughout this paper implies that the TCN's performance naturally *improves* beyond the training boundary: heavier-tailed innovations produce more pronounced autocorrelation patterns, making mean reversion easier to detect regardless of the specific distribution family (Tables 4 and 7). Extending the training range is therefore unnecessary—the network already benefits from the amplified autocorrelation signal at high κ_ε without additional training exposure. The training range is therefore both

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

empirically motivated and scientifically conservative, and the present robustness analysis provides independent evidence for extrapolation beyond it.

6. Real Data Application

Preprocessing. 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 [28] (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.

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 8 summarises the first four moments of the first differences ΔX_t; Table 9 reports the mean-reversion speed estimates $\hat{\alpha}$ and the implied half-lives for all estimators.

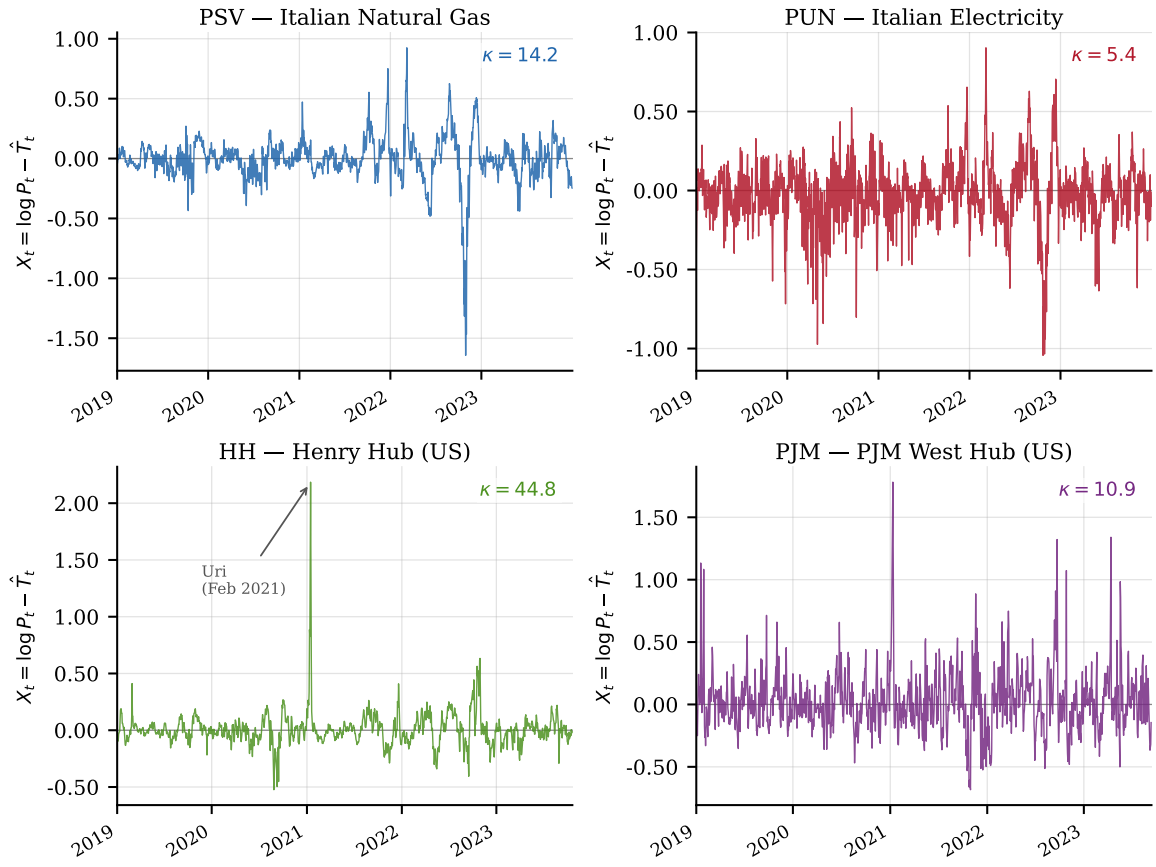

Figure 4. Detrended log-price series $X_t = \log P_t - \hat{T}_t$ for all four markets (LOESS bandwidth $h = 0.1$, period 2019–2023). Top row: Italian markets (PSV natural gas price, $\kappa = 14.2$; PUN electricity price, $\kappa = 5.4$). Bottom row: US markets (Henry Hub gas, $\kappa = 44.8$; PJM West 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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

Table 8. 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 demanding real-data validation across all regimes covered by the synthetic training protocol. The HH raw kurtosis is dominated by the Winter Storm Uri spike sequence (February 2021); excluding those observations yields a structural $\kappa \approx 9.3$, well within the training range (see the HH discussion below for full interpretation).

Table 9. 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.

Several features of Table 9 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 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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

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. Removing the Winter Storm Uri observations leaves a structural log-return kurtosis of approximately $\kappa \approx 9.3$, implying a structural innovation kurtosis κ_ε well within the training range $[3, 17]$; the full-sample raw $\kappa = 44.8$ reflects a jump event, not a change in the underlying mean-reversion dynamics. It should be noted that $\kappa = 44.8$ lies outside the TCN training range, so the estimate $\hat{\alpha} = 0.200$ carries additional uncertainty; the extended high-kurtosis robustness analysis (Section 5) provides empirical support for the stability of TCN performance beyond the training boundary. Importantly, the synthetic test confirms stability but not accuracy on HH, since the true α is unknown; the stronger evidential basis for the direction of the discrepancy is the classical estimators' systematic downward bias—identified on synthetic data where the ground truth is available—not the OOD robustness result alone.

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 for energy commodity prices. The estimator is a Temporal Convolutional Network trained on 2×10^5 synthetic AR(1) series per epoch, with Sinh-ArcSinh innovations whose kurtosis is sampled uniformly from $\kappa_\varepsilon \sim \text{Uniform}(3, 17)$ and series length randomised over $[1200, 1826]$ steps. The TCN architecture ($L = 7$ residual blocks, kernel size $w = 8$, $n_f = 64$ filters, receptive field of 97.4% of the input length, 431,873 parameters) is trained with MAE loss using Adam and a cosine learning-rate schedule. On 20,000 held-out test series the model achieves MAE $= 0.01004$, with a training-to-validation gap of only 0.00013.

The test-set analysis reveals two complementary patterns. Across α bins, the TCN achieves its lowest MAE of 0.00484 for $\alpha \in [0.01, 0.11)$ (slow mean reversion, long half-life, strong autocorrelation signal) and its highest MAE of 0.01285 for $\alpha \in [0.80, 0.90)$; the monotone degradation as $\alpha \to 1$ is expected from the progressive weakening of the autocorrelation signal. Across kurtosis bins, a complementary trend emerges: TCN accuracy *improves* with tail weight, from MAE $= 0.01474$ at $\kappa_\varepsilon \in [3, 5)$ to MAE $= 0.00765$ at $\kappa_\varepsilon \in [15, 17)$, reflecting the stronger autocorrelation signal that heavy-tailed innovations impart on the process.

The kurtosis-stratified benchmark comparison on 1,400 synthetic series reveals the mechanism underlying the TCN's advantage. For near-Gaussian innovations ($\kappa_\varepsilon \in [3, 5)$), classical estimators are competitive and the TCN already achieves the lowest MAE (0.01407), though the margin over Jump-Diffusion (0.01522) and GARCH (0.01575) is modest; this is expected, as parametric models designed under Gaussian-family assumptions perform well in the near-Gaussian regime. As tail weight increases, the TCN's advantage widens monotonically: at $\kappa_\varepsilon \in [15, 17)$ the TCN (MAE $= 0.00646$) achieves 2.2× lower error than GARCH (0.01450) and 1.9× lower than Regime-Switching (0.01214).

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

GARCH's MAE remains nearly constant across all kurtosis bins (≈ 0.015), confirming its insensitivity to innovation tail shape. The regime-switching model is evaluated in its standard practical specification (two Gaussian regimes, single AR coefficient); more complex specifications—additional regimes, non-Gaussian innovations, regime-specific AR—would increase calibration burden without resolving the fundamental model-dependence problem, since the estimated $\hat{\alpha}$ would remain contingent on the chosen regime structure. The robustness analysis on three out-of-distribution distribution families (Student-t, GED, Gaussian mixture) further validates the distribution-free claim: MAE values are stable across families and in all cases comparable to or below the SAS reference, with no performance cliff at the $\kappa_\varepsilon = 17$ training boundary. Most importantly, the three classical estimators produce *different* $\hat{\alpha}$ values from the same series, confirming the fundamental model-dependence identified in Section 1.2. Applied to four energy markets spanning Italian and US gas and electricity prices, the TCN yields physically plausible estimates: $\hat{\alpha}_{\text{PSV}} = 0.0975$ (half-life 7.1 days), $\hat{\alpha}_{\text{PUN}} = 0.286$ (half-life 2.4 days), $\hat{\alpha}_{\text{PJM}} = 0.354$ (half-life 2.0 days), and $\hat{\alpha}_{\text{HH}} = 0.200$ (half-life 3.5 days). The Henry Hub case is particularly instructive: the three classical estimators produce $\hat{\alpha} \approx 0.11$ and agree with each other (spread $= 0.015$) but all diverge from the TCN, consistent with a systematic downward bias caused by the extreme Winter Storm Uri events being absorbed by the non-AR model components rather than reflected in the mean-reversion coefficient. Excluding the Winter Storm Uri spike sequence yields a structural log-return kurtosis of approximately 9.3, within the TCN training range $[3, 17]$, confirming that the bias reflects the treatment of isolated spike events and not a fundamental incompatibility with the training distribution. Across all four markets, the three classical estimators produce $\hat{\alpha}$ values that differ by 0.041 for PSV, 0.053 for PUN, 0.015 for HH, and 0.040 for PJM. The PSV spread (0.041) corresponds to a factor-of-two half-life discrepancy (8.3 vs 16.5 days) driven by heavy-tailed log-returns ($\kappa = 14.2$) being absorbed differently by each classical likelihood. The PUN and PJM spreads are larger in absolute α units but small in relative terms—19% and 13% of the respective mean estimates—consistent with the limited model-dependence expected when innovations are near-Gaussian or only moderately heavy-tailed ($\kappa \in \{5.4, 10.9\}$). The HH spread of 0.015 is the smallest in absolute terms but the most consequential: all three classical estimators agree in underestimating α relative to the TCN, revealing a shared systematic bias that translates into a factor-of-two underestimate of the mean-reversion speed.

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. By contrast, classical parametric estimators must solve a numerical maximisation problem (MLE or EM) for every new series, with computation times ranging from seconds (GARCH) to minutes (Regime-Switching), and with no guarantee of convergence in the presence of short series or heavy-tailed observations. In an operational setting where α must be estimated for hundreds of instruments, contract tenors, or bootstrap scenarios, this difference in computational cost is material. A single model-free estimate of mean-reversion speed can moreover be used consistently across risk models, derivative pricing engines, and portfolio optimisers without introducing distributional assumptions that would compromise internal consistency. Risk managers using different classical estimators would obtain half-lives differing by a factor of two for PSV—a discrepancy large enough to materially affect Value-at-Risk calculations, swing-option prices, and portfolio rebalancing frequencies. The TCN provides a distribution-free benchmark that is not subject to this ambiguity. More broadly, 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, functioning as a reusable analytical tool rather than a model to be estimated anew for each application. The distribution-free $\hat{\alpha}$ also enables the two-stage pipeline introduced in Section 6: once α is fixed by Stage 1, the residuals $\hat{\varepsilon}_t = X_{t+1} - (1 - \hat{\alpha})X_t$ are available for independent distributional analysis in Stage 2, free from the coupling to α that contaminates classical likelihood-based approaches. This decoupling—structurally impossible in GARCH, Jump-Diffusion,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

or Regime-Switching, which jointly optimise α and distributional parameters—is a direct consequence of estimating α from autocorrelation structure alone. As neural architectures continue to advance in expressiveness and efficiency, this paradigm opens a path toward increasingly powerful universal estimators of stochastic dynamics: networks trained on progressively richer synthetic environments will deliver more accurate and robust mean-reversion estimates across a wider range of markets, instruments, and distributional regimes.

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 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 [29,30]. Second, the assumption of a time-invariant α could be relaxed by adding a non-stationarity detection module or by recasting the problem as a regime-conditional estimation task. Third, the principle underlying the proposed approach—training a neural estimator on synthetic data with distributional variation spanning the empirically relevant range—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: Conceptualization, 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.; visualization, E.M.; supervision, C.M.; 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). 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).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Moments of AR(1) Log-Returns

Let $\{X_t\}$ be the stationary AR(1) process in Equation (1) with iid 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.

Cumulant Formula

Since X_t and ε_t are independent, cumulant additivity applies throughout [31]: 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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

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

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

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 log-returns are zero-mean regardless of $\mathbb{E}[\varepsilon_t]$.

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 log-returns coincide with innovations in the absence of mean reversion. For $\alpha > 0$, the kurtosis scaling factor in (A6) is strictly less than one, so $\kappa(\Delta X_t) < \kappa_\varepsilon$: log-return 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.

Implied Innovation Moments for the Four Markets

Table A1 reports, for each market, the log-return moments from Table 8 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 8) and implied innovation moments obtained by inverting Equations (A4)–(A6) using the TCN estimates $\hat{\alpha}$. The implied innovation kurtosis κ_ε confirms that all markets except Henry Hub fall within the training range $[3,17]$; for HH the implied $\kappa_\varepsilon \approx 54$ reflects the Winter Storm Uri spike and is consistent with the OOD robustness analysis of Section 5.

	Log-return moments ΔX_t				**Implied innovation moments ε_t**		
Market	$\hat{\alpha}$	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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

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. *International Journal of Forecasting* **2014**, *30*, 1030–1081.
3. De Sanctis, A.; Mari, C. Modeling spikes in electricity markets using excitable dynamics. *Physica A: Statistical Mechanics and its Applications* **2007**, *384*, 547–560.
4. Leung, T.; Lu, K.W. Monte Carlo simulation for trading under a Lévy-driven mean-reverting framework. *Applied Mathematical Finance* **2024**, *30*, 207–230.
5. Schwartz, E.S. The stochastic behavior of commodity prices: Implications for valuation and hedging. *Journal of Finance* **1997**, *52*, 923–973.
6. Lucheroni, C.; Mari, C. Internal hedging of intermittent renewable power generation and optimal portfolio selection. *Annals of Operations Research* **2019**. https://doi.org/10.1007/s10479-019-03221-2.
7. Mari, C. Stochastic NPV based vs stochastic LCOE based power portfolio selection under uncertainty. *Energies* **2020**, *13*, 3677.
8. Geman, H. *Commodities and Commodity Derivatives: Modelling and Pricing for Agriculturals, Metals and Energy*; Wiley: Chichester, UK, 2005.
9. Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics* **1986**, *31*, 307–327.
10. Merton, R.C. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics* **1976**, *3*, 125–144.
11. Hamilton, J.D. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica* **1989**, *57*, 357–384.
12. 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. *Applied Energy* **2021**, *293*, 116983.
13. Marcjasz, G.; Narajewski, M.; Weron, R.; Ziel, F. Distributional neural networks for electricity price forecasting. *Energy Economics* **2023**, *125*, 106843.
14. Mari, C.; Mari, E. Deep learning based regime-switching models of energy commodity prices. *Energy Systems* **2023**. https://doi.org/10.1007/s12667-022-00515-6
15. Fein-Ashley, J. A comparison of traditional and deep learning methods for parameter estimation of the Ornstein-Uhlenbeck process. *arXiv preprint arXiv:2404.11526* **2024**.
16. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* **2018**.
17. Jones, M.C.; Pewsey, A. Sinh-arcsinh distributions. *Biometrika* **2009**, *96*, 761–780.
18. Jones, M.C.; Pewsey, A. The eschewed sinh-arcsinh *t* distribution. *Statistics and Probability Letters* **2025**, *228*, 110560.
19. Cranmer, K.; Brehmer, J.; Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences* **2020**, *117*, 30055–30062.
20. Lenzi, A.; Bessac, J.; Rudi, J.; Stein, M.L. Neural networks for parameter estimation in intractable models. *Computational Statistics & Data Analysis* **2023**, *185*, 107762.
21. Sainsbury-Dale, M.; Zammit-Mangion, A.; Huser, R. Likelihood-free parameter estimation with neural Bayes estimators. *The American Statistician* **2024**, *78*, 1–14.
22. Zammit-Mangion, A.; Sainsbury-Dale, M.; Huser, R. Neural methods for amortized inference. *Annual Review of Statistics and Its Application* **2025**, *12*, 311–335.
23. Gloeckler, M.; Toyota, S.; Fukumizu, K.; Macke, J.H. Compositional simulation-based inference for time series. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Singapore, 2025. arXiv:2411.02728.
24. Lara-Benitez, P.; Carranza-Garcia, M.; Luna-Romera, J.M.; Riquelme, J.C. Temporal convolutional networks applied to energy-related time series forecasting. *Applied Sciences* **2020**, *10*, 2322.
25. 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, 2015; 1026–1034.
26. Mari, C.; Mari, E. Gaussian clustering and jump-diffusion models of electricity prices: a deep-learning analysis. *Decisions in Economics and Finance* **2021**. https://doi.org/10.1007/s10203-021-00332-z.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 March 2026 doi:10.20944/preprints202603.0996.v1

27. Mari, C.; Baldassari, C. Ensemble methods for jump-diffusion models of power prices. *Energies* **2021**, *14*, 4404.
28. Cleveland, W.S. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association* **1979**, *74*, 829–836.
29. Benth, F.E.; Koekebakker, S. Pricing of forwards and other derivatives in cointegrated commodity markets. *Energy Economics* **2015**, *52*, 104–117.
30. Farkas, W.; Gourier, E.; Huitema, R.; Necula, C. A two-factor cointegrated commodity price model with an application to spread option pricing. *Journal of Banking & Finance* **2017**, *77*, 249–268.
31. Brockwell, P.J.; Davis, R.A. *Time Series: Theory and Methods*, 2nd ed.; Springer-Verlag: New York, NY, USA, 1991.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.