

Politecnico di Torino

Politecnico di Torino

Computer Engineering - Artificial Intelligence and Data Analytics

A.y. 2025/2026

Graduation session: October 2026

Adaptive and Causally-Consistent Conformal Calibration for Non-Stationary Multi-Step Time Series Forecasting

Supervisors:

Michela Meo

Greta Vallero

Daniela Renga
(Politecnico di Torino)

Mario Bonansone

Matteo Di Cunzolo
(Modelway S.R.L.)

Candidate:

Federico Carollo

"We may regard the present state of the universe as the effect of its past and the cause of its future."

— **Pierre-Simon Laplace**, ***A Philosophical Essay on Probabilities***

Contents

Abstract 4
1. Introduction 6
2. Background and Related Work 10
   2.1. Deep Learning for Time Series Forecasting 10
      2.1.1. Time-series Dense Encoder (TiDE) 11
      2.1.2. iTransformer and Mixed Covariates Extension 12
      2.1.3. Time-Series Mixer (TSMixer) 14
   2.2. Quantile Regression and Calibration Challenges 17
   2.3. Conformal Predictions for Time Series Forecasting 18
      2.3.1. Classical Conformal Prediction 18
      2.3.2. Conformalized Quantile Regression 20
   2.4. Echo State Networks 22
      2.4.1. Echo State Property 23
   2.5. Recent Developments in Sequential Conformal Forecasting 24
3. Proposed Methodology: Adaptive Conformal Calibration for Non-Stationary and Multi-Step Forecasting 26
   3.1. Topological Initialization 27
   3.2. Similarity and Temporal Decay 29
      3.2.1. Similarity Functions 29
      3.2.2. Temporal Decay 29
   3.3. Weighted Calibration Intervals 30
      3.3.1. Adaptive Blending and Stability Guardrails 31
      3.3.2. Conformalized Quantile Regression as a Special Case 33
   3.4. Causal Resolution of Multi-Step Conformal Calibration 34
      3.4.1. Causal Residual Admissibility Framework for Multi-Step Conformal Calibration 36
      3.4.2. Multi-Step Residual Buffering 36
      3.4.3. Theoretical Guarantees for Multi-Step Resolution 37
      3.4.4. Online Inference Cycle 41
   3.5. Post-hoc Calibration for Risk-Sensitive Decision-Making 42
      3.5.1. Decoupling Prediction from Risk Control 42
      3.5.2. Dynamic Shift Update Mechanism 43
      3.5.3. Guaranteeing Quantile Monotonicity and Stability 44
4. Datasets and Evaluation Framework 46
   4.1. Exploratory Data Analysis 46
      4.1.1. The Renewables Distinction 46
   4.2. Datasets 47
      4.2.1. Scaling and Normalization 49
      4.2.2. Time encodings 51
   4.3. Experimental setup 52
      4.3.1. Framework 52
      4.3.2. Metrics 53

4.3.3. Loss function 54
5. Experimental Pipeline 57
5.1. Phase 1: Base Model Optimization (Bayesian Search) 57
5.2. Phase 2: Exploratory Calibration (Grid Search) 60
5.2.1. Conformalized Quantile Regression Baseline 61
5.3. Phase 3: Global Validation 61
5.4. Phase 4: Post-Hoc Risk Controller Evaluation 62
6. Results 64
6.1. Exploratory Calibration and Grid Search Results 64
6.1.1. Hyperparameter Dynamics: Insights 66
6.1.2. The Solar Challenge: The IWS-MIW Paradox 72
6.1.3. Robustness to Concept Drift: The COVID-19 Anomaly 73
6.1.4. Architectural Selection for Global Validation 75
6.1.5. Parameter Robustness and Elite Configurations 75
6.2. Global Validation Results 76
6.3. Post-hoc Calibration Experiments 78
6.3.1. Business-Aware Target Metric 78
6.3.2. Results 78
6.3.3. Analysis of Results 80
7. Conclusions 84
A Appendix: Extended Experimental Results 87
A.1 Experimental Results (Phase 1 and 2) 87
A.1.1 Hyperparameter Selection Trends 87
A.1.2 ResCP parameters 89
A.1.3 Comprehensive Sensitivity Plots (Phase 2) 90
A.1.4 Hyperparameter choice for base models for phase 3 97
A.2 Global Results (Phase 3) 99
A.2.1 Global Results Summary 99
B Appendix : Point-Forecast Benchmark: Deep Models vs. Gradient Boosting Baseline .. 102
B.1 Motivation and Scope 102
B.2 Experimental Setup 103
B.3 Results 104
B.4 Discussion 104
B.4.1 Overall Ranking 104
B.4.2 Scaling granularity strongly modulates TSMixer's performance 107
B.4.3 Inference latency 108
Bibliography 109

Abstract

Uncertainty quantification is critical in time series forecasting, particularly in energy systems, where unreliable predictions can directly impact operational decisions and system stability. While deep learning models trained via quantile regression provide flexible predictive intervals, they often suffer from miscalibration. Conformal Prediction offers a principled framework to enforce valid coverage guarantees. However, standard approaches like CQR assume stationarity, and existing dynamic methods like ResCP, while leveraging Echo State Networks (ESNs) for context, often rely on restricted calibration settings and struggle with the observation lag in multi-step forecasting.

In this work, we propose a significant extension of the Reservoir Conformal Prediction (ResCP) framework. First, we enhance the adaptability to non-stationary dynamics by introducing a versatile pool of similarity-based weighting functions and configurable temporal decay profiles, moving beyond the original fixed formulations. To ensure stability, we integrate an adaptive blending strategy based on Effective Sample Size (ESS) and a structural safety constraint. Second, we address the multi-step challenge by formalizing a theoretically sound causal framework. Through an asynchronous pending queue mechanism, we resolve the observation lag and provide bounds that scale dynamically with horizon-specific uncertainty.

Experimental results on real-world energy datasets demonstrate that our extended ResCP framework significantly improves the reliability and sharpness of prediction intervals compared to standard benchmarks.

Beyond calibration quality, we further propose an online post-hoc risk controller that decouples statistical calibration from business logic, enabling dynamic adjustment of predicted quantiles to satisfy application-specific cost targets in a streaming, non-stationary setting.

Declaration on the Use of Artificial Intelligence

During the preparation of this thesis, Artificial Intelligence (AI) tools were employed for formatting assistance, text refinement, and as an interactive pair-programming assistant during the software development phase. Nevertheless, the conception of the core ideas, the methodological design, the architectural decisions, the execution of the experiments, and the critical analysis of the results are entirely the original work of the author.

Chapter 1
Introduction

"It is the mark of an educated man to look for precision in each class of things just so far as the nature of the subject admits."

— **Aristotle, *Nicomachean Ethics***

Accurate and reliable time series forecasting is critical in real-world applications, particularly in the energy domain, where prediction errors directly impact operational costs, system stability, and decision-making processes. In these settings, point forecasts alone are insufficient, as they fail to capture the inherent uncertainty of the system. Prediction intervals instead provide a more informative representation, enabling risk-aware decisions and improving the robustness of downstream applications such as energy management systems.

Recent advances in deep learning have significantly improved forecasting accuracy, with architectures such as TiDE [1], iTransformer [2], and TSMixer [3] demonstrating strong capabilities in modeling complex temporal and multivariate dependencies. When adapted to quantile regression, these models provide flexible estimates of predictive uncertainty. However, such estimates lack formal statistical guarantees and are often miscalibrated, producing intervals that are either too narrow (undercoverage) or overly conservative (overcoverage).

Conformal Prediction [4] offers a principled, distribution-free framework to address this limitation by enforcing valid coverage guarantees. In particular, Conformalized Quantile Regression (CQR) [5] enables the post-hoc calibration of quantile-based models. Despite its theoretical appeal, standard conformal methods rely on a uniform treatment of past calibration residuals, implicitly assuming that all historical observations are

equally informative. This assumption rarely holds in practice, especially in non-stationary time series, where data distributions evolve due to seasonality, external factors, and structural shifts.

To mitigate this limitation, recent works have proposed dynamic extensions of conformal prediction, including ensemble-based approaches such as EnbPI [6] and SCPI [7], as well as memory-augmented methods like HopCPT [8] and CP-QRNN [9]. While these methods improve adaptability, a fundamental limitation remains: most conformal frameworks are designed for single-step forecasting and struggle to handle the observation lag inherent in multi-step settings. As a result, extending conformal guarantees to H-step ahead horizons remains an open challenge in the literature [10] — in particular, the causal observation lag in online settings is a structural limitation that most existing methods ignore or treat informally.

These limitations become particularly critical in operational environments, where forecasting models must support complex decision-making pipelines rather than act as standalone components. In such contexts, the quality of uncertainty estimates directly affects the reliability and effectiveness of downstream systems. This motivates the need for calibration methods that are not only theoretically sound, but also adaptive to evolving conditions and compatible with real-world deployment constraints.

In this work, we propose a comprehensive extension of the Reservoir Conformal Prediction (ResCP) [11] framework, built upon three fundamental pillars to address these structural challenges.

First, to tackle non-stationarity, our approach leverages Echo State Networks to encode temporal context into a latent representation. This is used to dynamically weight past calibration residuals based on their similarity to the current state. This mechanism is further enriched by incorporating temporal decay functions, allowing the model to prioritize recent observations, and an adaptive blending strategy based on Effective Sample Size (ESS), which prevents over-confident predictions in low-support regimes.

Second, rather than treating multi-step forecasting as an empirical afterthought with uniform scalar corrections, we introduce a theoretically sound causal framework for H-step conformal calibration. Through an asynchronous pending queue mechanism, we causally resolve the observation lag, enabling consistent association between historical states and future residuals. Grounded in the Non-Exchangeable Conformal Prediction (NexCP) theorem, this formulation provides horizon-specific marginal coverage guarantees that dynamically adapt to the natural accumulation of uncertainty over time – representing a principled advancement over the scalar, causally-unsound corrections predominant in existing multi-step work.

Third, to bridge the gap between statistical calibration and operational deployment, we introduce a streaming post-hoc risk controller that continuously adjusts the predicted quantile bounds based on a business-aware cost metric, without requiring model retraining or violating the underlying distributional structure.

We evaluate the proposed methodology on a diverse set of real-world datasets related to energy consumption and production. Experimental results demonstrate that the proposed framework consistently improves calibration while maintaining sharper prediction intervals, particularly in non-stationary and multi-step forecasting scenarios.

The main contributions of this work are summarized as follows:

- We extend the ResCP framework by introducing extended similarity-based weighting and configurable temporal decay mechanisms, improving adaptability to non-stationary dynamics.
- We propose a novel multi-step conformal calibration strategy based on a pending queue mechanism, enabling causal handling of observation lag and providing explicit mathematical bounds that adapt to the growth of uncertainty over time.

- We introduce an adaptive blending scheme based on Effective Sample Size (ESS), ensuring stability by dynamically interpolating between localized and global calibration regimes.
- We incorporate a safety constraint to prevent the collapse of prediction intervals in low-variance conditions, improving robustness.
- We design and implement a complete ResCP pipeline compatible with modern deep learning-based quantile forecasting models.
- We conduct an extensive empirical evaluation on heterogeneous real-world energy datasets, demonstrating the practical effectiveness of the proposed approach.
- We propose an online post-hoc risk calibration layer, inspired by Conformal Risk Control and adaptive feedback mechanisms, that decouples pure statistical learning from business logic by dynamically adjusting predicted quantiles to satisfy predefined operational targets.
- We demonstrate the effectiveness of this controller on challenging real-world scenarios, including extreme distributional shift induced by the COVID-19 pandemic, showing its capacity to track business targets without model retraining.

Chapter 2
Background and Related Work

2.1. Deep Learning for Time Series Forecasting

To establish robust baselines, we selected three specific architectures (TiDE, iTransformer, TSMixer) that approach the extraction of multivariate representations through distinct and complementary paradigms—specifically focusing on the temporal dimension, the feature dimension, and a decoupled combination of both.

- **TiDE (Time-series Dense Encoder)** [1]: An MLP architecture designed to challenge state-of-the-art Transformer-based models for time-series forecasting. By projecting the historical look-back window into a dense representation, it primarily focuses on extracting complex temporal dependencies while maintaining high computational efficiency.
- **iTransformer** [2]: A novel Transformer architecture that applies the self-attention mechanism strictly across the feature (variate) dimension rather than the temporal dimension. This paradigm shift allows the model to capture complex, non-linear correlations between different interacting variables, proving particularly beneficial for highly multivariate forecasting tasks, while offering improved scalability for long look-back windows.
- **TSMixer** [3]: An all-MLP architecture that explicitly bridges the two previous philosophies. By decoupling and alternating between time-mixing and feature-mixing operations, it captures the full spatio-temporal dynamics of the data efficiently, offering a balanced and highly expressive approach to multivariate dependencies.

2.1.1. Time-series Dense Encoder (TiDE)

Recent advancements in Time Series Forecasting have witnessed a notable paradigm shift. While Transformer-based architectures have dominated recent literature due to their attention mechanisms, emerging studies highlight the competitive performance and superior computational efficiency of pure Multi-Layer Perceptron (MLP) models. Among these, the Time-series Dense Encoder (TiDE), introduced by Das et al. [1], represents a state-of-the-art approach that leverages a meticulously designed dense architecture.

Unlike naive MLPs that flatten the temporal dimension indiscriminately, TiDE is specifically engineered to handle multiple covariates—both static attributes and dynamic features—alongside the historical look-back window. The model's architecture can be conceptually divided into three main sequential operations:

1. **Feature Projection and Encoding:** To mitigate the curse of dimensionality, especially when dealing with a large number of exogenous variables, TiDE first employs a residual feature projection block. Past dynamic covariates, future known dynamic covariates (e.g., day of the week, holidays), and static attributes are mapped to a lower-dimensional continuous space. This projected feature set is then concatenated with the historical time series data and fed into a Deep Dense Encoder, which extracts a unified, high-level hidden representation of the global context.
2. **Dense Decoding:** The encoded latent state is subsequently processed by a Dense Decoder. This component maps the high-dimensional hidden representation back into a sequence of future representations, effectively capturing the complex, non-linear temporal dynamics required for the forecasting horizon.
3. **Temporal Projection and Residual Connection:** The final forecasted sequence is generated by combining the decoder's output with a global linear residual connection. This linear path directly maps the

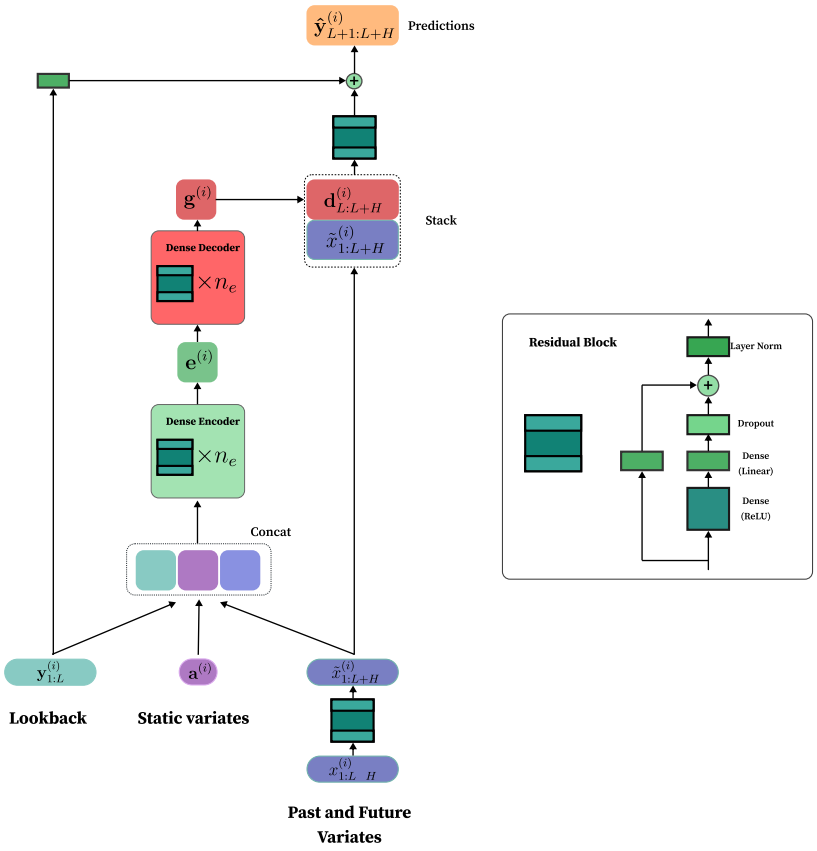


Figure 1: Schematic representation of the TiDE architecture, illustrating the feature projection, dense encoding-decoding process, and the residual connection that preserves linear dynamics.

historical look-back window to the prediction horizon. By doing so, the model ensures the preservation of fundamental linear trends and local dynamics, allowing the dense deep network to focus exclusively on modeling complex, non-linear variations and covariate interactions.

2.1.2. iTransformer and Mixed Covariates Extension

While Multi-Layer Perceptrons like TiDE offer computational efficiency, Transformer-based architectures remain highly relevant for multivariate forecasting due to their ability to model complex dependencies

through self-attention mechanisms. Since the introduction of the original Transformer architecture by Vaswani et al. [12], the attention mechanism has revolutionized sequence modeling. However, its direct application to time series has often struggled with temporal point-wise dependencies. A recent breakthrough in this domain is the iTransformer, proposed by Liu et al. [2].

The core innovation of the iTransformer lies in its "inverted" dimensional paradigm. Traditional Transformers applied to time series typically treat individual time steps as tokens and multivariate features as the embedding dimensions. In contrast, the iTransformer embeds the entire temporal look-back window of each individual variable into a single token. Consequently, the self-attention mechanism operates across the variable (channel) dimension rather than the temporal dimension. This approach naturally captures multivariate correlations and mitigates the risk of attention degradation over long sequences.

Architectural Extension for Future and Static Covariates

A fundamental limitation of the vanilla iTransformer architecture is its native design, which strictly accommodates historical data. It does not provide a built-in mechanism to process known future dynamic covariates (e.g., weather forecasts, calendar features like day-of-the-week) or static covariates, which are often crucial for accurate real-world forecasting.

To overcome this limitation and fully leverage available exogenous information, we introduce an architectural extension to the base iTransformer model. The augmented architecture processes three distinct streams of information:

1. **Past Inverted Embedding:** Following the original design, the historical target and past covariates, each of length L, are projected into a latent space of dimension d_{model}, yielding C_{past} tokens.
2. **Future Inverted Embedding:** Known future covariates over the prediction horizon H are independently normalized and mapped via a

dedicated linear layer to the same d_{model} dimension, producing C_{future} distinct tokens.

3. **Static Inverted Embedding:** Similarly, static features are projected into d_{model} representations, generating C_{static} tokens.

These diverse tokens are concatenated along the token axis, forming a unified input tensor of shape $\left(B, C_{\text{past}} + C_{\text{future}} + C_{\text{static}}, d_{\text{model}}\right)$. By feeding this concatenated sequence into the standard Transformer encoder blocks, the multi-head self-attention mechanism is empowered to explicitly route relevant future and static context to the past historical variables. This allows the model to learn complex interactions between past dynamics and future conditions without artificially conflating their temporal structures.

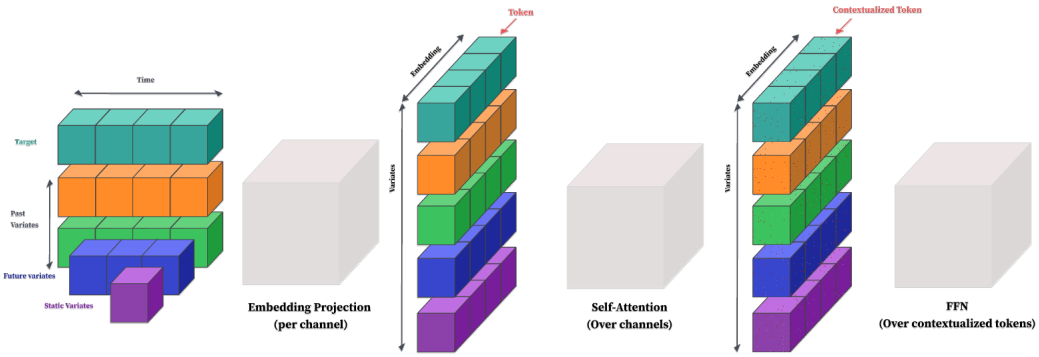


Figure 2: Schematic representation of the extended iTransformer architecture, illustrating the separate embedding and concatenation of past, future, and static covariates before processing through the Transformer blocks.

2.1.3. Time-Series Mixer (TSMixer)

Continuing the exploration of lightweight yet highly effective architectures, the Time-Series Mixer (TSMixer), introduced by Chen et al. [3], represents another significant milestone in pure Multi-Layer Perceptron (MLP) designs for Time Series Forecasting. Inspired by the success of MLP-Mixer models in computer vision, TSMixer demonstrates that complex

self-attention mechanisms are not strictly necessary to capture intricate multivariate dependencies and achieve state-of-the-art performance.

Core Architecture: Decoupling Time and Features

The fundamental design principle of TSMixer relies on explicitly decoupling temporal dynamics from cross-variate correlations. Instead of processing the entire input grid simultaneously or relying on attention matrices, the model passes the multivariate time series through a deep stack of interleaved, specialized MLP layers:

1. **Time-Mixing Operations:** In the first phase of a mixer block, MLPs are applied strictly along the temporal dimension for each feature independently. This operation effectively extracts historical patterns, trend dynamics, and seasonalities without being confounded by cross-channel noise.
2. **Feature-Mixing Operations:** Following the time-mixing step, a distinct set of MLPs operates across the feature (channel) dimension at each time step. This phase is dedicated exclusively to mapping the cross-variate correlations and structural dependencies among the different concurrent time series.

By alternating these time-mixing and feature-mixing blocks—augmented with standard deep learning practices such as residual connections and layer normalization—TSMixer constructs a highly expressive representation of the data. This decoupled approach drastically reduces the computational overhead compared to the quadratic complexity of standard Transformers, making it particularly well-suited for high-dimensional multivariate datasets.

Integration of Exogenous Variables

To address real-world forecasting scenarios that rely heavily on auxiliary data, the architecture incorporates a dedicated alignment and fusion

mechanism (often formalized in the extended architecture, TSMixer-Ext). Static attributes, historical dynamic covariates, and known future covariates are concatenated along the feature dimension.

Before entering the primary mixing backbone, these concatenated features undergo an initial feature-mixing projection. This step aligns the heterogeneous exogenous variables with the primary target series, allowing the subsequent time-and-feature mixing blocks to leverage the full contextual information. Consequently, TSMixer efficiently assimilates complex covariate structures while preserving the linear computational complexity inherent to its pure-MLP design.

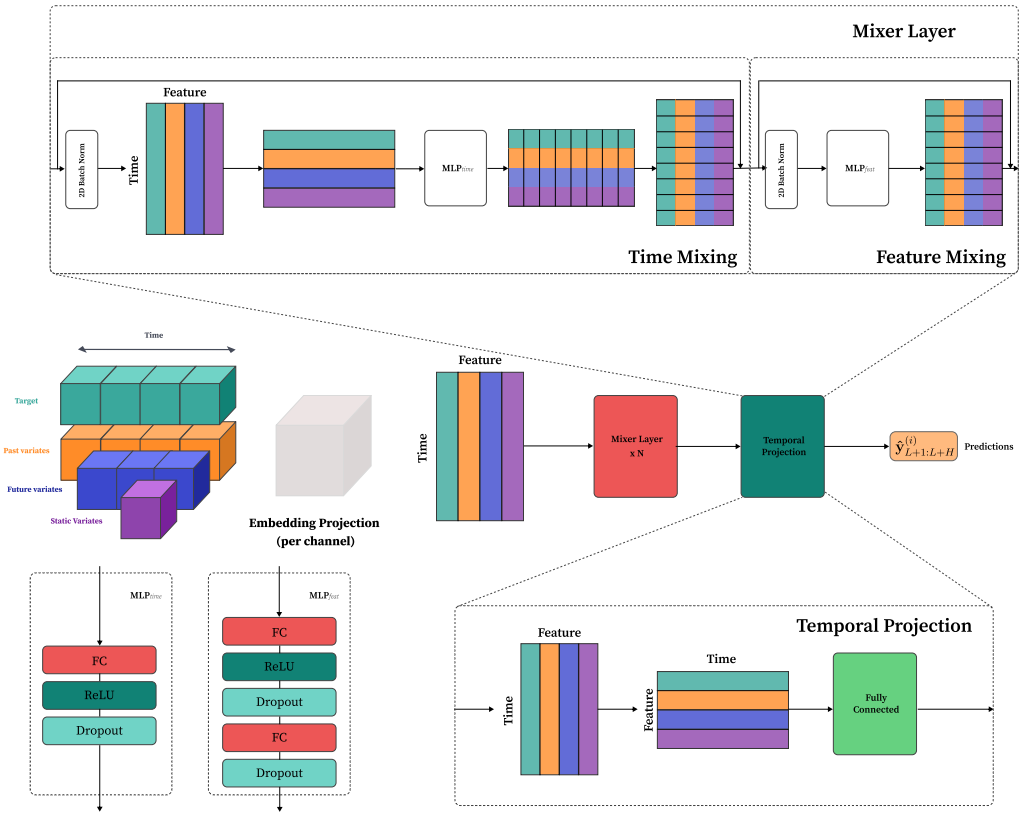


Figure 3: Schematic representation of the TSMixer architecture, illustrating the alternating time-mixing and feature-mixing operations, as well as the initial fusion of exogenous variables.

2.2. Quantile Regression and Calibration Challenges

While the aforementioned architectures – TiDE, iTransformer, and TSMixer – are widely used for deterministic point forecasting, in this work they are specifically adapted to perform Quantile Regression. Instead of outputting a single expected value $\hat{y}_{T+1}$ optimized via Mean Squared Error (MSE), the models are modified at their output heads and trained using the Pinball Loss (also known as Quantile Loss). This adaptation allows the models to directly output specific conditional quantiles, such as the lower bound $\hat{q}_{\frac{\alpha}{2}}$ and the upper bound $\hat{q}_{1-\frac{\alpha}{2}}$, thereby providing an initial, heuristic estimation of predictive uncertainty.

However, pure Deep Learning-based Quantile Regression models suffer from a significant limitation: they lack formal statistical guarantees and often fail to perfectly converge to the true empirical distribution. In practice, deep neural networks are prone to miscalibration. This lack of reliability typically manifests in two ways: **overcoverage**, where the generated intervals are excessively wide and consequently uninformative for practical decision-making, or **undercoverage**, where the intervals are dangerously narrow, leading to an empirical coverage that falls strictly below the nominal target level.

To bridge the gap between heuristic quantile estimation and strict statistical reliability, it becomes strictly necessary to introduce a post-hoc corrective framework. By computing distribution-free corrective terms based on a held-out calibration set, the initial quantile bounds can be systematically adjusted to guarantee valid marginal coverage. This rigorous calibration process, known as Conformal Prediction, constitutes the core methodological focus of the next chapter.

2.3. Conformal Predictions for Time Series Forecasting

Conformal prediction is a powerful framework for quantifying uncertainty in machine learning models. It provides a way to generate prediction sets that are guaranteed to contain the true label with a specified confidence level. This is particularly useful in applications where it is crucial to have reliable uncertainty estimates. In the case of time series forecasting, conformal prediction can be used to create prediction intervals that account for the temporal dependencies in the data. This allows for more accurate and reliable forecasts, which can be critical for decision-making in various domains.

2.3.1. Classical Conformal Prediction

The objective of conformal prediction is to construct a prediction set that contains the true label with a specified confidence level [4]. The key idea is to use a nonconformity measure, which quantifies how unusual a new data point is compared to the training data. By applying this measure to the training data, we can determine a threshold that allows us to construct prediction sets for new data points. In formulas, the objective is to produce, given a miscoverage rate $\alpha \in (0, 1)$ a prediction interval $C_{1-\alpha}(X_{T+1})$ for a new unobserved data point (X_{T+1}, Y_{T+1}) such that

$$\mathbb{P}(Y_{T+1} \in C_{1-\alpha}(X_{T+1})) \geq 1 - \alpha \tag{2.1}$$

To achieve this we split the training data $\mathcal{I}$ into a proper training set $\mathcal{I}_{\text{train}}$ and a so called calibration set $\mathcal{I}_{\text{cal}}$ such that $\mathcal{I} = \{\mathcal{I}_{\text{train}}, \mathcal{I}_{\text{cal}}\}$ and then proceed in steps:

1. Fit a regression model $\hat{\mu}$ using the proper training set $\mathcal{I}_{\text{train}}$.
2. Define a non-conformity score function s that measures the discrepancy or "non-conformity" of the predicted value against the true value. A common choice for regression problems is the absolute error:

$$s(X_i, Y_i) = |Y_i - \hat{\mu}(X_i)| \tag{2.2}$$

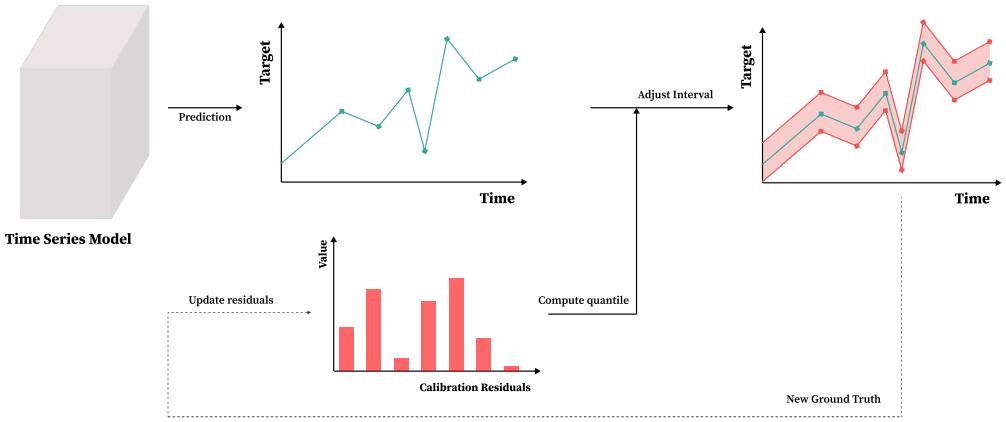


Figure 4: Schematic representation of the Conformal Prediction framework. In real applications, the calibration step is often performed in a rolling window fashion, allowing for dynamic adjustment of the prediction intervals as new data becomes available.

3. Compute the non-conformity scores for the calibration set $\mathcal{I}_{\text{cal}}$ to obtain a set of scores

$$S_{\text{cal}} = \{s_i \mid i \in \mathcal{I}_{\text{cal}}\} \tag{2.3}$$

4. To obtain a coverage level of $1-\alpha$ (i.e. the prediction set should contain the true label with probability at least $1-\alpha$), we need to find the $(1-\alpha)$-th empirical quantile on the calibration scores S_{cal}, denoted as $\hat{q}_{1-\alpha}$.

$$\hat{q}_{1-\alpha} = \text{Quantile}\left(S_{\text{cal}}, (1-\alpha)\left(1 + \frac{1}{|\mathcal{I}_{\text{cal}}|}\right)\right) \tag{2.4}$$

5. Finally apply conformal adjustment to point predictions to obtain the prediction set for a new data point X_{T+1}

$$C_{1-\alpha}(X_{T+1}) = \{y \in \mathcal{Y} : s(X_{T+1}, Y_{T+1}) \leq \hat{q}_{1-\alpha}\} \tag{2.5}$$

- If $s(X,Y) = |Y - \hat{\mu}(X)|$ then the formula collapses to

$$C_{1-\alpha}(X_{T+1}) = [\hat{\mu}(X_{T+1}) - \hat{q}_{1-\alpha}, \hat{\mu}(X_{T+1}) + \hat{q}_{1-\alpha}] \tag{2.6}$$

The potential of conformal prediction is that if $\mathcal{I}_{\text{cal}}$ and $(X_{\text{test}}, Y_{\text{test}})$ are *exchangeable* then

$$\mathbb{P}(Y_{\text{test}} \in C_{1-\alpha}(X_{\text{test}})) \geq 1 - \alpha \tag{2.7}$$

2.3.2. Conformalized Quantile Regression

The objective of Conformalized Quantile Regression (CQR), introduced by Romano et al. [5] is to extend the concept of Conformal Prediction to models that are already fitted using a Quantile Regression approach, predicting quantile points modeling a generic distribution. For simplicity we will explore the base case having a lower and upper quantile regression, but this line of reasoning can be easily extended for multi-paired quantile predictions.

Given any quantile regression algorithm $\mathcal{A}$ we fit the conditional quantiles $\hat{q}_{\alpha_{\text{lo}}}$ and $\hat{q}_{\alpha_{\text{hi}}}$ on the training set $\mathcal{I}_{\text{train}}$, i.e.

$$\left\{\hat{q}_{\alpha_{\text{lo}}}, \hat{q}_{\alpha_{\text{hi}}}\right\} \leftarrow \mathcal{A}\{(X_i, Y_i) : i \in \mathcal{I}_{\text{train}}\} \tag{2.8}$$

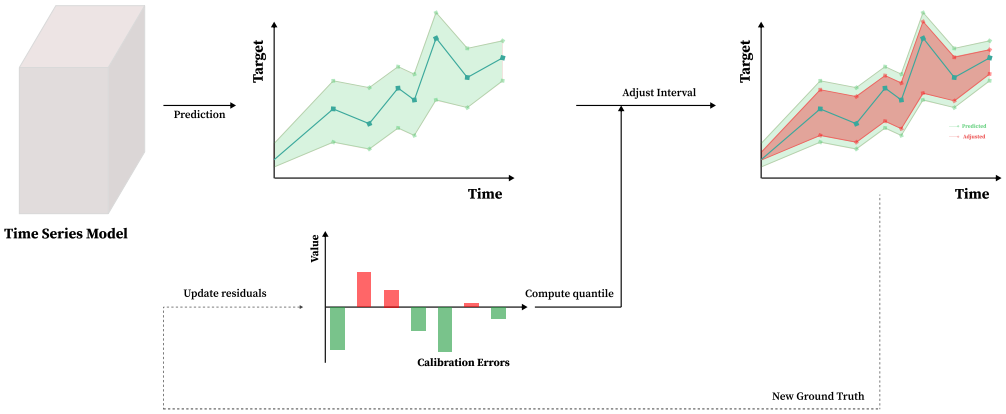


Figure 5: Schematic representation of the Conformalized Quantile Regression (CQR) framework. The non-conformity scores are computed based on the predicted quantiles, allowing for a more flexible adjustment of the prediction intervals.

For the calibration step we compute *conformity scores* on the calibration set $\mathcal{I}_{\text{cal}}$ to quantify the error made by the *inner* model's interval $\hat{C}(x) = \left[\hat{q}_{\alpha_{\text{lo}}}(x), \hat{q}_{\alpha_{\text{hi}}}(x)\right]$. The non-conformity scores are computed differently than the base CP algorithm:

$$E_i := \max\left(\hat{q}_{\alpha_{\text{lo}}}(X_i) - Y_i, Y_i - \hat{q}_{\alpha_{\text{hi}}}(X_i)\right) \quad \forall i \in I_{\text{cal}} \tag{2.9}$$

The conformity score can be interpreted as follows. If $Y_i < \hat{q}_{\alpha_{\text{lo}}}(X_i)$, meaning that the actual value is below the lower quantile, then $E_i = \mid \hat{q}_{\alpha_{\text{lo}}}(X_i) - Y_i \mid$ is the magnitude of such error.

The same line of reasoning can also be applied in the case of $Y_i > \hat{q}_{\alpha_{\text{hi}}}(X_i)$

If $\hat{q}_{\alpha_{\text{lo}}}(X_i) \leq Y_i \leq \hat{q}_{\alpha_{\text{hi}}}(X_i)$ then the non conformity score E_i is the largest, in an algebric sense, between two non positive quantities: this means that CQR can also account for overcoverage other than undercoverage.

We can compute the $(1-\alpha)$-th empirical quantile from the set of non conformity scores $E = \{E_i : i \in \mathcal{I}_{\text{cal}}\}$ as

$$Q_{1-\alpha}(E, \mathcal{I}_{\text{cal}}) := (1-\alpha)\left(1 + \frac{1}{|\mathcal{I}_{\text{cal}}|}\right) \tag{2.10}$$

Finally the predicted quantiles can be adjusted by the computed quantile correction in the calibration set

$$C(X_{T+1}) = \left[\hat{q}_{\alpha_{\text{lo}}}(X_{T+1}) - Q_{1-\alpha}, \hat{q}_{\alpha_{\text{hi}}}(X_{T+1}) + Q_{1-\alpha}\right] \tag{2.11}$$

The new interval mathematically satisfies the desired coverage

$$\mathbb{P}(Y_{T+1} \in C(X_{T+1})) \geq 1 - \alpha \tag{2.12}$$

Under the assumption that the calibration and test data are *exchangeable*. This can be a very strong assumption, all the more so when the calculated residual remains fixed for all the test time or inference.

For this reason, actual implementations of the CQR act as a rolling window on new data as it becomes available, allowing to adjust the calibration in a more dynamic and adaptive fashion.

Remark on Asymmetric Non-Conformity Scores: The formulation above treats lower and upper violations symmetrically by taking their maximum into a single score. An alternative – generally more powerful approach – is to retain the two sides separately:

$$E_i^{\text{lo}} := \hat{q}_{\alpha_{\text{lo}}}(X_i) - Y_i, \quad E_i^{\text{hi}} := Y_i - \hat{q}_{\alpha_{\text{hi}}}(X_i) \tag{2.13}$$

and calibrate each bound independently using its own empirical quantile. This *asymmetric* variant enables independent control of lower and upper coverage, allowing for sharper intervals when the model is more accurate on one side of the distribution.

The main drawback using the *naive* CQR approach, is that each sample used for the calibration is given the same weight, while it may be beneficial to weight the past samples according to the relevance to the current sample. In the next chapters we explore one of these novel approaches, which is the core element of this work.

2.4. Echo State Networks

Echo State Networks (ESNs) are a class of recurrent neural networks characterized by a large, fixed, and randomly connected hidden layer called the *reservoir*. The reservoir serves as a dynamic memory that captures the temporal dependencies in the input data.

A major advantage of ESNs is their ability to maintain and represent recent memory contextually, entirely bypassing the need to train the recurrent weights.

This capacity lies in the *Echo State Property* (ESP), which ensures that the reservoir's state is influenced by recent inputs while asymptotically forgetting older ones. In practice, this behavior is often approximated by

scaling the spectral radius of the reservoir's weight matrix to be less than one, keeping the system at the edge of chaos without diverging.

Various methods have been proposed in the literature to initialize the reservoir weights to satisfy the ESP, including random sparse initialization, orthogonal initialization, and deterministic topologies like the Simple Cycle Reservoir (SCR). Each of these methods offers different trade-offs regarding computational efficiency, memory capacity, and explainability.

Let $x_t \in \mathbb{R}^d$ be the input (or the feature extracted from the base model) at time t. The reservoir maintains an internal state $h_t \in \mathbb{R}^N$, which is updated sequentially according to the standard ESN dynamics:

$$h_t = (1 - r)h_{t-1} + r \tanh(W_{\text{xh}} x_t + W_{\text{hh}} h_{t-1}) \tag{2.14}$$

where $r \in (0, 1]$ is the *leak rate* that controls the speed of the memory update, $W_{\text{xh}} \in \mathbb{R}^{N \times d}$ is the input projection matrix, and $W_{\text{hh}} \in \mathbb{R}^{N \times N}$ is the recurrent reservoir matrix.

2.4.1. Echo State Property

The Echo State Property (ESP) can be mathematically defined as follows [13]: assume an ESN whose internal dynamics are described by a function F. The ESN is said to satisfy the ESP if, for any two initial states h_0 and h'_0, and for any input sequence $s_n = (x_1, x_2, ..., x_n)$, the following holds:

$$\|F(s_n, h_0) - F(s_n, h'_0)\| \to 0 \quad \text{as} \quad n \to \infty \tag{2.15}$$

This implies that the influence of the initial state on the reservoir diminishes over time, leaving the internal state primarily determined by the driving input sequence. This property is crucial to ensure that the ESN effectively captures temporal dependencies without being dominated by its initial, arbitrary conditions.

While the ESP is strictly guaranteed mathematically only if the maximum singular value of the recurrent matrix W_{hh} is strictly less than one (ensuring a global contraction of the state space), in practical applications, satisfying the necessary condition of keeping the spectral radius

$\rho(W_{\text{hh}}) < 1$ is often empirically sufficient to achieve the desired stability and memory properties.

In the following sections, we assume the reservoir is initialized to satisfy the ESP, and we will explore different strategies for achieving this in the next chapter.

2.5. Recent Developments in Sequential Conformal Forecasting

The growing necessity for adaptive uncertainty quantification in non-stationary environments has led to several sophisticated extensions of the Conformal Prediction framework.

- Ensemble and Bootstrap Methods: EnbPI [6] and SCPI [7] represent foundational efforts in sequential calibration. EnbPI leverages bootstrap ensembles to provide coverage without requiring full data exchangeability, while SCPI focuses on prediction intervals for synthetic control methods. However, these approaches often rely on aggregate historical statistics that may react slowly to sudden structural breaks.
- Memory-Augmented Architectures: To better capture temporal context, more recent works have integrated neural memory mechanisms. HopCPT [8] utilizes modern Hopfield Networks to retrieve relevant calibration samples via associative memory. Similarly, CP-QRNN [9], developed by the same authors of the original ResCP framework, explores the use of Quantile Regression Neural Networks for dynamic calibration.
- The Multi-Step Challenge: A common denominator among these SOTA methods is their primary focus on single-step-ahead forecasting ($H = 1$), not because multi-step forecasting is unsupported in practice, but because extending formal coverage guarantees to $H > 1$ horizons remains an open theoretical challenge, as explicitly identified in foundational literature [10]. When applied to multi-step settings, most existing methods

either ignore the observation lag — incorrectly associating calibration states with residuals observed at a different point in time — or apply a scalar, uniform correction across the entire horizon, failing to distinguish between the uncertainty at near and distant steps. Recent work has begun to address related aspects of this challenge from orthogonal angles: [14] extends Adaptive Conformal Inference to the multi-step setting via per-step dynamic threshold adaptation, while [15] bounds the impact of serial autocorrelation between residuals across horizon steps under rolling online updates. To our knowledge, however, no existing work provides an explicit, finite-sample theoretical grounding for the causal admissibility constraint itself — i.e., how the strict exclusion of not-yet-observable residuals from the calibration buffer impacts the validity of weighted quantile estimation in topological or time-adaptive split-conformal frameworks.

This work specifically addresses these gaps by building upon the Echo State Network encoding introduced in the original ResCP framework [11] — which offers a superior trade-off between memory capacity and training efficiency — and by introducing an original asynchronous pending queue mechanism to resolve the multi-step observation lag, together with extended similarity functions, configurable temporal decay profiles, and an adaptive blending strategy based on Effective Sample Size.

Chapter 3
Proposed Methodology: Adaptive Conformal Calibration for Non-Stationary and Multi-Step Forecasting

"Doubt is an uncomfortable condition, but certainty is a ridiculous one."

— **Voltaire,** ***Letter to Frederick II***

In this section, we formalize the Reservoir Conformal Prediction (ResCP) approach [11], inspired by recent advances in sequential calibration for time series. Unlike standard Conformal Quantile Regression (CQR) which assigns uniform weights to calibration residuals, ResCP leverages an Echo State Network (ESN) architecture to extract a latent state h_t that serves as a dynamic memory of the system's conditions.

Furthermore, we introduce a novel adaptive blending mechanism based on Effective Sample Size (ESS) monitoring. This provides strict stability guardrails that dynamically revert to uniform conformal bounds, preventing prediction interval collapse when encountering highly anomalous temporal states. Moreover, the method is applied directly to the outputs of a base quantile model trained using a *Quantile Regression* approach, even

though the method is general and can be applied to any regression model, in the same way in which standard CP can be extended to CQR.

Component	ResCP [11]	This work
ESN encoding	Random sparse	+ Orthogonal, SCR
Similarity weighting	Cosine, Gaussian	+ Laplacian
Temporal decay	basic (linear, exp)	Configurable profiles (power, exp) via temperature parameters
ESS guardrails	-	Yes
Multi-step	-	Yes

Table 1: Comparison of features between the original ResCP and our proposed extension.

3.1. Topological Initialization

Since the weights of an Echo State Network (ESN) are fixed after initialization, properly setting the reservoir weight matrix W_{hh} and the input projection matrix W_{xh} is a crucial step.

To move beyond the standard random sparse approach used in the original ResCP paper, we explore several alternative initialization strategies for W_{hh} found in the literature:

- **Random Sparse:** Weights are sampled from a normal distribution [16], masked to achieve a specific connectivity level c, and scaled so that the spectral radius is $\rho(W_{\text{hh}}) = \rho_{\max} < 1$.

```
W = rng.standard_normal((N, N))
mask = rng.random((N, N)) < connectivity
W = W * mask
rho = np.max(np.abs(np.linalg.eigvals(W)))
W_hh = W * (rho_max / rho)
```

- **Orthogonal:** W_{hh} is derived from the QR decomposition of a Gaussian matrix. This ensures the singular values are exactly 1 (prior to scaling by $\rho_{\max}$), which helps maximize the network's *linear memory capacity*. While the original formulation [17] involves an initial training phase to achieve orthogonality, we use a simplified, training-free approach to preserve the computational efficiency that motivates the use of ESNs in the first place. Crucially, scaling this orthogonal matrix by $\rho_{\max} < 1$ also ensures that all its singular values are strictly less than 1. As discussed, this global contraction mathematically guarantees the strict sufficient condition for the Echo State Property (ESP).

```
Q, _ = np.linalg.qr(rng.standard_normal((N, N)))
W_hh = Q * rho_max
```

- **Simple Cycle Reservoir (SCR):** A deterministic ring topology where each node connects only to its immediate neighbor with a uniform weight of $\rho_{\max}$ [18]. This strictly bounds the hyperparameter space and minimizes computational overhead, while providing a replicable method for initializing such matrix. Despite its simplicity, the SCR topology has been shown to achieve similar performance to more complex random reservoirs, while offering superior stability and interpretability.

```
W_hh = np.diag([rho_max] * (N - 1), k=-1)
W_hh[0, N - 1] = rho_max
```

Finally, following standard practices in the ESN literature, the input projection matrix W_{xh} is initialized by sampling from a standard normal distribution and then rescaling to a specified input scaling factor σ_{in}:

```
W_xh = rng.standard_normal((N, input_dim)) * sigma_in
```

3.2. Similarity and Temporal Decay

Let $D_{\text{cal}} = \{(h_s, r_s)\}_{s=1}^{m}$ be the calibration buffer containing past reservoir states and their corresponding non-conformity scores (quantile residuals) for each point of the forecast horizon. Given a new query state h_t, we quantify the relevance of each past residual r_s through a similarity function.

3.2.1. Similarity Functions

Further expanding upon the original work, we experimented with several similarity functions to evaluate their impact on the model's behavior:

- **Cosine Similarity:** A scale-invariant metric that measures the angular alignment between states.

$$s(h_t, h_s) = \frac{h_t^T h_s}{\gamma \, \|h_t\| \, \|h_s\|} \tag{3.1}$$

- **Gaussian Kernel:** A metric that strongly penalizes dissimilar states due to its rapid exponential decay.

$$s(h_t, h_s) = \exp\left(-\frac{\|h_t - h_s\|^2}{\gamma}\right) \tag{3.2}$$

- **Laplacian Kernel:** Compared to the Gaussian kernel, this function features a slower decay (longer tails), resulting in less severe penalization for dissimilar states.

$$s(h_t, h_s) = \exp\left(-\frac{\|h_t - h_s\|}{\gamma}\right) \tag{3.3}$$

where γ is a temperature parameter that controls the sharpness of the similarity distribution.

3.2.2. Temporal Decay

To handle non-stationary dynamics, the original work [11] introduced a basic temporal decay penalty. In this work, we systematize and extend this

concept by formulating decay as a configurable architectural hyperparameter. Specifically, we propose and compare two distinct decay profiles to better adapt to different speeds of concept drift:

- **Power Law:** Characterized by a slower, more gradual decay that allows older observations to retain influence for longer periods, which can be beneficial in scenarios with slow-changing dynamics.

$$\omega(\Delta) = \frac{1}{(1+\Delta)^{\lambda}} \tag{3.4}$$

- **Exponential:** Characterized by a rapid, aggressive decay that quickly diminishes the influence of older observations.

$$\omega(\Delta) = \exp\left(-\frac{\Delta}{\lambda}\right) \tag{3.5}$$

where $\Delta = t - s$ represents the age of the observation in the buffer and λ is a decay rate hyperparameter that controls how quickly the relevance of past observations diminishes over time.

The final weight w_s for each calibration residual is computed via a softmax function over the similarities, modulated by the temporal decay.

Mathematically, this is expressed as:

$$w_s = \frac{\omega(\Delta_s)\exp(s(h_t, h_s))}{\sum_{j=1}^{m} \omega(\Delta_j)\exp\big(s\big(h_t, h_j\big)\big)} \tag{3.6}$$

3.3. Weighted Calibration Intervals

The weights w_s define a discrete empirical distribution over the calibration residuals. We compute the weighted empirical cumulative distribution function (CDF) to find the weighted correction quantiles $\hat{Q}_{\text{weighted}}$:

$$F_w(r) = \sum_{s=1}^{m} w_s \cdot \mathbb{I}(r_s \leq r) \tag{3.7}$$

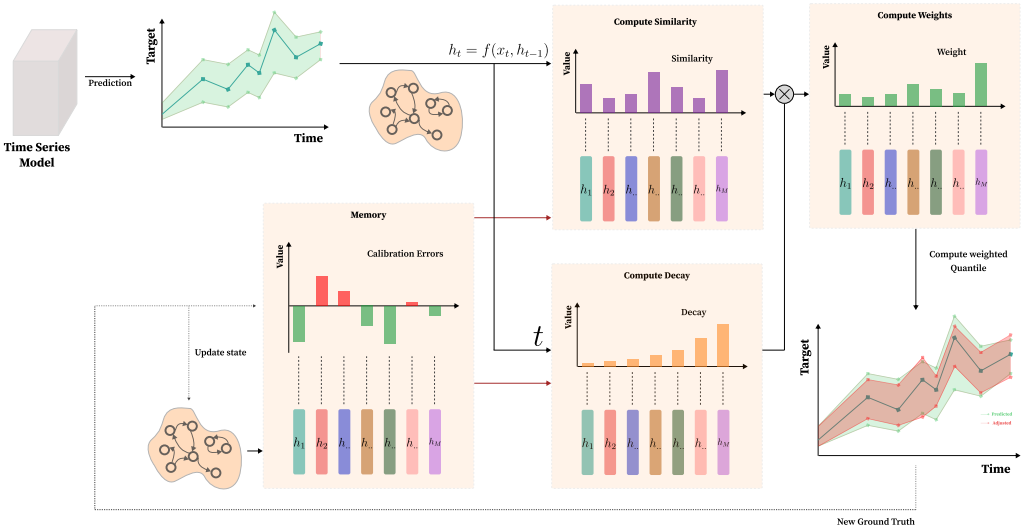


Figure 6: Schematic representation of the Reservoir Conformal Prediction (ResCP) framework. The Echo State Network (ESN) encodes temporal context into a latent state h_t, which is then used to compute similarity-based weights for past calibration residuals. These weights are combined with a temporal decay function to produce a weighted empirical CDF, from which the quantile correction is derived to adjust the base model's predictions.

3.3.1. Adaptive Blending and Stability Guardrails

While the weighted empirical CDF dynamically adapts to the temporal context, highly localized similarity scores can lead to an artificially low Effective Sample Size (ESS). This can result in over-confident and unstable prediction intervals.

To mitigate this, we compute the ESS of the normalized weights:

$$\text{ESS} = \frac{1}{\sum_{s=1}^{m} w_s^2} \tag{3.8}$$

We establish a dynamic target ESS, defined as:

$$E_{\text{target}} = \max\big(E_{\min}, \rho_{\text{target}} \cdot m\big) \tag{3.9}$$

where $E_{\min}$ is an absolute minimum threshold and ρ_{target} is a desired fraction of the available buffer size m.

If the current ESS falls below this target, we blend the weighted quantile shift $\hat{Q}_{\text{weighted}}$ with the naive uniform quantile shift $\hat{Q}_{\text{uniform}}$ (computed using standard unweighted CQR). This is achieved via a blending factor $\beta \in [0, 1]$:

$$\beta = \text{clip}\left(\frac{E_{\text{target}} - \text{ESS}}{E_{\text{target}} - 1}, 0.0, 1.0\right) \tag{3.10}$$

$$\hat{Q}_{\text{blend}} = (1 - \beta) \cdot \hat{Q}_{\text{weighted}} + \beta \cdot \hat{Q}_{\text{uniform}} \tag{3.11}$$

Finally, to prevent the intervals from collapsing excessively compared to the baseline conformal scale, we enforce a safety floor proportional to the uniform quantile shift:

$$\hat{Q} = \max\left(\hat{Q}_{\text{blend}}, \rho_{\text{floor}} \cdot \hat{Q}_{\text{uniform}}\right) \tag{3.12}$$

While the weighted empirical CDF dynamically adapts to the temporal context, similarity-based methods are susceptible to "over-confidence traps" in low-variance regimes. If the query state h_t strongly matches historical states that incidentally exhibit near-zero residuals, the resulting empirical distribution can cause the prediction interval to collapse to an infinitesimal width.

While such narrowing reflects high conditional confidence, taking it to the extreme ignores the intrinsic aleatoric noise always present in real-world time series. A fully collapsed interval is structurally brittle and likely to violate coverage guarantees upon encountering even minor unforeseen variance.

To prevent overly optimistic forecasts, we introduce this adaptive robust floor. By bounding the dynamically weighted quantile shift to a minimum fraction of the global uniform shift (typically $\rho_{\text{floor}} = 0.5$), we effectively bridge the gap between conditional and marginal coverage.

This hybrid approach allows the model to aggressively shrink uncertainty during highly predictable regimes—such as halving the baseline interval width—while retaining a structural safety net that preserves the fundamental reliability guarantees of Conformal Prediction.

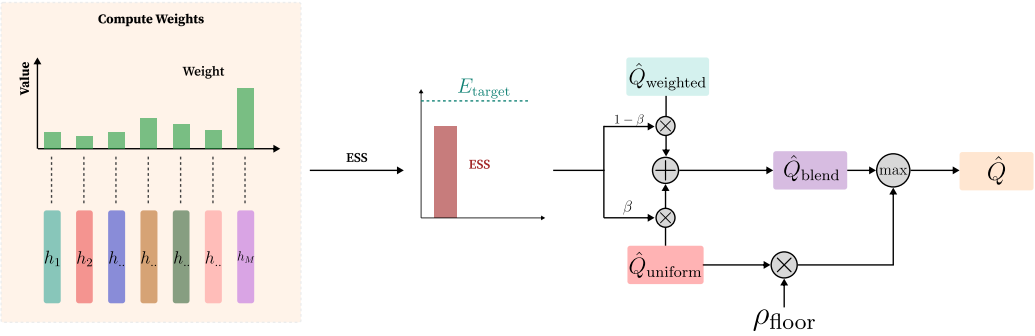


Figure 7: Schematic representation of the Adaptive Blending and Safety Guardrails mechanism. The Effective Sample Size (ESS) is monitored continuously to prevent over-confidence traps in low-variance regimes. When the ESS drops below the dynamic threshold E_{target}, a blending factor β is computed to smoothly interpolate between the high-variance topological shift ($\hat{Q}_{\text{weighted}}$) and the robust uniform baseline ($\hat{Q}_{\text{uniform}}$). A final max operator enforces a structural safety floor (ρ_{floor}), ensuring the prediction intervals do not collapse and mathematically guaranteeing the baseline marginal coverage.

For a given confidence level, these robust calibration shifts (calculated separately for the lower and upper bounds, denoted as $\hat{q}_{\text{lo}}$ and $\hat{q}_{\text{hi}}$) are then added to the base model's heuristic predictions:

$$\hat{C}(X_{n+1}) = \left[\hat{Y}^{(q_{\text{lo}})}_{n+1} - \hat{q}_{\text{lo}}, \quad \hat{Y}^{(q_{\text{hi}})}_{n+1} + \hat{q}_{\text{hi}}\right] \tag{3.13}$$

3.3.2. Conformalized Quantile Regression as a Special Case

A fundamental theoretical property of the Reservoir Conformal Prediction (ResCP) framework is that it strictly generalizes the standard Conformalized Quantile Regression (CQR) approach. Rather than being mutually

exclusive methodologies, sliding-window CQR can be mathematically derived as a specific, boundary-condition instance of ResCP.

Recall that the ResCP calibration weights w_s are determined by the interplay of topological similarity and temporal decay. If we configure the framework to suppress both mechanisms—specifically, by utilizing a uniform similarity metric (e.g., setting the similarity temperature $\gamma \to \infty$) and disabling the temporal decay (e.g. $\lambda \to \infty$ for the power-law decay) —the architectural behavior collapses.

Under these conditions, the network's latent states h_s lose their discriminative power, and the temporal relevance becomes flat. Consequently, the softmax function distributes the probability mass evenly, assigning a uniform weight $w_s = \frac{1}{m}$ to every residual in the calibration buffer. As illustrated in Figure 8, the dynamically weighted empirical CDF seamlessly degenerates into the standard empirical CDF. Therefore, the uniform sliding-window CQR is simply a ResCP model operating under the restrictive assumption that all past temporal states are equally informative.

3.4. Causal Resolution of Multi-Step Conformal Calibration

In classical Conformal Prediction applied to time series, the non-conformity score is often treated as a scalar value. However, in multi-step forecasting, aleatoric uncertainty is inherently non-stationary and typically grows monotonically as the forecast horizon H increases. Applying a uniform, scalar calibration shift across all future time steps would inevitably lead to prediction intervals that are overly conservative for the near future and dangerously narrow for the distant future, failing to provide optimal local coverage.

Furthermore, in an online sequential setting, a structural criticality emerges known as the *observation lag*. The prediction $\hat{y}_{t+h}$ generated

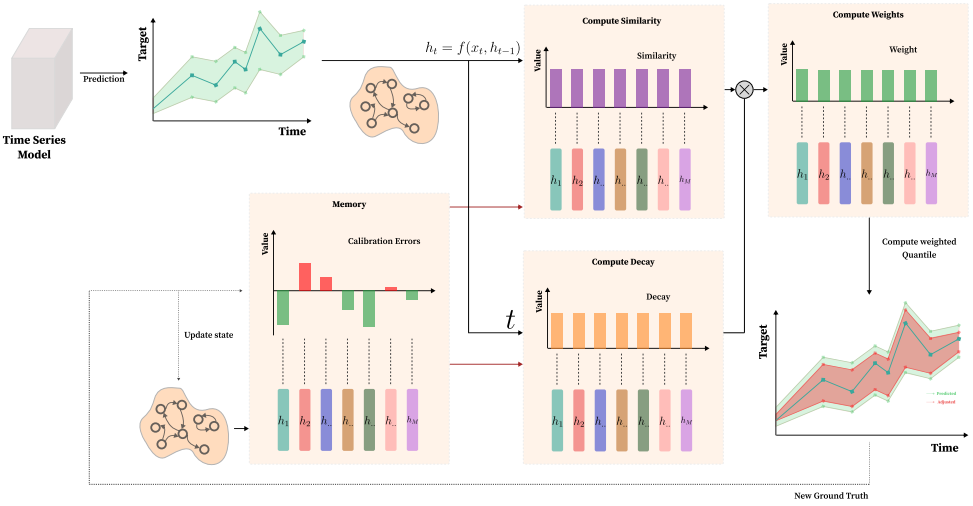


Figure 8: Visual demonstration of CQR as a special case of ResCP. When both the Similarity and Temporal Decay mechanisms are parameterized to yield flat distributions, the resulting calibration weights become perfectly uniform. In this boundary scenario, the ResCP framework exactly replicates the behavior of a standard sliding-window CQR, applying a static and constant quantile shift regardless of the local variance.

for a future horizon h from the latent state h_t cannot be validated until the actual time $t + h$ is reached. During this interval, the reservoir state continues to evolve driven by new incoming data. Without a proper mechanism, it becomes impossible to causally associate the state at time t with the error effectively observed h steps later.

While existing literature predominantly treats this multi-step observation lag as an empirical afterthought, relying on structurally suboptimal scalar corrections, we formalize a theoretically sound causal framework. By grounding our framework in the Non-Exchangeable Conformal Prediction (NexCP) [19] theorem, our methodology provides explicit mathematical bounds that dynamically adapt to the monotonic accumulation of temporal uncertainty, effectively bridging the gap between sequential online inference and rigorous H-step coverage guarantees.

3.4.1. Causal Residual Admissibility Framework for Multi-Step Conformal Calibration

To preserve causal rigor and prevent *data leakage* without introducing artificial delays, our ResCP implementation introduces an asynchronous architectural mechanism based on a dynamic **Pending Queue** paired with **In-Place Buffer Propagation**.

The online calibration and update process is strictly decoupled into two continuous phases:

1. **Query and Allocation Phase (Time t):** The model extracts the current reservoir state h_t to compute similarity weights and generate the multi-step forecast over the horizon H. Immediately, the "clean" state h_t is appended to the core calibration buffer, paired with a pre-allocated empty residual matrix $\mathbf{R}_t$. Concurrently, a reference to this state and its mutable residual matrix is pushed into a FIFO *Pending Queue*.
2. **Dynamic Real-Time Resolution Phase (Every Step):** At each subsequent timestep, rather than waiting for the entire horizon H to mature, the system loops through the pending queue and evaluates the intersection between past forecasts and the newly arrived ground truth. As time progresses, the rows of the matrix $\mathbf{R}_t$ mature sequentially. The matured non-conformity scores are written **in-place** into the matrix. Once a pending entry has completely filled all its H horizon rows, it is definitively popped from the queue.

This mechanism ensures that partially matured calibration pairs are immediately available to the system, maximizing sample efficiency while guaranteeing that the topological similarity function strictly operates on causal and admissible information.

3.4.2. Multi-Step Residual Buffering

By causally resolving the observation lag step-by-step, the framework redefines the calibration buffer $\mathcal{D}_{\text{cal}}$. Unlike standard implementations,

our buffer stores full error matrices that are progressively populated in real time:

$$\mathcal{D}_{\text{cal}} = \{(h_s, \mathbf{R}_s)\}_{s=1}^{m} \tag{3.14}$$

Where each element $\mathbf{R}_s$ represents the matrix of non-conformity residuals computed for each predictive step, which may contain unobserved values for horizons that have not yet matured:

$$\mathbf{R}_s = \left[r_{s\to s+1}, r_{s\to s+2}, ..., r_{s\to s+H}\right]^T \tag{3.15}$$

During inference at time t, the topological similarity and temporal decay weights $\mathbf{w}$ are computed once with respect to the query state h_t over all states in the buffer. However, the conformal adjustment is computed and applied *independently* for each temporal slice $h \in \{1, ..., H\}$.

To respect causality, the weighted empirical quantile evaluates exclusively on matured steps:

$$\hat{q}_{\text{w}}^{(h)} = \text{Quantile}\left(\left\{\mathbf{R}_s^{(h)}\right\}_{s\in\mathcal{I}_t^{(h)}}, \mathbf{w}, 1-\alpha\right) \;\; \forall h \in \{1, ..., H\} \tag{3.16}$$

This matrix-based formulation empowers the ResCP framework to dynamically adapt to the temporal heteroscedasticity of the series. For any given latent state h_t, the system autonomously learns to enforce wider corrections for distant time steps and tighter bounds for imminent ones, effectively optimizing the trade-off between coverage reliability and interval sharpness across the entire forecasting horizon.

3.4.3. Theoretical Guarantees for Multi-Step Resolution

Let t be the current timestep and $h \in \{1, 2, ..., H\}$ the predictive horizon step. We aim to generate a valid confidence interval for the future target value Y_{t+h}, conditioned on the current reservoir state h_t.

Due to the observation lag inherent in multi-step forecasting, the most recent admissible residual for a specific horizon h, observed at time t, is the

one generated from the state at time $t-h$. Therefore, the valid calibration set for the h-th step is defined as:

$$\mathcal{I}_t^{(h)} = \{s \in \mathbb{N} \mid s \leq t-h\} \tag{3.17}$$

Remark on Buffer Completeness: In our dynamic architecture, this condition is enforced naturally by the data availability itself: for any past state where $s > t-h$, the h-th slice of $\mathbf{R}_s$ will not yet be available, thus being automatically excluded from the quantile estimation. For a horizon h, the valid calibration set excludes the most recent h observations, reducing the effective buffer size to $m-h$ entries.

This reduction has two compounding consequences. First, at large horizons the empirical distribution is estimated from fewer points, increasing variance in the quantile estimate. Second, and more critically, the NexCP framework provides meaningful coverage guarantees only asymptotically as the calibration buffer grows large: with a small effective buffer, the practical error $\varepsilon_{\text{prac}}^{(h)}$ can be arbitrarily close to 1, rendering the bound nearly vacuous.

In practice, this motivates choosing a window size m substantially larger than the forecasting horizon H, so that even for the most distant step $h = H$ the residual buffer remains large enough to support reliable weighted quantile estimation. In our experiments, we adopt $m = 500$ with $H = 24$, ensuring a minimum effective buffer size of 476 — well above any reasonable threshold for stable quantile estimation.

To formalize the validity of our weighting mechanism, we rely on the Non-Exchangeable Conformal Prediction (NexCP) framework introduced by Barber et al. (2022) [19]. We assign to each calibration point s an unnormalized weight $v_s^{(h)}$, defined in our architecture as:

$$v_s = \omega(t-s)\exp(s(h_t, h_s)) \tag{3.18}$$

which strictly corresponds to the unnormalized version of the calibration weights used in ResCP.

The NexCP theorem requires evaluating this weight function also for the test point at time t (i.e., the future state we are predicting from). In our framework, the test weight evaluates the query state against itself:

$$v_t = \omega(0) \exp(s(h_t, h_t)) \tag{3.19}$$

which correctly yields the maximum achievable weight, serving as a theoretical anchor.

Let $P_s^{(h)} = \mathbb{P}\Big(R_s^{(h)} \mid h_s\Big)$ and $P_t^{(h)} = \mathbb{P}\Big(R_t^{(h)} \mid h_t\Big)$ denote the conditional distributions of the h-step forecasting residuals, given the reservoir states at time s and t, respectively. The NexCP theorem guarantees that if we construct the prediction intervals using the theoretical normalized weights:

$$\tilde{w}_s^{(h)} = \frac{v_s^{(h)}}{v_t^{(h)} + \sum_{j \in \mathcal{I}_t^{(h)}} v_j^{(h)}} \tag{3.20}$$

then the marginal coverage probability is strictly lower-bounded by:

$$\mathbb{P}\Big(Y_{t+h} \in \hat{C}_t^{(h)}(X_{t+h}, 1-\alpha)\Big) \geq 1 - \alpha - \varepsilon_{\text{exact}}^{(h)} \tag{3.21}$$

Here, $\varepsilon_{\text{exact}}^{(h)}$ is the theoretical distributional error, defined as the weighted sum of the Total Variation (TV) distances between the calibration distributions and the test distribution:

$$\varepsilon_{\text{exact}}^{(h)} = \sum_{s \in \mathcal{I}_t^{(h)}} \tilde{w}_s^{(h)} \cdot d_{\text{TV}}\Big(P_s^{(h)}, P_t^{(h)}\Big) \tag{3.22}$$

where $d_{\text{TV}}\Big(P_s^{(h)}, P_t^{(h)}\Big) = \sup_A | \ P_s^{(h)}(A) - P_t^{(h)}(A) \ |$. This theorem provides a powerful mathematical guarantee: the coverage error is bounded by the distributional shifts across the calibration points, modulated by their topological and temporal relevance to the current state.

Remark on Horizon Scaling: The framework naturally accommodates the empirical observation that forecasting uncertainty tends to grow with the horizon h. In non-stationary settings, the conditional distribu-

tions of residuals at distant steps are generally more sensitive to latent state shifts than those at near steps, due to the autoregressive accumulation of aleatoric uncertainty and the temporal amplification of concept drift. While a formal monotonicity guarantee on $\varepsilon_{\text{exact}}^{(h)}$ would require additional distributional assumptions that fall outside the scope of this work, the horizon-specific calibration structure introduced here is well-motivated: by computing independent quantile corrections for each h, the framework avoids the structural suboptimality of a uniform scalar shift, which would inevitably over-correct for near horizons and under-correct for distant ones.

Asymptotic Approximation (Practical Implementation)

In a practical online setting, the ResCP weights are computed using a standard weighted softmax over the available calibration buffer, inherently omitting the currently unobservable test point from the denominator:

$$w_s^{(h)} = \frac{v_s^{(h)}}{\sum_{j \in \mathcal{I}_t^{(h)}} v_j^{(h)}} \tag{3.23}$$

This operational approach yields a practical coverage error bound, defined as:

$$\varepsilon_{\text{prac}}^{(h)} = \sum_{s \in \mathcal{I}_t^{(h)}} w_s^{(h)} \cdot d_{\text{TV}}\left(P_s^{(h)}, P_t^{(h)}\right) \tag{3.24}$$

We can demonstrate that this practical implementation maintains the mathematical soundness of the framework by providing a valid, albeit slightly looser, upper bound on the distributional error. We provide the following demonstration to offer an intuition of this bound while remaining strictly within the NexCP framework; for the complete and formal proof, we refer the reader to the original ResCP paper [11]. By isolating the denominator, we establish the relationship between the theoretical and practical weights:

$$\begin{aligned}\tilde{w}_s^{(h)} &= \frac{v_s^{(h)}}{v_t^{(h)} + \sum_{j \in \mathcal{I}_t^{(h)}} v_j^{(h)}} \\ &= \frac{v_s^{(h)}}{\sum_{j \in \mathcal{I}_t^{(h)}} v_j^{(h)}} \cdot \frac{\sum_{j \in \mathcal{I}_t^{(h)}} v_j^{(h)}}{v_t^{(h)} + \sum_{j \in \mathcal{I}_t^{(h)}} v_j^{(h)}} \\ &= w_s^{(h)} \cdot \underbrace{\frac{\sum_{j \in \mathcal{I}_t^{(h)}} v_j^{(h)}}{v_t^{(h)} + \sum_{j \in \mathcal{I}_t^{(h)}} v_j^{(h)}}}_{<1}\end{aligned} \tag{3.25}$$

Since $v_t^{(h)} > 0$, the scaling fraction is strictly less than 1. Therefore:

$$\tilde{w}_s^{(h)} < w_s^{(h)} \quad \forall s \in \mathcal{I}_t^{(h)} \tag{3.26}$$

which strictly implies that the weighted error sum grows:

$$\varepsilon_{\text{exact}}^{(h)} < \varepsilon_{\text{prac}}^{(h)} \tag{3.27}$$

By substituting this inequality into the original NexCP guarantee, we obtain the final coverage bound for the ResCP multi-step implementation:

$$\mathbb{P}\Big(Y_{t+h} \in \hat{C}_{t,1-\alpha}^{(h)}(X_{t+h})\Big) \geq 1 - \alpha - \varepsilon_{\text{exact}}^{(h)} > 1 - \alpha - \varepsilon_{\text{prac}}^{(h)} \tag{3.28}$$

This mathematically guarantees that the practical approximation omitting the test weight simply acts as a more conservative estimator of the error bound, fully preserving the theoretical integrity and safety of the multi-step conformal framework.

3.4.4. Online Inference Cycle

The strict separation between state advancement and lag resolution allows the model to be efficiently deployed in a real-time production environment. During online inference:

1. **State Update:** The ESN acts as a continuous streaming memory, updating h_t at every time step with the latest available ground truth.

2. **Uncertainty Estimation:** Forecasts are calibrated using the current state h_t in a purely read-only manner, evaluating quantiles exclusively over the valid matured historical residuals available at that specific moment.
3. **Asynchronous Calibration Update:** The core calibration buffer matrices are dynamically updated in-place as individual steps mature, removing the need for batch steps or structural pauses.

3.5. Post-hoc Calibration for Risk-Sensitive Decision-Making

In real-world operational scenarios, the costs of overestimation and underestimation are often highly asymmetric. In energy load forecasting, for instance, underestimating peak demand can trigger blackouts and severe disruptions, while overestimation typically leads to increased operational costs without immediate safety risks.

3.5.1. Decoupling Prediction from Risk Control

Traditional approaches often attempt to embed asymmetric loss functions directly into the training of predictive models. However, this tightly couples the prediction task with the business risk control strategy, optimizing for potentially volatile and context-specific cost structures.

Our approach is philosophically inspired by Conformal Risk Control (CRC) [20], which extends conformal prediction to control the expected value of any monotone loss function beyond simple miscoverage – allowing the practitioner to target arbitrary, application-specific risk measures. However, CRC operates in a batch setting under exchangeability assumptions, which are often violated in non-stationary operational environments such as energy load forecasting.

To address this, we adopt the online update mechanism of Adaptive Conformal Inference (ACI) [21], which employs a control-theoretic feedback rule to continuously adjust the prediction threshold in a streaming

fashion. Our method adapts this idea to operate directly on the predicted quantiles via an additive shift, driven by business metrics rather than coverage targets, and without requiring exchangeability or formal statistical guarantees.

This framework allows the base model (in our case the conformal predictor) to focus purely on capturing statistical dynamics, while the post-hoc calibration layer dynamically adjusts quantiles to satisfy predefined business targets in a non-stationary streaming setting.

3.5.2. Dynamic Shift Update Mechanism

The core of the proposed framework relies on an adaptive shift applied to the predicted quantiles. Let E_B be the empirical "business" metric evaluated over a sliding window of size B, and M be the target metric value defined by the business requirements (e.g. underestimation cost or hourly aware cost for the energy load forecasting case). For the upper quantiles ($q > 0.5$), the controller maintains a non-negative shift w_t that is updated at each time step t based on an error function r:

$$w_{t+1} = \max(0, \gamma w_t + \eta \cdot r(E_B, M)) \tag{3.29}$$

Where $\gamma \in (0, 1)$ is a decay factor that controls the memory of past adjustments, $\eta > 0$ is a learning rate that determines the sensitivity of the updates, and $r(E_B, M)$ is a risk function that quantifies the deviation of the empirical metric from the target. In our experiments, we use a simple linear relative error function defined as $r(E_B, M) = \frac{E_B - M}{M}$, which provides a normalized measure of the performance gap, while other risk functions can be easily integrated into the framework depending on the specific business context. This allows the controller to cautiously decrease the safety margin when the performance is better than the target, while aggressively increasing it when the performance deteriorates, thus providing a dynamic and responsive calibration strategy.

To avoid excessive reactivity to short-term fluctuations, we also implemented a safeguard that clips the risk function to a predefined threshold,

ensuring that the shift adjustments remain within reasonable bounds and preventing overfitting to transient noise in the data.

To further promote conservative behaviour, the update rule implements an asymmetric learning rate: when the empirical metric outperforms the target (i.e. the shift could safely decrease), the effective step size is attenuated by a factor $\beta \in (0, 1]$,

$$\eta_t = \begin{cases} \eta & \text{if } r(E_B, M) \geq 0 \\ \beta \cdot \eta & \text{if } r(E_B, M) < 0 \end{cases} \tag{3.30}$$

This asymmetry causes the controller to widen the prediction interval aggressively when the business target is violated, but to narrow it cautiously when performance is satisfactory, reducing the risk of prematurely relaxing the safety margin in response to transient improvements.

A parallel and independent update rule governs the lower quantiles ($q < 0.5$), where a non-positive shift is maintained to control underestimation risks. The decoupling of the upper and lower quantile adjustments allows for asymmetric calibration that can be tailored to the specific cost structure of overestimation and underestimation in the application domain.

3.5.3. Guaranteeing Quantile Monotonicity and Stability

A critical challenge in post-hoc quantile calibration is the risk of quantile crossing or violating the central tendency. To preserve the structural integrity of the probabilistic forecast, the additive shift is constrained point-by-point. The applied shift to a given quantile q is bounded relative to its distance to the median $q_{0.5}$:

$$s = \min(w_t, \kappa \ |q - q_{0.5}|) \tag{3.31}$$

Where κ is the maximum relative margin cap. This bounding mechanism ensures that the quantiles are never pushed too far away from the median, and that the magnitude of the distortion is proportional to the quantile's distance from the center, thus preserving the overall shape and

coherence of the predictive distribution while allowing for flexible adjustments to meet business targets.

Note on units and scaling. Since the risk function $r(E_B, M) = \frac{E_B - M}{M}$ is a dimensionless relative error, the shift w_t accumulates dimensionless increments. For the additive correction to be meaningful, w_t must therefore live on the same scale as the predicted quantiles. In the general case – e.g. when predictions are in absolute units such as MW – a direct addition would be dimensionally inconsistent, and one would need to rescale w_t by a representative factor of the target variable (e.g. its local median). In our setting, predictions are Min-Max scaled to $[0, 1]$, so w_t and the quantiles share the same dimensionless scale and no further adjustment is required.

Chapter 4
Datasets and Evaluation Framework

"Observe constantly that all things take place by change, and accustom yourself to consider that the nature of the Universe loves nothing so much as to change the things which are, and to make new things like them."

— **Marcus Aurelius, *Meditations***

4.1. Exploratory Data Analysis

This exploratory analysis focuses on the benchmark datasets used in our experiments. During the initial screening phase, we surveyed a wide range of candidate datasets. Our final selection was strictly guided by data cleanliness, structural integrity, and overall relevance to the study's scope. The goal of this section is to characterize the statistical diversity of this carefully curated benchmark.

As illustrated by the descriptive statistics, the selected datasets exhibit substantial variability in volatility, power scales, and temporal regularity. This structural heterogeneity is essential for our evaluation, as it allows us to comprehensively test the robustness of ResCP across highly diverse operating conditions.

4.1.1. The Renewables Distinction

The renewable-energy category features a single, distinct dataset: **Solar PV PoliTO**. Unlike the rest of the benchmark, which tracks energy **consumption**, this dataset measures energy **production**. We intentionally retained it to serve as a targeted stress test. Its strong sensitivity to weather

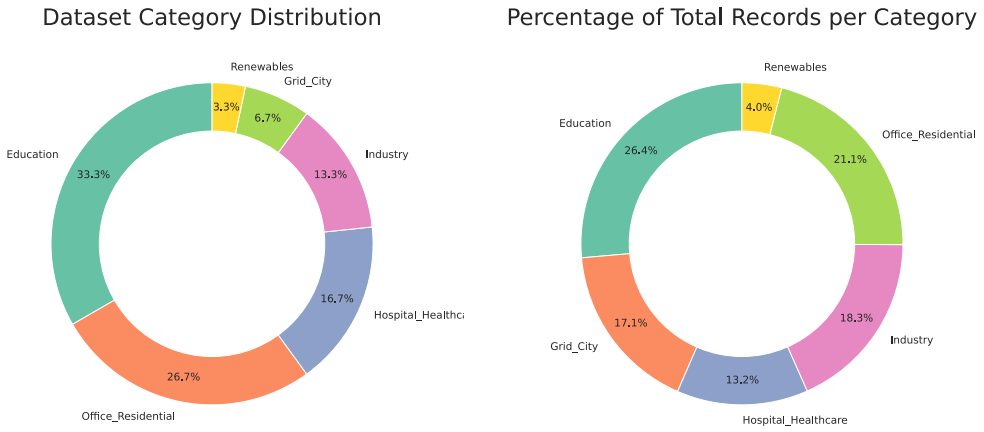


Figure 9: Distribution of the final selected datasets by functional category (left) and total recorded hours per category (right).

covariates and higher volatility allow us to assess model robustness and behavior under a markedly different statistical regime.

4.2. Datasets

For our experimental evaluation, we primarily selected datasets from the **Building Data Genome Project 2** [22]. This data comprises histor-

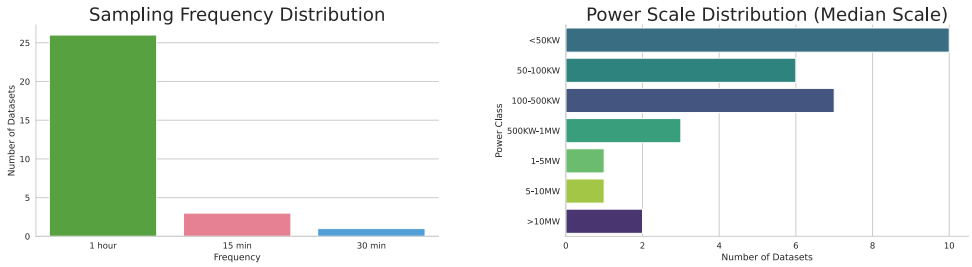


Figure 10: Distribution of temporal sampling frequencies (left) and median power scales (right), highlighting the structural heterogeneity of the final benchmark.

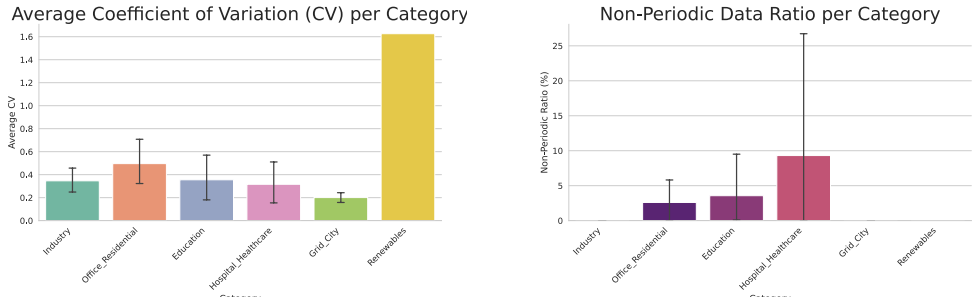


Figure 11: Box plots of the Coefficient of Variation (CV) per category (left) and the ratio of non-periodic intervals (right).

ical energy consumption time series for offices, schools, industrial facilities, and healthcare buildings, anonymized for privacy reasons. Among the available original covariates, we retained only the air temperature (`airTemperature`), as it proved to be the only variate (weakly) correlated with the target metric.

To test the robustness and generalization capabilities of our methodology across different domains, we integrated additional datasets characterized by heterogeneous scales and dynamics:

- **Zurich**: Contains electricity consumption data from the city of Zurich, divided into residential (NE5) and industrial (NE7) users.
- **Australia**: Reports the aggregated electricity consumption of the city of Sydney (`SYSLoad`). Compared to BDG2, it offers a macroscopic perspective (an entire metropolis) rather than focusing on a single building.
- **Solar PV PoliTo**: Obtained in collaboration with *Politecnico di Torino*, this dataset collects energy production data from the university campus's solar panels. Unlike the previous cases, this dataset focuses on energy **production** rather than consumption, allowing us to validate our approach on a phenomenon governed by inherently different physical and environmental dynamics.

The inclusion of such a diversified pool of data presents significant challenges, but it allows us to evaluate the performance of our methodology

Dataset	Target	Covariates	Train (Start–End)	Val (Start–End)	Cal (Start–End)	Test (Start–End)
BDG2 (Various buildings)	Electricity Consumption	Air Temp	2016-01-01 2017-01-01	2017-01-01 2017-03-31	2017-04-01 2017-04-30	2017-05-01 2018-01-01
Zurich (NE5, NE7)	Electricity Consumption	Rel. Hum., Solar Rad., Air Temp, Atm. Press., Wind Speed, Wind Dir.	2018-01-01 2019-09-01	2019-09-01 2019-12-31	2020-01-01 2020-01-31	2020-02-01 2020-12-31
Australia (Sydney)	Electricity Consumption	Dry Bulb, Dew Point, Wet Bulb, Hum.	2006-01-02 2007-09-30	2007-10-01 2007-12-31	2008-01-01 2008-01-31	2008-02-01 2008-12-31
Solar PoliTo	Solar PV	Air Temp, Glob. Solar Rad., Diff. Solar Rad., Dir. Solar Rad., Horiz. Dir. Solar Rad., Illuminance	2023-01-01 2023-12-31	2024-01-01 2024-01-31	2024-02-01 2024-03-01	2024-03-02 2024-12-31

Table 2: Summary of the datasets used for the experiments, detailing targets, covariates, and the specific start and end dates for each subset split.

across a wide range of scenarios, providing strong empirical validation for the generality of our approach.

Table 2 provides a structured summary of the datasets used in the experiments. For each dataset, we report the target variable, the covariates employed, and the chronological splitting criteria defined to separate the training (Train), validation (Val), calibration (Cal) and test (Test) sets.

We provide a visual representation of the temporal dynamics of the target datasets in Figure 12, where we also highlight the temporal splits defined for training, validation, and calibration/test sets. This visualization allows us to qualitatively assess the variability and trends in the target variables across different datasets, as well as to verify the appropriateness of the defined splits in capturing different temporal patterns and ensuring a robust evaluation of our methodology.

4.2.1. Scaling and Normalization

To ensure that the models can effectively learn from the data, we applied a `MinMaxScaler` to all the datasets, normalizing the target variable and covariates to a common scale. This transformation is crucial for models that are sensitive to the scale of the input data, such as neural networks,



Figure 12: *All datasets: target dynamics and highlighted temporal splits. Note how the datasets exhibit a wide range of temporal patterns, scales, and variability, which poses significant challenges for forecasting models. Highlighted datasets (Orange) are those selected for preliminary grid search analysis*

and it helps to improve convergence during training. The `MinMaxScaler` rescales the data to a specified range, typically [0, 1], by applying the following formula:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{4.1}$$

where X is the original value, $X_{\min}$ and $X_{\max}$ are the minimum and maximum values of the feature in the training set, respectively. This normalization process preserves the relative relationships between data points while ensuring that all features contribute equally to the model's learning process.

4.2.2. Time encodings

We enriched all the datasets with time-based covariates, specifically providing them for the future time steps, to enable the models to capture temporal patterns and seasonality. We included the following time encodings: Hour of the day, Day of the week, Day of the month, Day of the year, Month of the year, Week of the year

In principle it is possible to also include past time encodings, but literature generally indicates that they do not provide significant benefits, as the model can already infer them from the temporal structure of the input data and from the future time encodings. Therefore, we opted to include only future time encodings to avoid unnecessary redundancy and to keep the input space more compact.

To preserve the inherent continuous periodicity of these variables, each time feature t with a maximum period T (e.g., $T = 24$ for the hour of the day) was transformed into two distinct cyclical covariates. By projecting the features onto a unit circle using sine and cosine functions, we applied the following mapping:

$$x_{\sin} = \sin\left(\frac{2\pi t}{T}\right), \quad x_{\cos} = \cos\left(\frac{2\pi t}{T}\right) \tag{4.2}$$

This dual representation eliminates the artificial discontinuities that occur at the boundaries of cyclical periods (such as the sharp transition from hour 23 to 0, or from December to January). Consequently, it allows the models to effectively learn temporal dependencies and smooth cyclic patterns across different time scales.

4.3. Experimental setup

In this section, we provide a detailed description of the experimental setup used to evaluate the performance of the proposed models. We describe the framework for the experiments, the evaluation metrics employed, the procedures for hyperparameter tuning, and the computational resources utilized.

4.3.1. Framework

The experiments were conducted using Darts [23], a comprehensive Python library for time series forecasting that provides a wide range of models and tools for data manipulation, model training, and evaluation. Darts offers a unified interface for various forecasting models, including both classical statistical methods and modern machine learning approaches, while also supporting basic conformal prediction forecasts, making it an ideal choice for our experiments.

We implemented custom models using PyTorch [24], a widely-used deep learning framework that allows for flexible model design and efficient training. The combination of Darts and PyTorch enabled us to leverage the strengths of both libraries, facilitating the development and evaluation of our proposed models in a robust and efficient manner.

After the model was implemented in PyTorch, we wrapped it in a Darts interface to ensure compatibility with the Darts framework and to take advantage of its functionalities for training and evaluation. This integration allowed us to seamlessly incorporate our custom models into the

Darts ecosystem, enabling us to utilize its tools for data handling, model training, and performance assessment.

Regarding the MLOps pipeline, we adopted Weights & Biases [25] for experiment tracking and management. We used it to log hyperparameters and models, training metrics, and evaluation results for each experiment, allowing us to systematically compare different model configurations and to keep a comprehensive record of our experimental process. This approach facilitated the analysis of results and the identification of optimal model settings based on empirical evidence.

We also set up a Data Version Control (DVC) [26] data storage to manage our datasets and to ensure reproducibility of our experiments. By versioning our data and tracking changes, we can maintain a clear history of our datasets and ensure that our experiments can be replicated accurately in the future.

4.3.2. Metrics

To evaluate the performance of our models, we employed a comprehensive set of metrics that capture different aspects of forecasting accuracy and uncertainty quantification. Since we are dealing with probabilistic forecasts, we focused on metrics that assess quality of the predicted distributions.

- **(M)IWS** (Mean Interval Winkler Score): evaluates the quality of prediction intervals by considering both the width of the interval and whether the actual value falls within it, providing a balanced measure of interval forecasts. It strongly penalizes intervals that do not contain the true value, while also rewarding narrower intervals that do contain it.

$$\text{IWS} = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} \alpha \cdot (u_i - l_i) + 2 \cdot (l_i - y_i) \text{ if } y_i < l_i \\ \alpha \cdot (u_i - l_i) \text{ if } l_i \leq y_i \leq u_i \\ \alpha \cdot (u_i - l_i) + 2 \cdot (y_i - u_i) \text{ if } y_i > u_i \end{cases} \tag{4.3}$$

Where u_i and l_i are the upper and lower bounds of the prediction interval for the i-th forecast, y_i is the actual value, and α is the significance

level corresponding to the prediction interval (e.g., $\alpha = 0.1$ for a 90% prediction interval).

- **MIW** (Mean Interval Width): measures the average width of the prediction intervals, providing insight into the sharpness of the probabilistic forecasts. Narrower intervals indicate more confident predictions, while wider intervals suggest greater uncertainty.

$$\text{MIW} = \frac{1}{N} \sum_{i=1}^{N} (u_i - l_i) \tag{4.4}$$

- **MIC** (Mean Interval Coverage): which measures the proportion of actual values that fall within the prediction intervals, providing a measure of the reliability of the probabilistic forecasts. A well-calibrated model should have a coverage close to the nominal confidence level (e.g. 0.90 for a 90% prediction interval).

$$\text{MIC} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\{l_i \leq y_i \leq u_i\} \tag{4.5}$$

Those metrics together provide a comprehensive evaluation of both the accuracy of point forecasts and the quality of probabilistic forecasts. The rule of thumb for interpreting these probabilistic metrics is to look for coverage (**MIC**) close to the nominal confidence level, while also considering the width of the intervals (**MIW**), while **IWS** should be as low as possible, indicating better probabilistic forecasts, meaning that the predicted distributions are well-aligned with the observed values and that the prediction intervals are appropriately calibrated and sharp.

4.3.3. Loss function

For training our inner models (TiDE, iTransformer, TSMixer), we used a smooth version of the pinball loss for quantile regression, which is defined as follows:

$$\mathcal{L}_{\text{Pinball}} = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} \tau \cdot (\hat{y}_i - y_i) \text{ if } y_i < \hat{y}_i \\ (1-\tau) \cdot (y_i - \hat{y}_i) \text{ if } y_i \geq \hat{y}_i \end{cases} \tag{4.6}$$

Where $\tau \in (0, 1.0)$ is the quantile level we are considering.

The smooth version is called Huber Pinball Loss [27] and it combines the properties of the traditional pinball loss with the robustness of the Huber loss, providing a more stable training process while still effectively optimizing for quantile regression. The Huber Pinball Loss is defined as follows:

$$\mathcal{L}_{\text{Hub}} = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} \tau z_i - \frac{\tau^2 \delta}{2} & \text{if } z_i \geq \tau\delta \\ \frac{1}{2\delta} z_i^2 & \text{if} -(1-\tau)\delta < z_i < \tau\delta \\ -(1-\tau) z_i - \frac{(1-\tau)^2 \delta}{2} & \text{if } z_i \leq -(1-\tau)\delta \end{cases} \tag{4.7}$$

Where $z_i = y_i - \hat{y}_i$ is the residual error and δ is a hyperparameter that controls the transition point between the linear and quadratic regions of the loss function. We fixed δ to $5 \cdot 10^{-4}$ in our experiments, given the scaled target variable. It is important to note that using a quadratic surrogate for Quantile Regression leads to convergence to the optimal expectiles (squared quantiles), rather than the optimal quantiles. In this context it is used to smoothen the optimization landscape and to provide a more stable training process when errors approach zero.

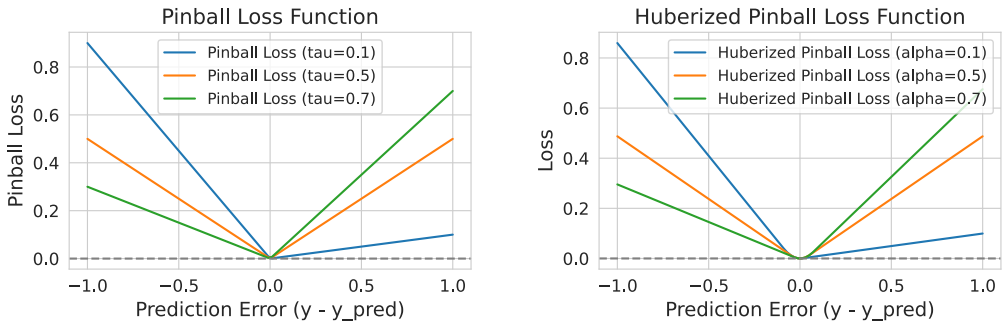


Figure 13: Visualization of the Pinball Loss (left) and the Huber Pinball Loss (right) for different values of τ. The Huber Pinball Loss provides a smoother optimization landscape, particularly for small errors, while still effectively optimizing for quantile regression.

Chapter 5
Experimental Pipeline

In this section, we detail the experimental design of our study. The evaluation framework is structured to bridge high-performance probabilistic forecasting models with rigorous uncertainty quantification techniques, specifically focusing on the integration of Reservoir Conformal Prediction (ResCP) with state-of-the-art forecasting architectures.

To ensure methodological rigor, we strictly decoupled the training of the underlying deep learning models from the tuning of the conformal calibration wrapper. The experimental strategy is organized into four distinct phases: an exploratory optimization phase on a representative subset of datasets, an exploratory grid search to map the behavior of the ResCP extensions, a final global validation phase on the entire benchmark, and an independent evaluation of the post-hoc risk controller introduced in Section 3.5 on two targeted datasets.

5.1. Phase 1: Base Model Optimization (Bayesian Search)

The first phase establishes the optimal configurations for the point forecasters (inner models). Because deep learning architectures possess massive, continuous hyperparameter spaces, exhaustive search methods are computationally prohibitive. Consequently, we utilized Bayesian Optimization, which builds a probabilistic surrogate model of the objective function to efficiently balance exploration and exploitation, converging on near-optimal configurations in a reduced number of iterations. Specifically, for each model architecture, we conducted 20 iterations of Bayesian optimization, which is sufficient to explore the defined hyperparameter

space while maintaining computational feasibility and ensuring a competitive performance baseline for the subsequent calibration phases.

To ensure generalizability and avoid dataset-specific overfitting, this exploratory search was conducted on a curated subset of 6 representative datasets spanning different functional domains and volatility profiles. The objective was to maximize the predictive performance of the three selected "base" architectures: TiDE, iTransformer, and TSMixer.

During this phase, the models were trained independently of any conformal adjustments, minimizing the Huber Pinball Loss on the validation set. Upon completion of the sweeps, the single best-performing configuration for each architecture was identified, and the model weights were frozen to serve as the fixed computational base for the subsequent calibration phases.

The hyperparameters search spaces shared across all models are summarized in Table 3, while the specific hyperparameters for each architecture are detailed in Tables Table 4, Table 5, and Table 6.

Common Parameters	
Parameter	**Value**
Output Chunk Length	24
Input Chunk Length	[96, 168, 336]
Dropout	$\sim \mathcal{U}(0, 0.3)$
Quantiles	(0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95)
Huber Pinball Loss δ	$5 \cdot 10^{-4}$
Batch Size	128
Num Epochs	50
Early Stopping Patience	10

Table 3: Common hyperparameter configuration for the base models.

Note: The output chunk length is fixed to 24 for all models. A 24-hour horizon is standard in day-ahead energy forecasting, aligning the

benchmark with practical industry requirements; fixing it also avoids running the computationally expensive calibration phase separately for each possible horizon length, keeping the experimental pipeline tractable given the size of the benchmark and the number of configurations tested.

TiDE	
Parameter	**Value**
hidden_size	[64, 128, 256]
num_encoder_layers	[1, 2, 3]
num_decoder_layers	[1, 2]

Table 4: Hyperparameter configuration for TiDE

iTransformer	
Parameter	**Value**
d_model	[64, 128, 256]
num_heads	[2, 4]
num_layers	[2, 3]
d_ff	[128, 256, 512]

Table 5: Hyperparameter configuration for iTransformer

TSMixer	
Parameter	**Value**
hidden_size	[64, 128]
ff_size	[64, 128, 256]
num_blocks	[1, 2, 3]

Table 6: Hyperparameter configuration for TSMixer

5.2. Phase 2: Exploratory Calibration (Grid Search)

In the second phase, the ResCP framework was wrapped around the frozen base models. At this stage, the objective shifted from minimizing standard prediction error to optimizing the reliability and sharpness of the predictive intervals.

Unlike Phase 1, we opted for an **exhaustive Grid Search** rather than a Bayesian approach. This methodological shift is motivated by two critical factors:

1. **Preventing Data Leakage:** Using a surrogate loss function to actively optimize calibration parameters risks introducing data leakage—essentially "training" the calibration parameters on the same statistics that are supposed to guarantee theoretical coverage. A grid search allows us to evaluate the impact of each extension deterministically and independently, preserving the statistical integrity of the conformal framework.
2. **Mapping Structural Extensions:** The proposed extensions (Topological Initialization, Similarity Functions, and Temporal Decay profiles) introduce discrete structural choices that are best analyzed through a systematic evaluation of all possible combinations, rather than relying on an optimizer that might skip suboptimal but highly informative regions of the hyperparameter space.

In this phase, we systematically compare the trade-offs between interval width (IWS) and empirical coverage (MIC) to identify the most stable configurations relative to the $1 - \alpha$ confidence target.

ResCP Hyperparameters	
Parameter	**Value**
reservoir_size	1000
spectral_radius	0.9
leak_rate	0.7
window_size	500
adaptive_min_ess	10
symmetric	false
reservoir_type	[random, orthogonal, scr]
similarity_metric	[gaussian, laplacian, cosine]
gamma	$[10^{-1}, 1.0, 3.0]$
decay_type	[power, exponential]
lambda	[0.0, 0.5, 2.0]
adaptive_min_ess_ratio	[0.1, 0.5, 0.9]

Table 7: Hyperparameter configuration for ResCP grid search.

5.2.1. Conformalized Quantile Regression Baseline

In parallel to the ResCP grid search, we evaluated classical Conformalized Quantile Regression (CQR) to establish a rigorous and competitive baseline for comparison. Since standard CQR relies on a uniform evaluation of calibration residuals and does not possess tunable hyperparameters (other than the *symmetric* flag, which we set to `false` to ensure a fair comparison with ResCP setup), this phase reduces to a single, deterministic evaluation for each optimized base model.

5.3. Phase 3: Global Validation

The final phase of the experimental pipeline is designed to definitively validate the robustness of the proposed methodology. We selected the top-performing ResCP configurations—those that demonstrated the best average performance across the 6 guiding datasets during the exploratory

grid search (Phase 2)–and deployed them across the entirety of our benchmark.

By subjecting these best performing configurations to an extended set of datasets, we can rigorously assess the generalizability of the ResCP framework and its extensions, ensuring that the observed performance improvements are not artifacts of specific data characteristics but rather indicative of a broadly applicable enhancement to time series uncertainty quantification.

The exhaustive hyperparameter grids utilized for the base models are detailed in Appendix A.1.4 while the specific ResCP configurations selected for this final validation phase are reported in Table 10.

5.4. Phase 4: Post-Hoc Risk Controller Evaluation

This phase constitutes an independent evaluation track for the post-hoc risk controller described in Section 3.5. Unlike the preceding phases, which assess calibration quality through standard probabilistic metrics (IWS, MIC, MIW), this phase evaluates the controller's ability to satisfy predefined business targets in a streaming, non-stationary setting.

Dataset Selection

Rather than evaluating across the full benchmark, we deliberately selected two datasets that exhibit qualitatively distinct failure modes in the best-performing base conformal predictor from Phase 3:

- **`bdg2_Education_Fox`**: presents mild but systematic upper undercoverage, serving as a low-severity stress test of the controller's ability to correct persistent bias.
- **`zurich_ne5`**: contains a severe concept drift episode induced by the COVID-19 pandemic, serving as a high-severity stress test of the controller's adaptive capacity under prolonged distributional shift.

In both cases, the base model is the best-performing `iTransformer` + ResCP configuration identified in Phase 3, ensuring the controller is evaluated on top of a strong predictive foundation.

Hyperparameter Search

We conducted an exhaustive $3 \times 3 \times 3$ grid search over the controller hyperparameters γ (decay), η (learning rate), and β (LR asymmetry factor), as reported in Table 8. The search objective is to minimize the business target metric $\mathcal{M}$ on the test set relative to the uncontrolled baseline.

Hyperparameter	**Values**
Decay (γ)	0.9, 0.95, 1.0 (no decay)
Learning rate (η)	0.1, 0.3, 0.5
LR asymmetry (β)	0.25, 0.5, 1.0 (no asymmetry)

Table 8: Hyperparameter grid search space for the post-hoc risk controller.

Evaluation Criterion

The target threshold is set to $0.9 \times \mathcal{M}_{\text{base}}$, i.e. a 10% relative improvement over the uncontrolled ResCP baseline. This relative formulation avoids dependence on arbitrary absolute cost thresholds and allows fair comparison across datasets with different scales. Configurations that satisfy the target are then ranked by IWS to identify the best trade-off between business compliance and interval quality.

Chapter 6
Results

In this chapter, we present the empirical outcomes of our experimental pipeline. The analysis is organized into three sections: first, we discuss the findings from the exploratory grid search (Phases 1 and 2) conducted on a representative subset of datasets; second, we present the global validation (Phase 3) of the best-performing configurations across the entire benchmark; finally, we evaluate the post-hoc risk controller (Phase 4) on two targeted datasets selected for their qualitatively distinct failure modes.

6.1. Exploratory Calibration and Grid Search Results

To identify the optimal architectural configurations and evaluate the impact of the proposed extensions, an extensive grid search was performed on the guiding datasets. Table 9 summarizes the performance of the Reservoir Conformal Prediction (ResCP) framework compared to the uncalibrated Base models and the standard Conformalized Quantile Regression (CQR) baseline.

Dataset	Method	TiDE			iTransformer			TSMixer		
		Δ Cov	MIW	IWS	Δ Cov	MIW	IWS	Δ Cov	MIW	IWS
Education_Bear	Base	−1.6%	78.19	116.94	−5.3%	61.22	110.41	−14.0%	64.36	134.23
	ResCP	−0.2%	76.39	**110.26**	−3.0%	65.59	**105.49**	−6.4%	67.52	115.59
	CQR	−1.0%	77.90	112.89	−0.1%	73.26	107.14	−2.6%	79.28	**114.91**
Healthcare_Eagle	Base	−1.7%	121.76	200.33	−16.4%	77.10	191.92	−18.0%	91.36	257.74
	ResCP	−1.4%	108.50	**180.88**	−4.4%	83.49	**155.30**	−2.6%	106.72	199.92
	CQR	−1.2%	112.07	183.42	−2.8%	96.89	158.84	−4.1%	114.79	**192.54**
Industry_Hog	Base	−2.7%	52.48	77.10	−4.9%	49.60	78.75	−20.3%	55.69	137.83
	ResCP	+6.3%	66.92	74.97	+5.7%	72.17	79.70	+3.9%	71.08	79.99
	CQR	+0.6%	55.12	**71.39**	+1.2%	55.21	**70.11**	+0.1%	56.97	**73.53**
Office_Eagle	Base	−5.7%	31.18	55.66	−4.1%	26.36	46.28	−6.4%	28.24	62.01
	ResCP	−0.7%	34.54	**50.70**	−1.3%	28.91	**44.34**	−0.7%	34.94	**50.74**
	CQR	−1.1%	34.55	52.89	−2.4%	28.86	47.07	−2.5%	35.08	52.80
solar_poli	Base	−9.4%	131.59	264.39	+0.1%	163.17	226.35	+3.1%	153.52	220.24
	ResCP	+3.2%	169.28	**248.95**	+3.9%	175.56	**222.55**	+4.5%	163.98	**216.51**
	CQR	−0.8%	144.97	281.89	+0.8%	160.10	236.74	+0.1%	148.40	234.40
zurich_ne5	Base	−24.4%	9007.57	40912.00	−25.3%	7467.43	34394.40	−4.9%	12693.55	21072.72
	ResCP	+0.3%	15435.38	**24172.54**	−2.8%	13385.17	**22848.93**	+0.6%	14036.45	**18991.58**
	CQR	−3.0%	14684.12	26003.96	−2.7%	14500.38	24454.00	−2.7%	13579.16	20471.91

Table 9: Quantitative comparison of forecasting performance across the considered datasets. The Δ Cov. column indicates the deviation from the 90% target nominal coverage, color-coded for reliability: green for acceptable bounds ($\Delta \geq -2.5\%$), orange for moderate under-coverage ($[-5\%, -2.5\%)$), and red for severe under-coverage ($< -5\%$). The lowest (best) IWS for each architecture is highlighted in **bold**, while the absolute best IWS across all models for a given dataset is additionally marked in **blue**.

The results in Table 9 show that the proposed ResCP extension consistently improves calibration across all datasets and architectures, reducing the deviation from the nominal 90% coverage target. In particular, the method effectively mitigates severe under-coverage observed in both the Base and CQR models, especially in highly non-stationary scenarios.

Importantly, these improvements are achieved while maintaining competitive interval sharpness. Although in some cases improved calibration results in slightly wider intervals, the overall trade-off between reliability and sharpness remains favorable, as reflected by the IWS metric.

Furthermore, the observed gains are consistent across different forecasting architectures, indicating that the proposed calibration strategy is model-agnostic and robust to different temporal representations.

A representative example is the `zurich_ne5` dataset, where both Base and CQR exhibit severe under-coverage, while ResCP brings the coverage close to the nominal level and significantly improves the IWS metric. This highlights the effectiveness of the proposed method in highly non-stationary environments.

6.1.1. Hyperparameter Dynamics: Insights

To dissect the mechanics of the framework, we utilize the `bdg2_Office_Eagle` dataset as a representative scenario of structured energy consumption. In this regime, the iTransformer base model already captures the primary trends, allowing us to isolate how ResCP refines the residual uncertainty, as illustrated by the Pareto front analysis in Figure 14.

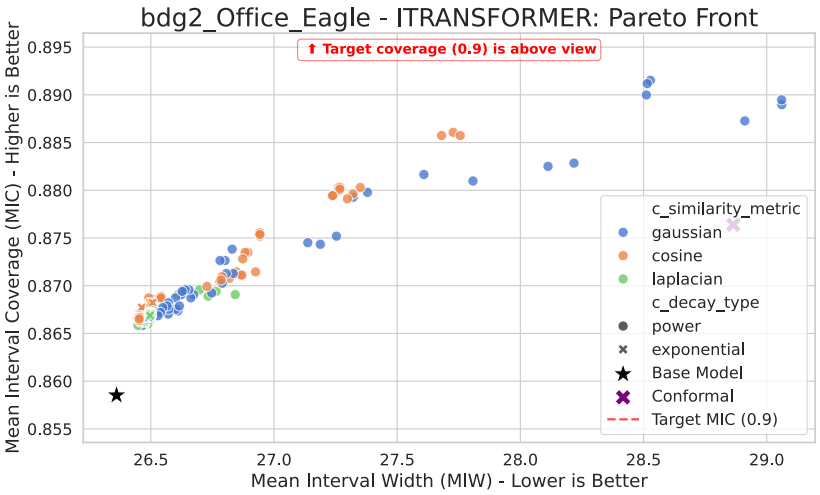


Figure 14: Pareto Front for Office Eagle. ResCP configurations (Gaussian and Cosine) successfully push the boundary toward the 0.90 target coverage, effectively bridging the gap left by the base model.

The Stability of Adaptive Guardrails

The analysis of the Effective Sample Size (ESS) ratio, shown in Figure 15, demonstrates that the system is intrinsically stable. The optimal performance consistently clusters at an `adaptive_target_ess_ratio` of 0.1. This suggests that the topological weighting is robust enough to provide reliable intervals without frequently triggering the uniform CQR fallback mechanism. The framework "trusts" its internal memory state to define the calibration window.

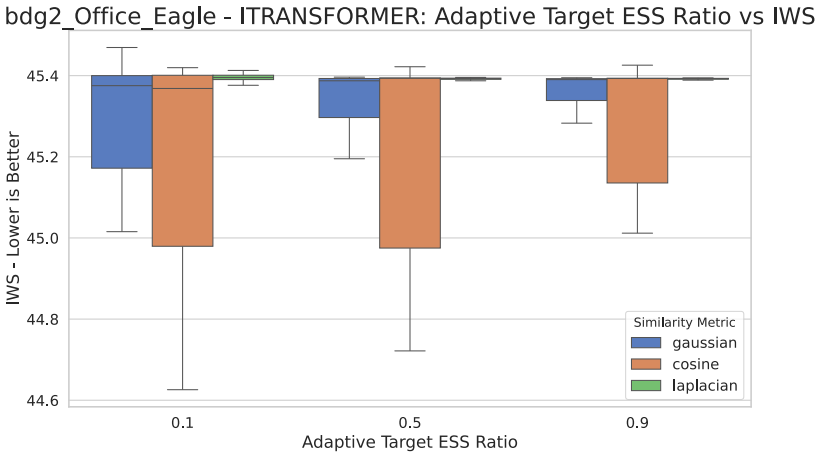


Figure 15: IWS vs. Adaptive Target ESS Ratio. The 0.1 ratio provides the best balance between adaptability and stability.

Metric Robustness and Topology Sensitivity

A key finding is the divergent behavior between similarity metrics regarding reservoir initialization (Figure 16). The **Gaussian** metric exhibits high structural robustness, showing minimal variance across different initializations (Random, Orthogonal, SCR). Conversely, the **Cosine** metric is significantly more sensitive to the reservoir topology; while it can achieve superior peak performance (lowest IWS), its efficacy is more tightly coupled with the specific memory structure of the reservoir. It is to be noted that this behaviour changes across datasets, with some scenarios where the **Cosine** metric is less sensitive to the topology than the **Gaussian** one. This suggests that the optimal metric-topology pairing may be dataset-dependent, potentially influenced by the underlying temporal dynamics and noise characteristics of the series.

Importantly, the parity between deterministic topologies (SCR) and random ones (Orthogonal/Random sparse) allows for a fully reproducible setup without performance degradation.

The **Laplacian** metric serves as a negative baseline, performing consistently worse than the alternatives across all datasets. This suggests that its long tails reduce its efficacy as a similarity score, failing to provide the sharp discriminative power that characterizes the other metrics.

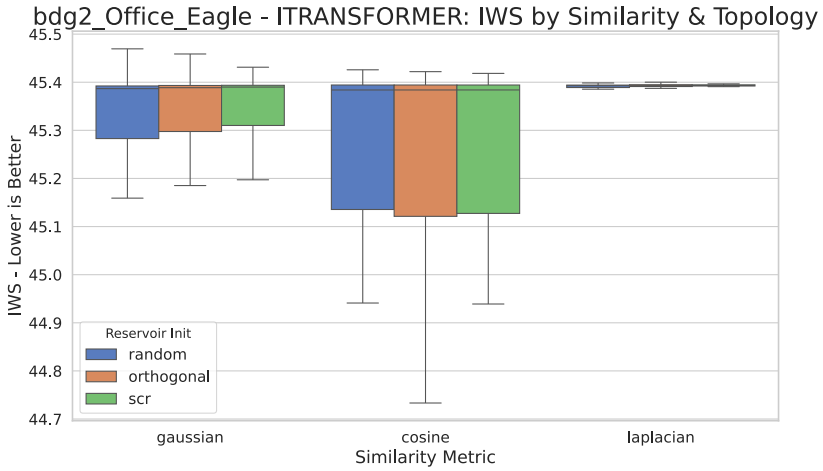


Figure 16: IWS by Similarity and Topology. Note the robustness of Gaussian vs. the sensitivity of Cosine initialization.

An Empirical Rule of Temperature Scaling

The sensitivity analysis on the temperature parameter (γ), presented in Figure 17, revealed an empirical law that remains constant across all tested datasets. We observe a strict divergence:

- **Gaussian Similarity** optimizes at **high** temperatures ($\gamma \approx 3.0$), which softens the spatial penalty to aggregate a broader, more robust neighborhood of states.
- **Cosine Similarity** optimizes at **low** temperatures ($\gamma \approx 0.1$), amplifying angular differences to isolate only the most hyper-relevant directional states.

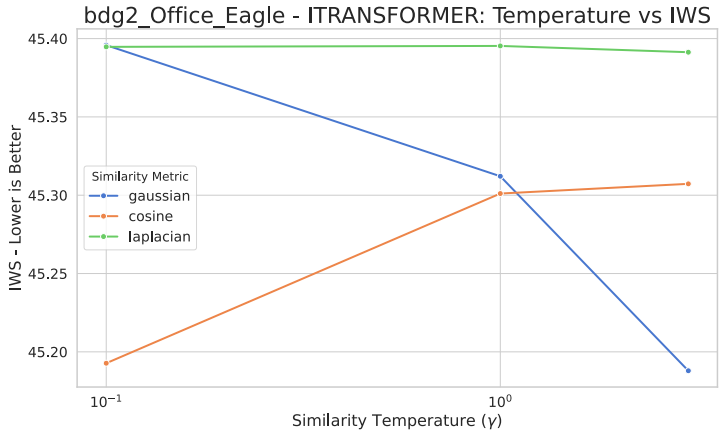


Figure 17: Empirical Temperature Behaviour: Gaussian requires higher γ while Cosine demands lower values for optimal IWS.

Temporal Decay: The Need for an Anchor

Contrary to a pure ablation scenario, the results confirm that a mild chronological anchor is necessary (see Figure 18). A decay rate of $\lambda = 0.5$ consistently outperforms the "no-decay" ($\lambda = 0.0$) configuration. This suggests that while the reservoir's spatial mapping is the primary driver of similarity, a gentle preference for recent history helps the model track subtle concept drifts more effectively than a purely topological approach. A practical visualization of this adaptive capability on the eagle dataset can be seen in Figure 19, where bounds correctly adapt to local volatility.

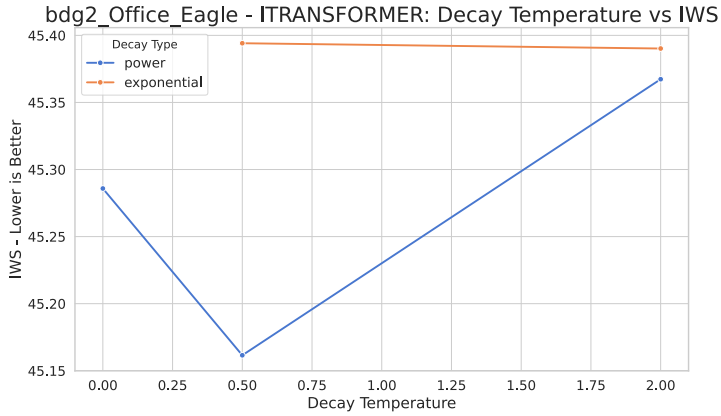


Figure 18: Impact of Decay Temperature. The $\lambda = 0.5$ setting provides the global optimum across the power decay type.

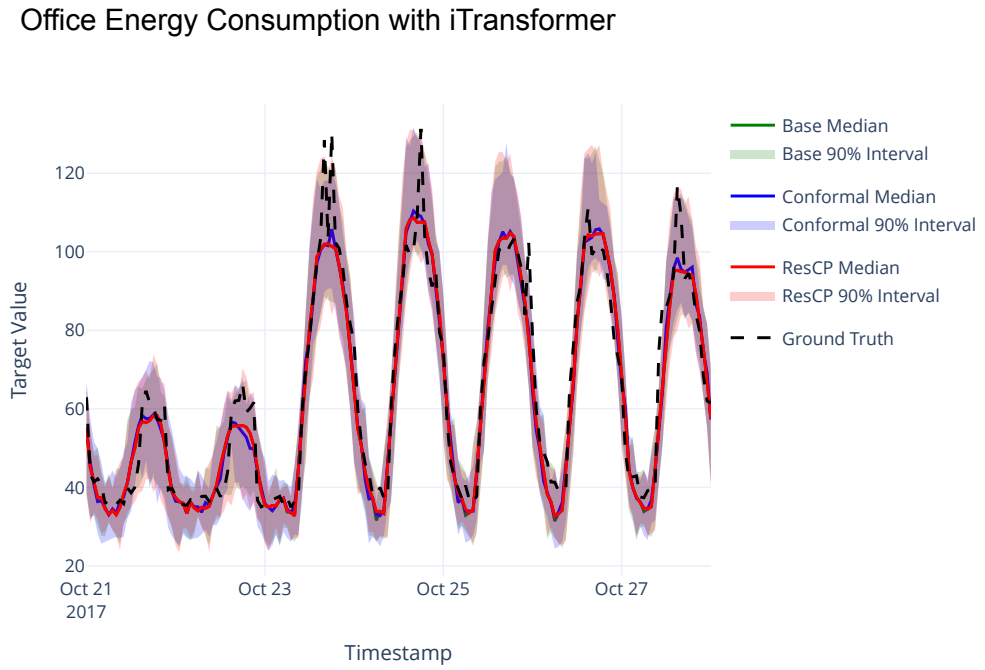


Figure 19: Visual comparison of probabilistic forecasts on the bdg2_Office_Eagle dataset using the iTransformer architecture. The graph demonstrates how the ResCP bounds (red) dynamically adapt to the local variance of the series.

6.1.2. The Solar Challenge: The IWS-MIW Paradox

The `solar_poli` dataset highlights the true adaptive value of ResCP in high-volatility regimes. In Table 9, one might notice that CQR yields a narrower average width (MIW) than ResCP for Solar. However, ResCP achieves a vastly superior Winkler Score (248.95 vs 281.89).

This paradox, illustrated in Figure 20, is explained by the heteroscedastic nature of solar production (predictable nights vs. erratic cloudy days). CQR applies a static, compromised interval that is too wide at night and dangerously narrow during volatile days. ResCP dynamically expands its

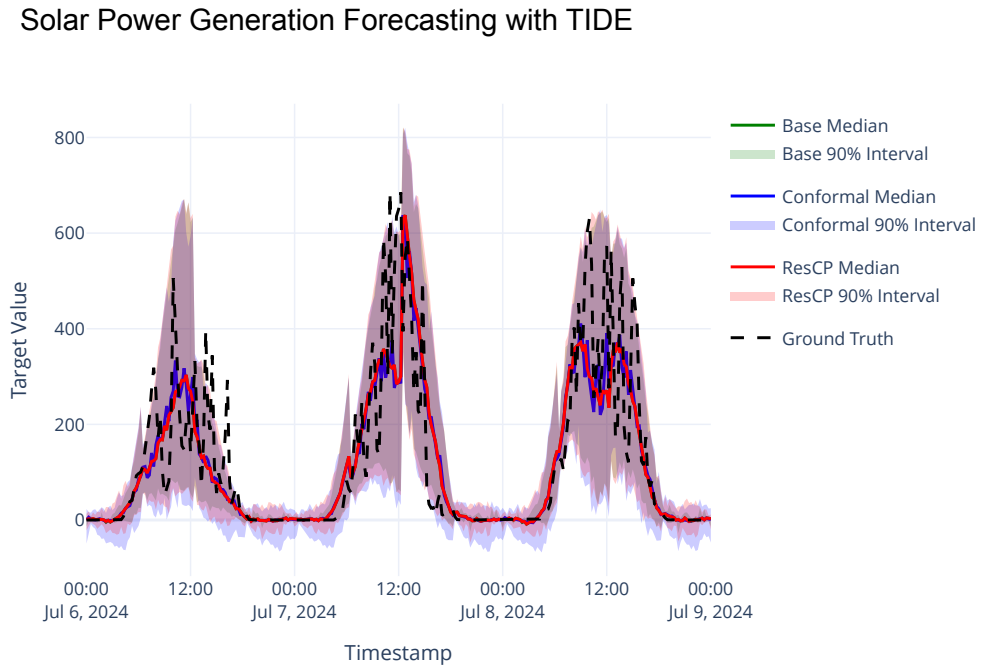


Figure 20: Illustration of the "Solar Paradox" on the solar_poli dataset (using the TiDE base model). While the uniform CQR approach applies a compromised static interval—dangerously wide during predictable nights and too narrow during volatile days—ResCP successfully collapses its uncertainty bounds to zero at night and expands them correctly during the highly variable daytime peaks.

intervals during high-variance states, minimizing the heavy geometric penalties of the Winkler Score when bounds are violated.

6.1.3. Robustness to Concept Drift: The COVID-19 Anomaly

To further validate the adaptability of the ResCP framework, we examined its behavior under conditions of extreme concept drift. The `zurich_ne5` dataset, which records residential electricity consumption in

the city of Zurich, provides a unique natural experiment: the test set encompasses the Spring 2020 COVID-19 lockdown period.

During this timeframe, sudden shifts in human mobility and the rapid adoption of remote working policies caused a massive structural break in standard energy consumption patterns. Traditional probabilistic forecasting models, along with standard Conformalized Quantile Regression (CQR), fundamentally struggle in these scenarios. Their uniform calibration residuals are based on pre-pandemic or transitional data distributions, making them slow to adapt to the new reality.

As illustrated in Figure 21, which details a specific anomaly window in April 2020, the standard conformal intervals (blue) fail to react efficiently to the sudden drops and altered volatility of the ground truth load. Because

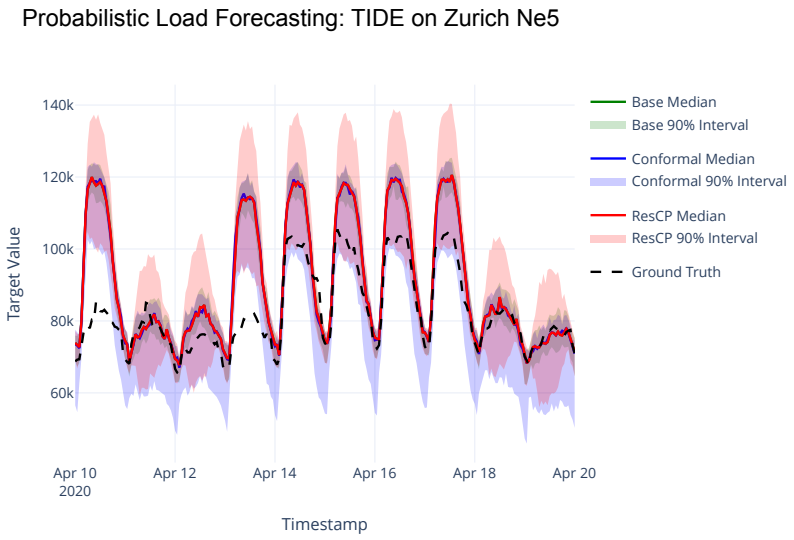


Figure 21: Impact of extreme concept drift on the `zurich_ne5` dataset during the April 2020 COVID-19 lockdown. Despite the anomalous and sudden structural drop in electricity load, the ResCP framework demonstrates robustness by rapidly adjusting its confidence intervals, maintaining valid marginal coverage where traditional static methods might fail

CQR assigns equal weight to all past calibration errors, its intervals remain rigidly anchored to an outdated regime.

In contrast, ResCP (red) leverages its topological memory and temporal decay mechanisms to rapidly discard outdated residual states. By heavily weighting the most recent and topologically relevant conditions, the ResCP bounds dynamically adapt to the anomalous load profile. The framework swiftly shifts its intervals downward and adjusts its width to wrap the new, lower-variance profile, maintaining valid marginal coverage without artificially inflating the Mean Interval Width (MIW). This real-world stress test demonstrates the framework's intrinsic resilience to sudden, severe exogenous shocks.

6.1.4. Architectural Selection for Global Validation

Finally, the grid search reveals a distinct performance hierarchy among the base models. The **TiDE** architecture (a purely dense MLP) consistently underperforms, showing a limited capacity to resolve fine-grained, short-term temporal variations compared to its counterparts. TiDE was retained in the global validation phase as a lower-bound reference, given its comparatively limited capacity in the provided settings. The **iTransformer** and **TSMixer** architectures, which incorporate more sophisticated temporal modeling capabilities, achieve significantly better performance. However, the iTransformer's self-attention mechanism appears to provide a more effective representation of temporal dependencies in the context of ResCP's topological calibration, leading to superior overall results.

6.1.5. Parameter Robustness and Elite Configurations

A fundamental outcome of the exploratory grid search is the observed parameter robustness.

As detailed in Appendix Appendix A.1.2, the optimal hyperparameter configurations exhibited a striking convergence across all base architectures and datasets. Rather than requiring exhaustive, domain-specific

Similarity (γ)	**Topology**
Gaussian ($\gamma = 3.0$)	Random, Orthogonal, SCR
Cosine ($\gamma = 0.1$)	Random, Orthogonal, SCR

Table 10: Reduced search space of elite configurations for the global validation phase. The final 6 configurations evaluated correspond to the permutations of the selected similarity profiles and reservoir topologies.

tuning, the ResCP framework naturally collapsed into a highly restricted sub-space of high-performing configurations.

In Table 10 we report the drastically reduced search space of elite configurations that consistently emerged as top performers across the guiding datasets. The framework unanimously favors an `adaptive_target_ess_ratio` of 0.1, a `power` temporal decay, and a `decay_temp` of 0.5, highlighting a structural stability that is almost completely agnostic to the underlying base model. This robustness significantly simplifies the deployment of ResCP in real-world scenarios, as practitioners can confidently select from this elite subset without the need for extensive hyperparameter optimization.

6.2. Global Validation Results

In this section, we present the results of the global validation phase, evaluating the best-performing ResCP configurations across the entire benchmark of datasets.

To synthesize the extensive empirical outcomes, Table 11 quantifies the average relative IWS improvement of each calibration method over the uncalibrated Base model, broken down by base architecture. The relative improvement is defined as:

$$\text{Rel. Improvement} = \frac{\text{IWS}_{\text{base}} - \text{IWS}_{\text{method}}}{\text{IWS}_{\text{base}}} \times 100\% \qquad (6.1)$$

Base Model	Method	Mean Rel. IWS Impr.	Std	ResCP < CQR
TiDE	ResCP	+14.96%	11.64%	17/26
	CQR	+13.40%	12.86%	
iTransformer	ResCP	+11.30%	12.66%	16/26
	CQR	+10.39%	15.02%	
TSMixer	ResCP	+20.06%	19.73%	18/26
	CQR	+18.49%	22.40%	
Aggregate	ResCP	+15.44%	15.34%	51/78
	CQR	+14.09%	17.35%	

Table 11: Average relative IWS improvement over the uncalibrated Base model, grouped by base architecture. A positive value indicates a lower (better) IWS. The **ResCP < CQR** column counts datasets on which ResCP directly outperforms CQR.

A positive value indicates a reduction in the Winkler Score — i.e., tighter and more calibrated prediction intervals — compared to the uncalibrated base model. The **ResCP < CQR** column reports the number of datasets on which ResCP achieves a strictly lower IWS than CQR.

As shown in Table 11, ResCP consistently achieves higher average relative IWS improvements than CQR across all three base architectures, with a global average improvement of $+15.44\%$ versus $+14.09\%$ for CQR. The advantage is most pronounced with TSMixer ($+20.06\%$ vs. $+18.49\%$), where the residual-based calibration effectively corrects the stronger non-stationarity exhibited by that architecture. In direct head-to-head comparisons, ResCP achieves a strictly lower IWS than CQR in 51 out of 78 model-dataset pairs, confirming its robustness and generalizability across diverse temporal dynamics.

For the complete and disaggregated quantitative results, including Mean Interval Width (MIW) and marginal coverage deviations (Δ Cov.) for each dataset and architecture, please refer to Appendix A.2.

Overall, the experimental results demonstrate that the proposed ResCP extension provides a robust and adaptive calibration strategy, capable of handling non-stationarity and multi-step forecasting scenarios while maintaining a favorable balance between reliability and sharpness.

6.3. Post-hoc Calibration Experiments

6.3.1. Business-Aware Target Metric

To avoid making assumptions about exact monetary costs, we designed a custom weighted miscoverage metric that captures the asymmetric costs of overestimation and underestimation in the energy load forecasting domain:

$$\mathcal{M}(\boldsymbol{y}, \hat{\boldsymbol{q}}) = \frac{\sum_{t=1}^{T} c_t \cdot \mathbb{1}[y_t > \hat{q}_t^{+}]}{\sum_{t=1}^{T} c_t} \quad \text{(upper)} \tag{6.2}$$

$$\mathcal{M}(\boldsymbol{y}, \hat{\boldsymbol{q}}) = \frac{\sum_{t=1}^{T} c_t \cdot \mathbb{1}[y_t < \hat{q}_t^{-}]}{\sum_{t=1}^{T} c_t} \quad \text{(lower)} \tag{6.3}$$

where $\hat{q}_t$ is the monitored quantile bound at time t, and c_t is a cost weight defined as:

$$c_t = \begin{cases} 4 & \text{if } t \in [12{:}00, 19{:}00] \\ 1 & \text{otherwise} \end{cases} \tag{6.4}$$

This weighting reflects the operational reality of the energy sector: errors during daytime peak hours (12:00–19:00) carry a significantly higher operational cost than nocturnal deviations, as they are more likely to affect grid stability and dispatch decisions. The 4x penalty factor captures this relative cost asymmetry between peak and off-peak periods.

6.3.2. Results

The full grid search space is detailed in Table 8. The best configuration by MIWS satisfying the $0.9 \times \mathcal{M}_{\text{base}}$ target threshold for each dataset

is reported in Table 12, while Figure 22 and Figure 23 provide a visual comparison of the controlled and uncontrolled predictions.

Bdg2 Education Fox

Metric	Not Controlled	Controlled
BFN	0.1688	0.0811
MIC	0.8381	0.9226
MIW	72.5076	84.4036
MIWS	120.9455	112.4913

Zurich NE5

Metric	Not Controlled	Controlled
BFN	0.1274	0.0632
MIC	0.8402	0.8964
MIW	13460.9484	15195.5994
MIWS	25066.3620	21888.2918

Table 12: Comparison between not controlled and risk controlled configurations. BFN is the evaluated business function. In the controlled configuration, MIW increases as the controller widens the prediction interval to meet the risk target; however, the resulting MIWS improvement confirms that the added width is justified by the reduction in high-cost miscoverage events.

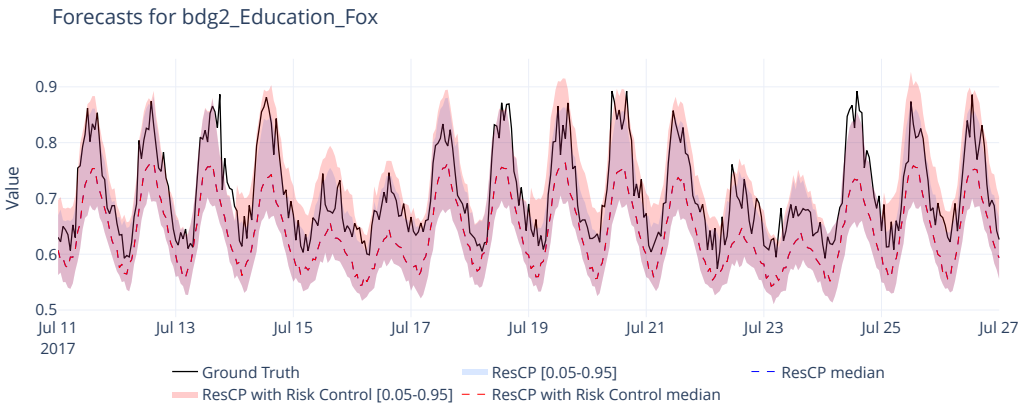


Figure 22: Visual comparison of predictions in `bdg2_Education_Fox` with (red) without (blue) risk control. Risk control acts on the upper bound only as a reaction to recent past miscoverage and fades overtime if the transient phase of undercoverage is passed

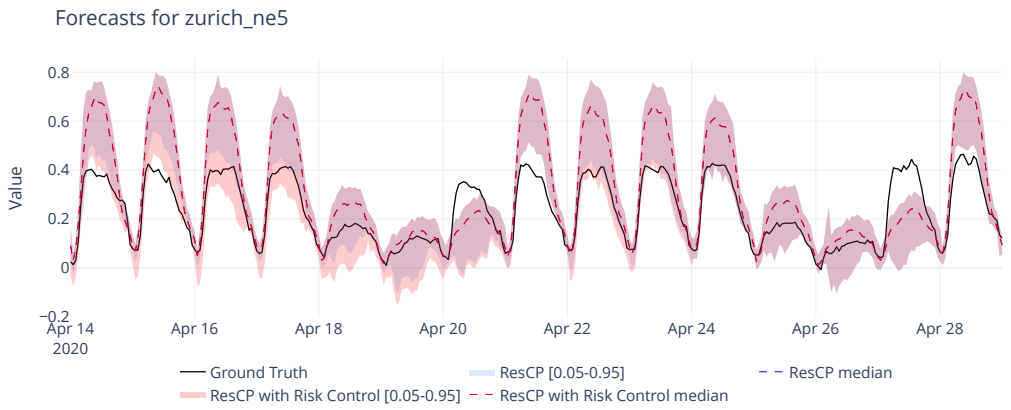


Figure 23: Visual comparison of predictions in `zurich_ne5` with (red) without (blue) risk control. The system experiences a massive concept drift, the risk-controlled approach manages to adjust lower predictions to contain the actual value

6.3.3. Analysis of Results

The grid search results, detailed in Figure 24 and Figure 25, reveal a consistent hierarchy of hyperparameter importance across both datasets. The learning rate (η) is by far the most influential parameter: only low values ($\eta = 0.1$) reliably improve upon the baseline, while higher values ($\eta = 0.3$ or 0.5) introduce excessive reactivity to short-term fluctuations, producing oscillatory adjustments that ultimately degrade MIWS. The decay factor (γ) and the LR asymmetry (β) act as secondary fine-tuning knobs, with limited but non-negligible marginal effects visible in the box plots.

The Pareto front analysis highlights an important trade-off: configurations that achieve high marginal interval coverage (MIC) tend to do so at the cost of wider intervals (MIW), and vice versa. In `bdg2_Education_Fox`, the Pareto frontier is compact and the best achievable MIWS improvement over the baseline is moderate (7%), consistent with the dataset's mild systematic undercoverage. In `zurich_ne5`, the relative improvement is

far more pronounced (13%), reflecting the controller's capacity to meaningfully adapt during the prolonged distributional shift induced by the pandemic.

Taken together, these results confirm that the post-hoc controller is most effective when operated conservatively: a low learning rate ensures that quantile adjustments remain incremental, preserving the distributional coherence of the base model while progressively correcting persistent miscoverage.



Figure 24: Hyperparameter search analysis for `bdg2_Education_Fox`. *Top left:* MIWS heatmaps across the grid, sliced by decay factor; darker cells indicate lower (better) MIWS. *Top right:* Marginal effect of each hyperparameter on MIWS — the learning rate shows the strongest and most consistent influence. *Center:* Pareto front in the MIC–MIW space (maximize coverage, minimize interval width); Pareto-optimal configurations dominate the baseline (red cross, MIWS=120.9) with a best MIWS of 112.49. *Bottom:* Per-run ΔMIWS relative to the baseline, broken down by hyperparameter; only low learning rates (η=0.1) reliably yield improvements, while higher values cause degradation.



Figure 25: Hyperparameter search analysis for `zurich_ne5`. *Top left:* MIWS heatmaps across the grid, sliced by decay factor; *Top right:* Marginal effects confirm the learning rate as the dominant factor, with moderate values (η=0.3) appearing more competitive than in the `bdg2` case. *Center:* Pareto front in the MIC–MIW space; all Pareto-optimal configurations substantially improve upon the baseline (MIWS=25066.4), with a best MIWS of 21888.29, reflecting the controller's ability to adapt under prolonged distributional shift. *Bottom:* ΔMIWS breakdown by hyperparameter; improvements are spread across a wider range of settings compared to `bdg2`, suggesting the controller is more robust when the signal-to-noise ratio of the error is high.

Chapter 7
Conclusions

"The illusion that one understands the past fosters overconfidence in one's ability to predict the future."

— **Daniel Kahneman, *Thinking, Fast and Slow***

In this work, we addressed the problem of uncertainty quantification in non-stationary time series forecasting. While Conformal Prediction provides strong theoretical guarantees, its standard formulations rely on uniform calibration schemes that fail to adapt to evolving data distributions and are not well-suited for multi-step forecasting.

To overcome these limitations, we proposed an extension of the Reservoir Conformal Prediction (ResCP) framework based on topological representations of temporal context. By leveraging Echo State Networks, our approach dynamically weights calibration residuals according to their similarity to the current state, while temporal decay mechanisms allow the model to prioritize recent observations. We further introduced an adaptive blending strategy based on Effective Sample Size (ESS) to ensure stability, together with a safety constraint that prevents the collapse of prediction intervals in low-variance regimes.

A key contribution of this work is a principled, theoretically grounded extension of conformal calibration to multi-step forecasting. Through an asynchronous pending queue mechanism, we resolve the observation lag in a strictly causal manner, capturing the natural accumulation of uncertainty over time. Crucially, we provided rigorous theoretical guarantees for this multi-step architecture using the Non-Exchangeable Conformal Prediction (NexCP) framework. We demonstrated that our dynamic weighting scheme preserves mathematical validity, yielding coverage

error bounds that explicitly scale with the distributional shift across the forecasting horizon.

Experimental results on heterogeneous real-world energy datasets demonstrate that the proposed approach significantly improves the trade-off between reliability and sharpness of prediction intervals, particularly under non-stationary conditions and extreme concept drift.

Beyond the core methodological contributions, we explored the deployment of the proposed framework in risk-sensitive operational contexts. Driven by the operational reality that the costs of overestimation and underestimating demand are fundamentally asymmetric, we proposed an online post-hoc risk calibration layer. Philosophically inspired by Conformal Risk Control (CRC) and adaptive feedback mechanisms, this layer decouples pure statistical learning from business logic by applying a dynamic, additive shift directly onto the predicted quantiles.

Through an asymmetric learning rate and strict structural guardrails that guarantee quantile monotonicity, the controller continuously adjusts the uncertainty bounds in a streaming fashion based on time-weighted business metrics – such as penalizing peak-hour undercoverage more heavily than nocturnal deviations. Empirically evaluated on challenging scenarios, including the extreme structural drift of the COVID-19 pandemic, this mechanism proves capable of swiftly adapting to critical anomalies and tracking predefined business targets without requiring model retraining or violating the underlying distribution's shape.

By demonstrating that the final operational operating point – balancing violation rates, interval sharpness, and economic utility – can be managed as a transparent business decision rather than an algorithmic constraint, this architecture provides a complete, multi-layered solution safely backed by a stable and calibrated predictive foundation.

Crucially, because our causal multi-step mechanism and matrix-based buffering are model-agnostic, they are not strictly limited to Reservoir Computing. Future work could explore the integration of this architec-

tural framework with other state-of-the-art adaptive conformal methods (e.g., attention-based or associative memory models). By decoupling the temporal representation from the multi-step lag resolution, this approach provides a robust and mathematically sound blueprint for sequential uncertainty quantification across diverse domains.

Appendix B provides complementary empirical evidence for the architectural choices made throughout this work. Through a systematic point-forecast evaluation of the three deep learning backbones on the EMS benchmark — including a zero-shot generalisation study on held-out meters belonging to seen building categories — the results show that iTransformer consistently outperforms the XGBoost baseline even without any fine-tuning on the target series, while TSMixer fails to transfer altogether. These findings further substantiate the suitability of iTransformer as a backbone for streaming conformal frameworks deployed across heterogeneous building portfolios, and suggest that its robustness to distribution shift at the meter level may prove particularly valuable in production scenarios where per-meter adaptation is infeasible.

Nevertheless, from a theoretical perspective, the guarantees established in this work are marginal per-step: each horizon h is covered independently at the nominal level, but no joint coverage guarantee is provided across the full output vector $Y_{t+1:t+H}$. Extending conformal calibration to joint coverage over the entire horizon remains an open theoretical challenge. Achieving such guarantees without strong distributional assumptions — such as independence across steps or a known error covariance structure — appears fundamentally difficult, if feasible at all, in the online non-stationary setting, where the dependence structure between residuals at different horizons is unknown and time-varying. We leave this as a direction for future work, noting that the marginal per-step formulation adopted here is not merely a pragmatic compromise, but arguably the most principled choice available under minimal assumptions.

Appendix: Extended Experimental Results,

A.1 Experimental Results (Phase 1 and 2)

This appendix provides the complete set of visual analyses and sensitivity studies conducted during the exploratory grid search (Phase 1 and 2). While Chapter 6 focuses on representative case studies (`Office_Eagle` and `solar_poli`), the following figures ensure full transparency and reproducibility of the results across all guiding datasets and architectures.

A.1.1 Hyperparameter Selection Trends

Before analyzing the coverage and sharpness metrics, we visualize the hyperparameter selection trends resulting from phase 1 and phase 2. For each model (TiDE, iTransformer, TSMixer, ResCP), we identify the most frequently selected hyperparameter values across the 6 guiding datasets. For each configuration we selected the top 10 performing hyperparameter combinations (in terms of IWS) and we report the most frequently selected value for each hyperparameter across those top 10 configurations. This analysis provides insights into which architectural choices and calibration parameters are most consistently associated with optimal performance, guiding the selection of configurations for the subsequent global validation phase.

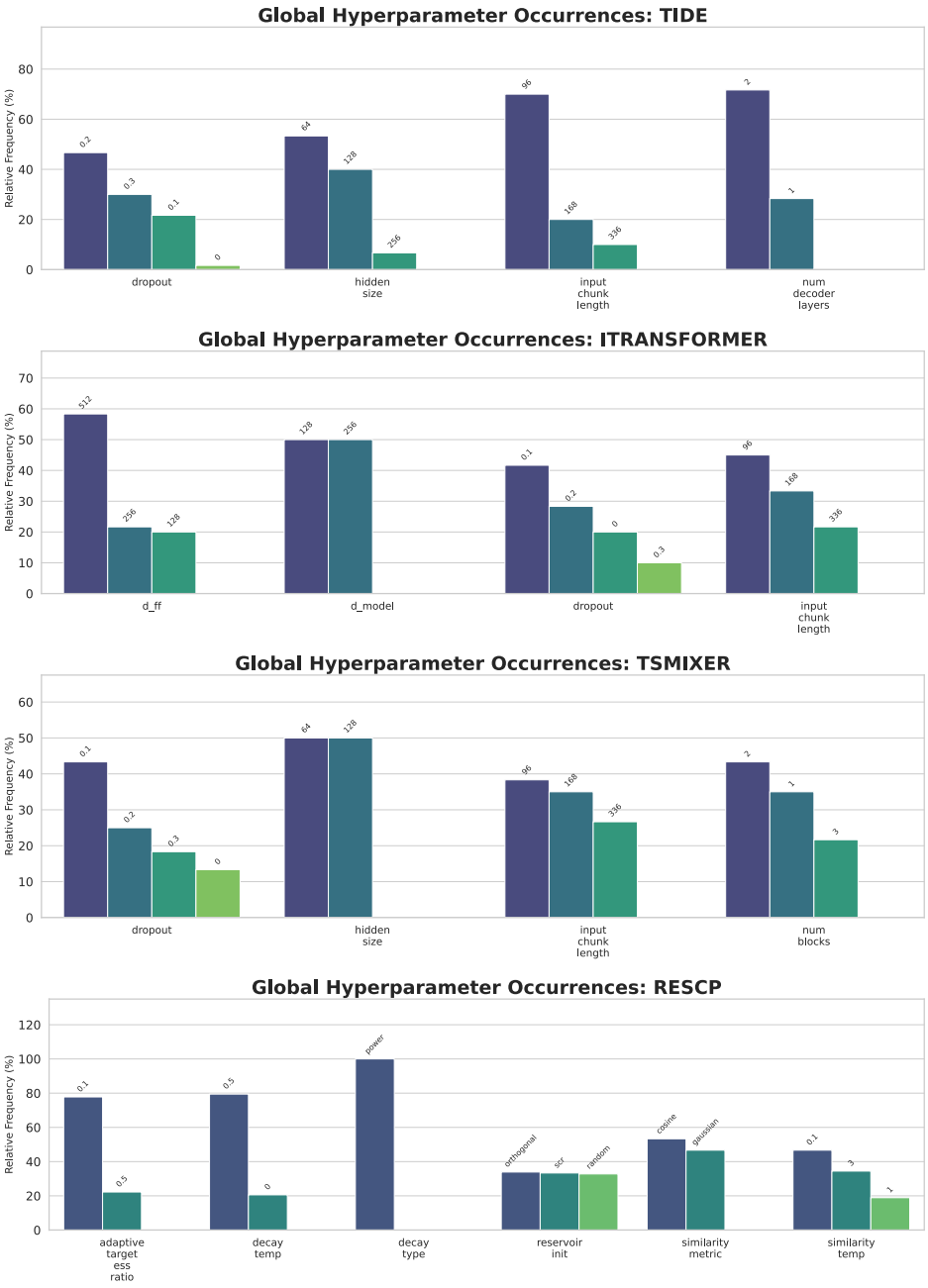


Figure 26: Most frequently selected hyperparameter values across the top 10 performing configurations for each model (TiDE, iTransformer, TSMixer, ResCP) during the grid search phase. This analysis highlights the architectural and calibration choices that are most consistently associated with optimal performance across the guiding datasets.

A.1.2 ResCP parameters

Table 13 reports the optimal ResCP hyperparameter configurations identified during the Phase 2 exploratory grid search across the guiding datasets, detailed for all three base architectures (TiDE, iTransformer, and TSMixer).

Rather than exhibiting high variance across different models and domains, the empirical results demonstrate a striking parameter robustness. The optimal setups consistently collapse into a highly restricted subset of the original search space. Specifically, the framework unanimously favors an `adaptive_target_ess_ratio` of 0.1, a `power` temporal decay, and a `decay_temp` of 0.5, highlighting a structural stability that is almost completely agnostic to the underlying base model.

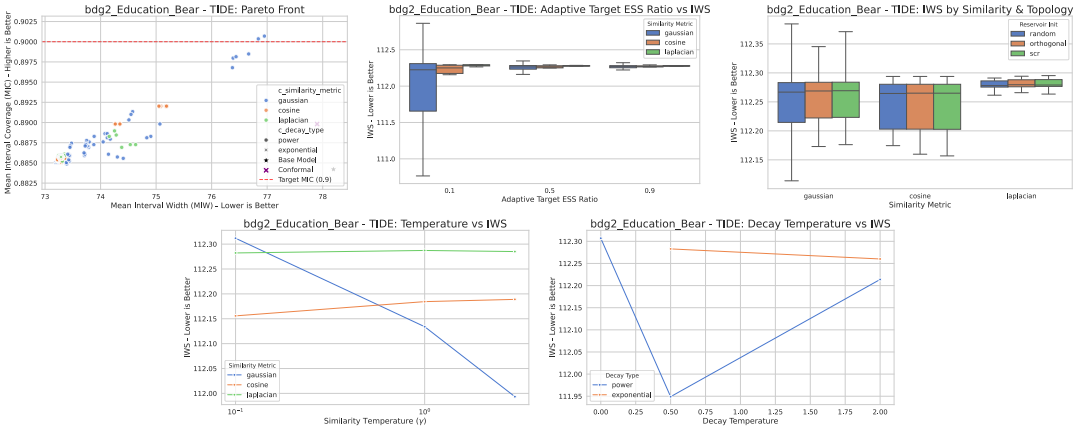
Model	Dataset	Temp	Decay	Decay Type	Res Init	Metric	Target ESS
iTransformer	bdg2_Education_Bear	1.0	0.5	power	random	gaussian	0.1
	bdg2_Healthcare_Eagle	3.0	0.5	power	random	gaussian	0.1
	bdg2_Industry_Hog_idx0	3.0	0.5	power	orthogonal	gaussian	0.1
	bdg2_Office_Eagle	3.0	0.0	power	random	gaussian	0.1
	solar_poli	0.1	0.5	power	random	cosine	0.1
	zurich_ne5	0.1	0.5	power	scr	cosine	0.1
TiDE	bdg2_Education_Bear	3.0	0.5	power	random	gaussian	0.1
	bdg2_Healthcare_Eagle	3.0	0.5	power	random	gaussian	0.1
	bdg2_Industry_Hog_idx0	1.0	0.5	power	random	gaussian	0.1
	bdg2_Office_Eagle	3.0	0.5	power	random	gaussian	0.1
	solar_poli	0.1	0.5	power	random	cosine	0.1
	zurich_ne5	3.0	0.0	power	random	gaussian	0.1
TSMixer	bdg2_Education_Bear	3.0	0.5	power	random	gaussian	0.1
	bdg2_Healthcare_Eagle	3.0	0.5	power	orthogonal	gaussian	0.1
	bdg2_Industry_Hog_idx0	3.0	0.5	power	random	gaussian	0.1
	bdg2_Office_Eagle	0.1	0.5	power	orthogonal	cosine	0.1
	solar_poli	0.1	0.5	power	random	cosine	0.1
	zurich_ne5	3.0	0.0	power	random	gaussian	0.1

Table 13: Best ResCP configurations for each model and dataset.

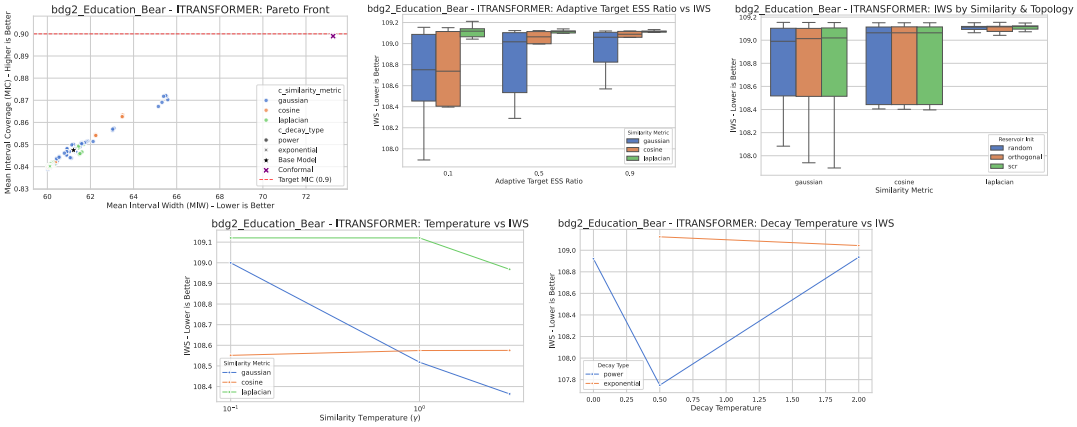
A.1.3 Comprehensive Sensitivity Plots (Phase 2)

The following figures depict the Pareto front (MIC vs. MIW), the IWS distribution by topology, the temperature (decay and similarity) sensitivity the `adaptive_target_ess_ratio` for all remaining dataset-architecture pairs.

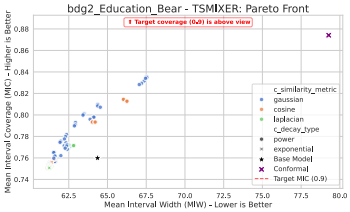
bdg2_Education_Bear (TiDE)

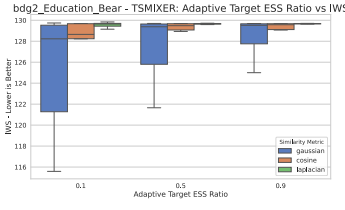


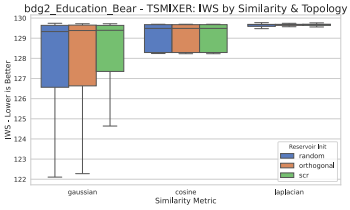
bdg2_Education_Bear (iTransformer)

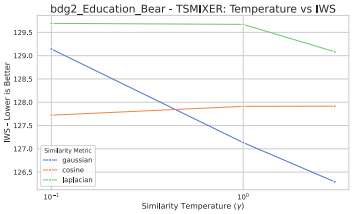


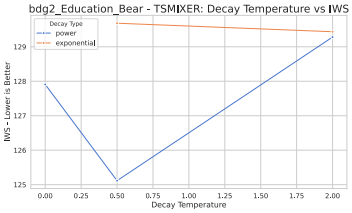
bdg2_Education_Bear (TSMixer)



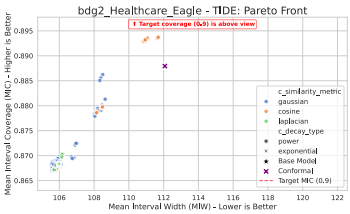


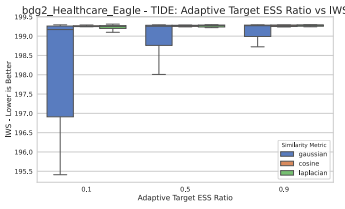


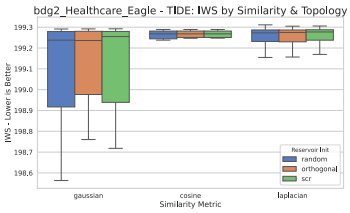


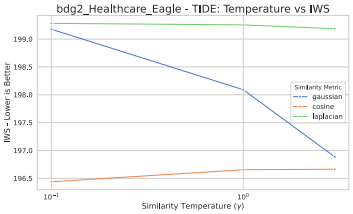


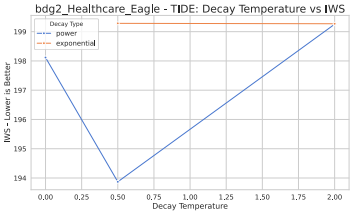
bdg2_Healthcare_Eagle (TiDE)



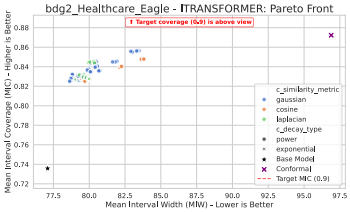


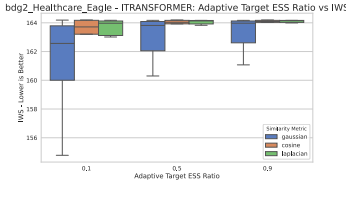


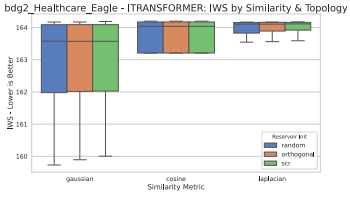


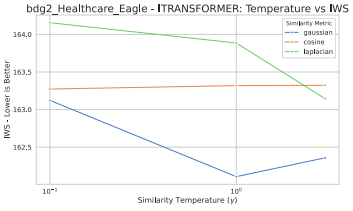


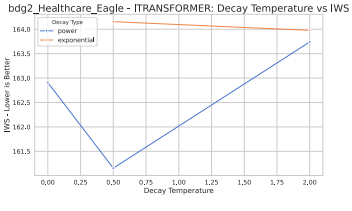
bdg2_Healthcare_Eagle (iTransformer)



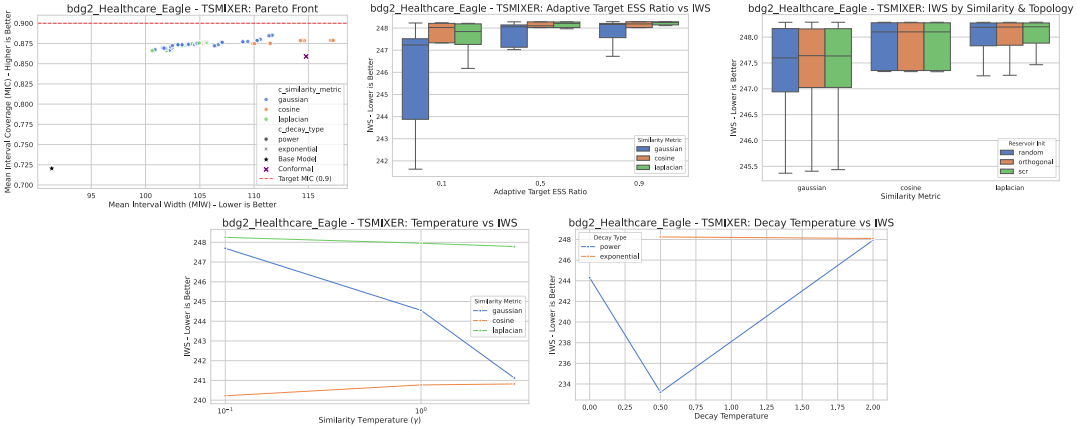




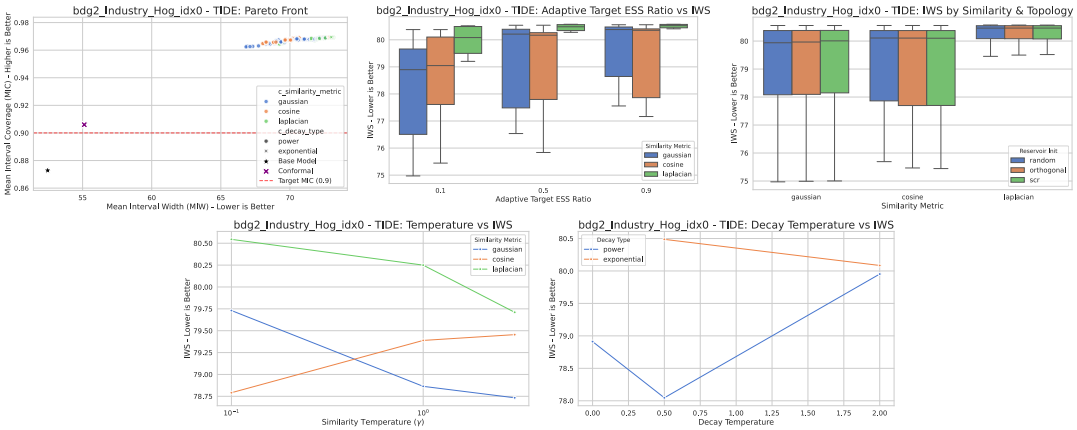




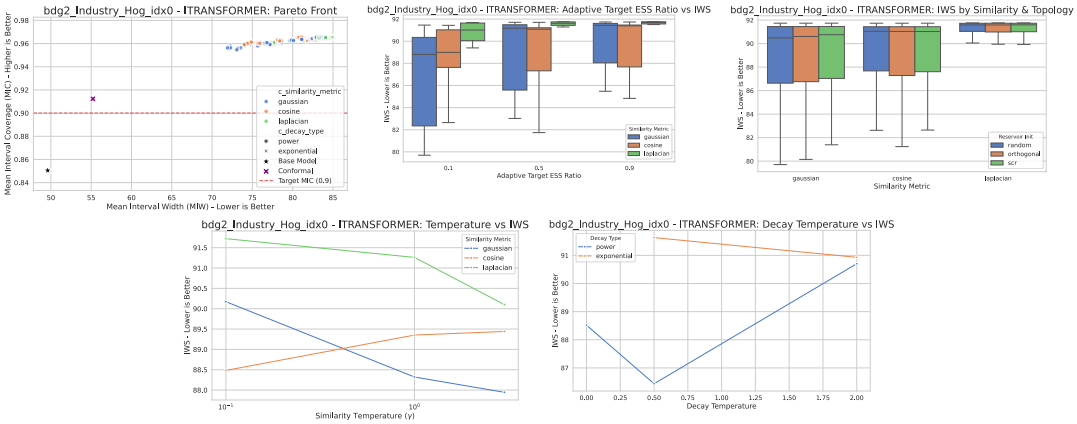
bdg2_Healthcare_Eagle (TSMixer)



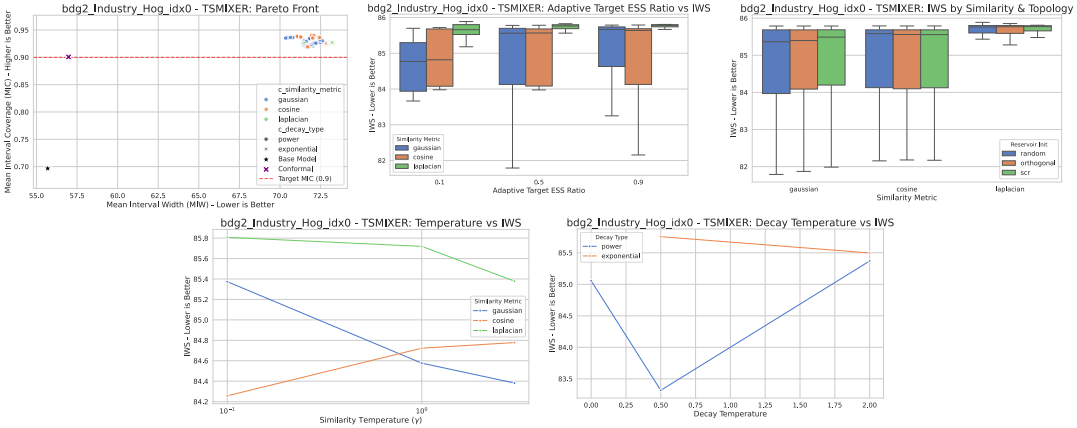
bdg2_Industry_Hog_idx0 (TiDE)



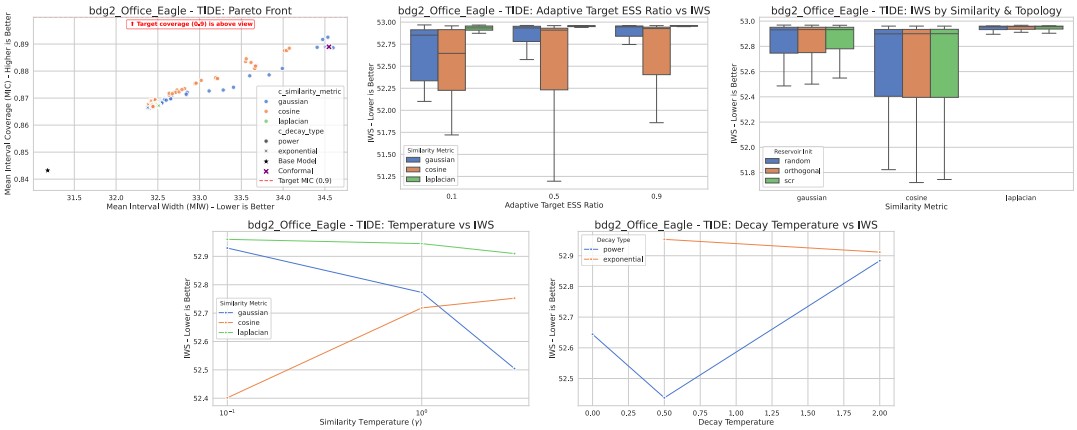
bdg2_Industry_Hog_idx0 (iTransformer)



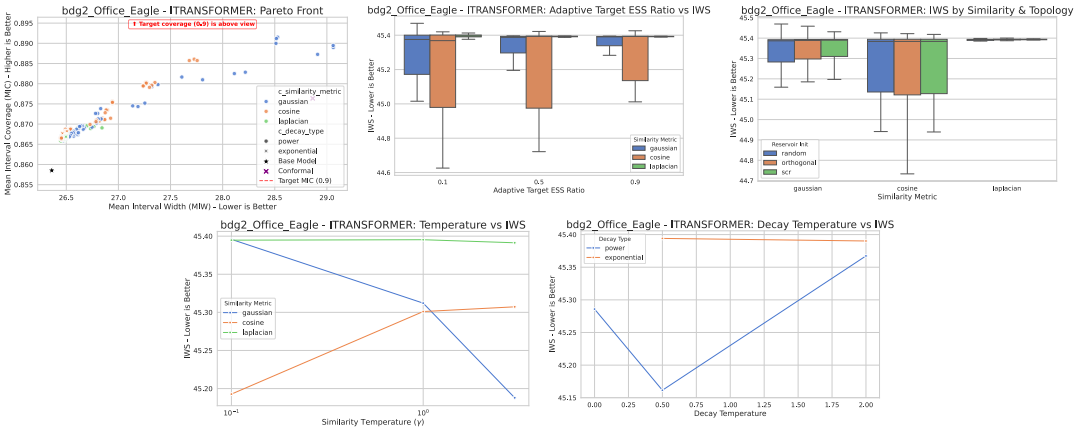
bdg2_Industry_Hog_idx0 (TSMixer)



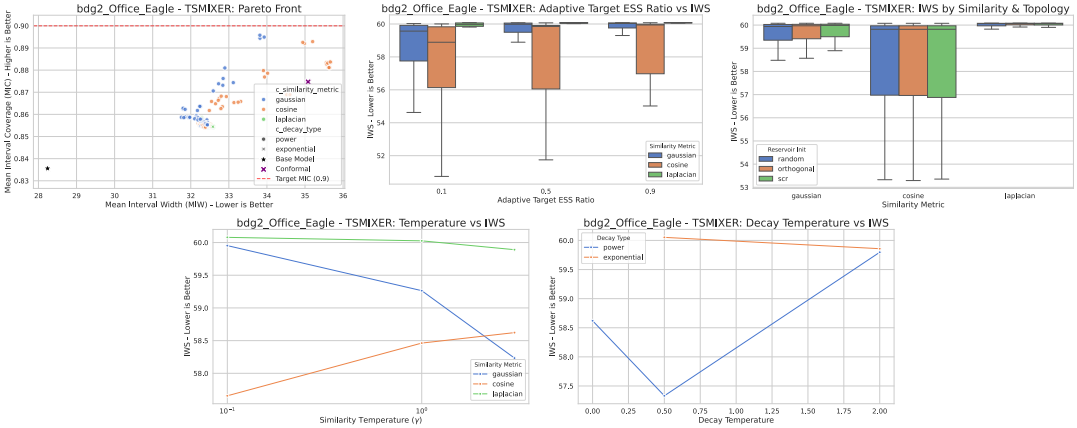
bdg2_Office_Eagle (TiDE)



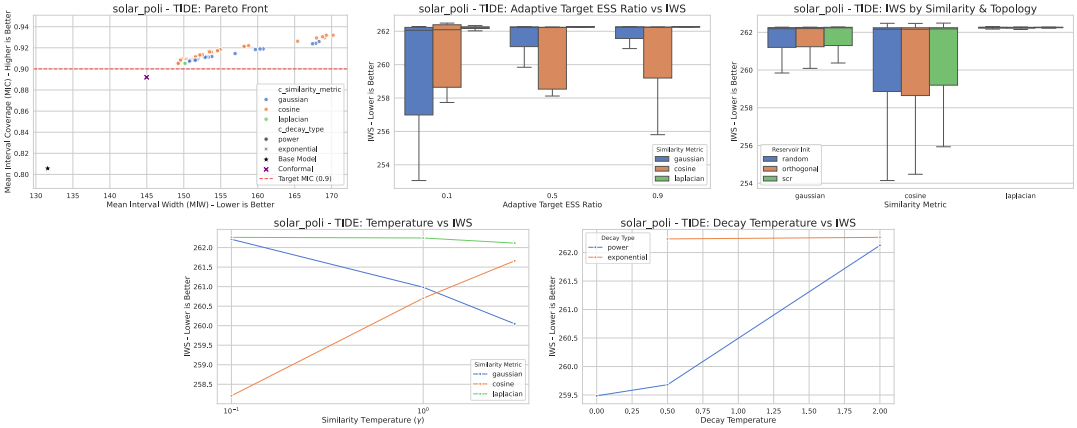
bdg2_Office_Eagle (iTransformer)



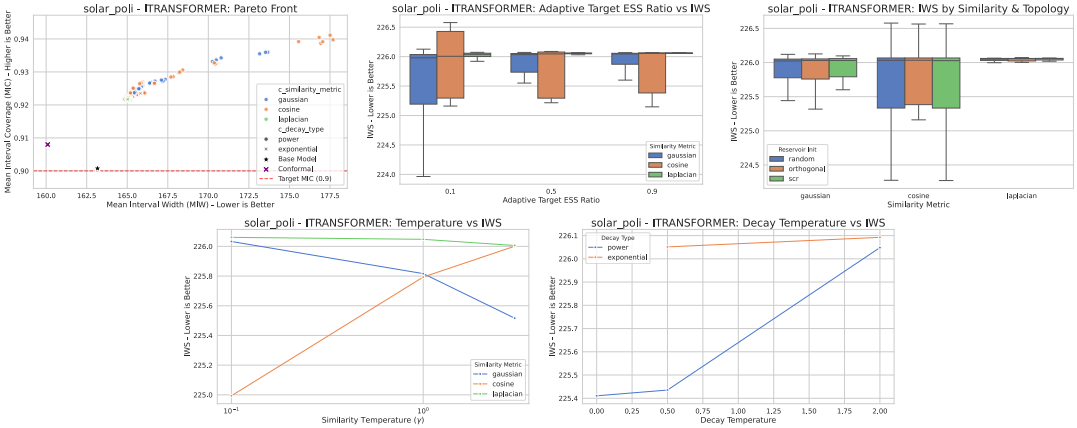
bdg2_Office_Eagle (TSMixer)



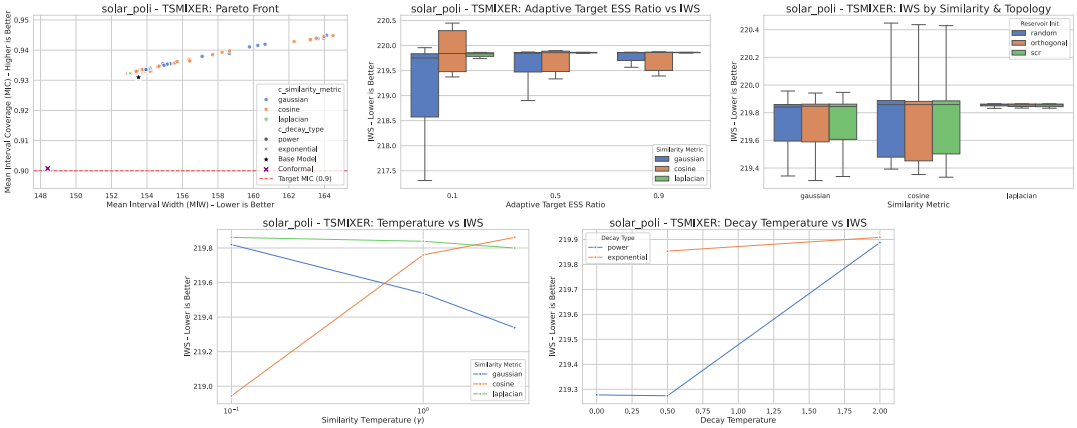
solar_poli (TiDE)



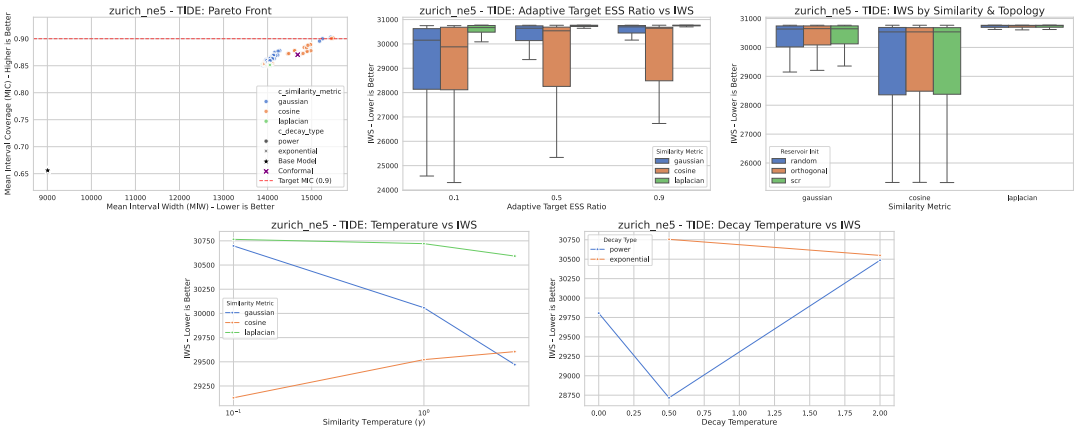
solar_poli (iTransformer)



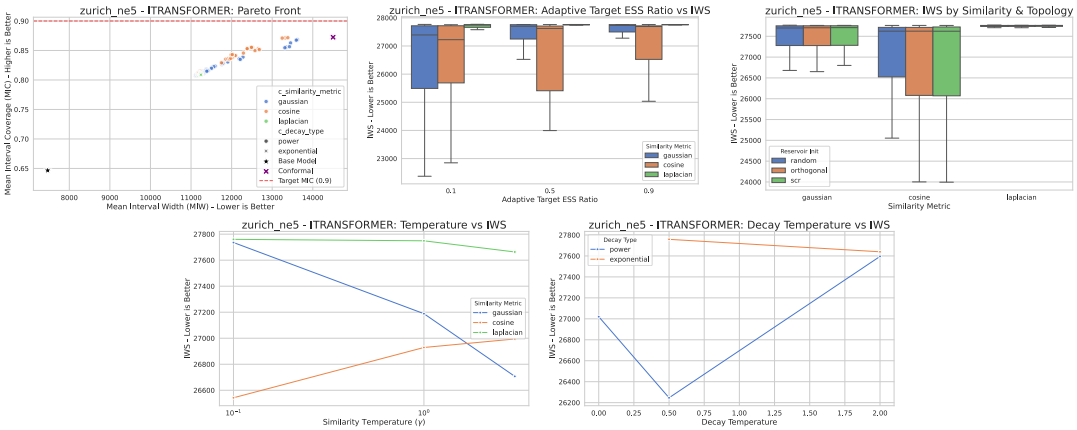
solar_poli (TSMixer)



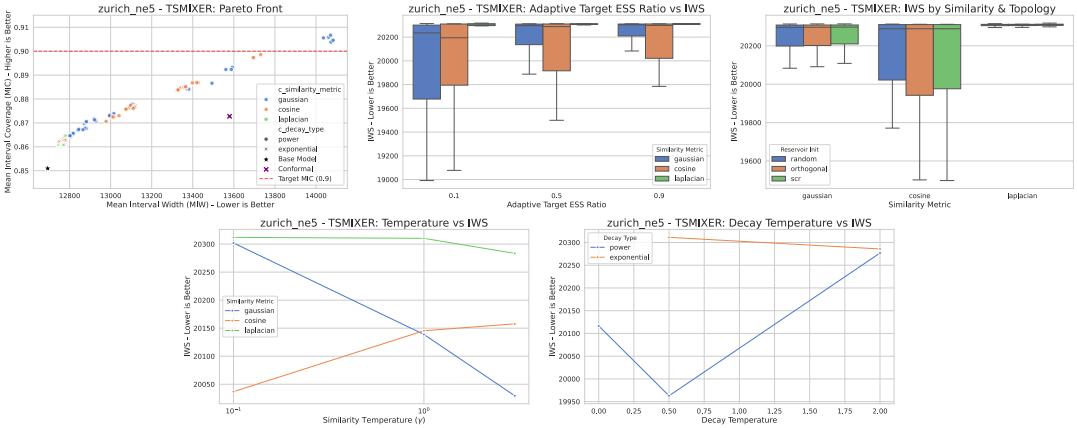
zurich_ne5 (TiDE)



zurich_ne5 (iTransformer)



zurich_ne5 (TSMixer)



A.1.4 Hyperparameter choice for base models for phase 3

Common Parameters	
Parameter	**Value**
Output Chunk Length	24
Input Chunk Length	[96, 168]
Dropout	$\sim \mathcal{U}(0.05, 0.25)$
Quantiles	(0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95)
Huber Pinball Loss δ	$5 \cdot 10^{-4}$
Batch Size	128
Num Epochs	50
Early Stopping Patience	10

Table 14: Common hyperparameter configuration for the base models.

TiDE	
Parameter	**Value**
hidden_size	[64, 128]
num_encoder_layers	[2, 3]
num_decoder_layers	2

Table 15: Hyperparameter configuration for TiDE

iTransformer	
Parameter	**Value**
d_model	[128, 256]
num_heads	[2, 4]
num_layers	2
d_ff	[256, 512]

Table 16: Hyperparameter configuration for iTransformer

TSMixer	
Parameter	**Value**
hidden_size	[64, 128]
ff_size	[64, 128]
num_blocks	[1, 2]

Table 17: Hyperparameter configuration for TSMixer

A.2 Global Results (Phase 3)

This appendix presents a comprehensive overview of the global results obtained from our extensive grid search and validation phases.

A.2.1 Global Results Summary

In this section, we present a comprehensive overview of the global results obtained from our extensive grid search and validation phases. Table 18 summarizes the key performance metrics across all benchmark datasets and model architectures, providing insights into the reliability and effectiveness of each approach in forecasting tasks.

Table 18: Quantitative comparison of forecasting performance across all benchmark datasets. The Δ Cov. column indicates the deviation from the 90% target nominal coverage, color-coded for reliability: green for acceptable bounds ($\Delta \geq -2.5\%$), orange for moderate under-coverage ($[-5\%, -2.5\%)$), and red for severe under-coverage ($< -5\%$). The lowest (best) IWS for each architecture is highlighted in **bold**, while the absolute best IWS across all models for a given dataset is additionally marked in **blue**.

Dataset	Method	TiDE			iTransformer			TSMixer		
		Δ Cov.	MIW	IWS	Δ Cov.	MIW	IWS	Δ Cov.	MIW	IWS
australia	Base	−4.1%	667.51	1097.89	+3.9%	997.62	1189.60	−2.0%	790.46	1152.58
	ResCP	+3.5%	832.14	1061.59	+4.3%	988.48	1179.68	+2.2%	858.98	**1119.17**
	CQR	−0.7%	712.00	**1059.00**	+0.5%	865.70	**1147.22**	+0.4%	821.57	1126.65
Education_Bear	Base	−2.0%	72.90	109.24	−8.0%	58.23	109.40	−16.1%	62.75	136.28
	ResCP	−1.0%	70.64	**104.72**	−3.3%	62.89	**101.63**	−5.1%	69.86	**117.73**
	CQR	−0.5%	75.39	107.50	−1.6%	69.21	104.53	−2.7%	84.34	123.51
Education_Eagle	Base	+0.6%	132.69	177.26	+2.9%	103.54	138.98	−2.5%	134.42	188.01
	ResCP	+3.4%	143.41	180.40	+1.7%	95.57	133.96	−2.0%	145.38	195.77
	CQR	−1.6%	104.73	**156.32**	−2.4%	81.43	**131.44**	−2.6%	111.79	**163.78**
Education_Fox	Base	−2.6%	69.24	107.08	−19.8%	73.76	167.14	−40.3%	84.00	283.71
	ResCP	+2.9%	75.84	100.05	+1.5%	79.45	107.33	+2.2%	94.97	122.12
	CQR	−1.3%	65.43	**98.00**	−0.3%	65.74	**97.42**	−0.7%	71.05	**105.82**
Education_Hog	Base	−4.8%	31.86	44.61	−2.3%	33.23	44.07	−15.9%	25.65	47.75
	ResCP	+1.5%	32.84	**39.15**	+1.2%	32.01	**40.49**	−0.5%	29.29	**37.45**
	CQR	−2.2%	29.46	40.24	−1.4%	30.53	42.42	−1.0%	28.68	38.11
Education_Moose	Base	−31.8%	151.87	567.74	−32.7%	237.00	586.42	−64.2%	149.83	1038.28
	ResCP	+3.8%	224.05	384.30	+4.8%	250.33	356.52	+5.1%	233.54	315.59
	CQR	−1.7%	164.74	**372.70**	−0.1%	170.89	**326.81**	−2.1%	154.80	**286.85**
Education_Peacock	Base	−14.7%	22.89	49.23	−5.5%	22.67	38.15	−19.2%	23.85	59.76
	ResCP	+0.7%	27.54	**38.22**	−0.9%	24.64	**35.67**	+0.3%	31.41	**42.44**
	CQR	−2.8%	26.26	39.42	−0.8%	25.45	37.18	−2.7%	32.89	47.15
Education_Rat	Base	−14.7%	0.34	1.03	−16.9%	0.46	1.05	−12.9%	0.39	1.03
	ResCP	+2.7%	0.47	**0.85**	+2.2%	0.53	**0.87**	+1.7%	0.50	**0.89**
	CQR	−1.5%	0.40	0.85	−0.5%	0.50	0.89	−2.7%	0.44	0.94
Education_Wolf	Base	−12.4%	48.73	131.89	−7.7%	58.58	136.34	−4.0%	62.62	102.36
	ResCP	+2.8%	83.76	**111.98**	−0.0%	77.90	**125.65**	+3.1%	76.75	**95.19**
	CQR	−2.3%	71.12	124.38	−2.8%	69.41	134.60	−0.6%	69.96	102.63
Healthcare_Eagle	Base	−7.4%	104.63	208.25	−11.0%	74.56	169.24	−16.3%	84.71	246.23
	ResCP	−2.2%	103.86	**176.51**	−5.1%	81.46	**153.67**	−2.1%	119.68	202.03
	CQR	−1.2%	113.66	184.99	−2.6%	96.40	156.65	−3.6%	117.08	**194.42**
Healthcare_Fox	Base	−8.0%	14.57	27.90	−3.2%	15.25	24.09	−21.3%	11.68	30.57
	ResCP	+0.9%	17.02	**24.44**	+2.1%	14.95	**21.53**	−2.5%	14.37	**23.39**
	CQR	−0.3%	17.21	25.31	−0.4%	15.36	22.55	−0.8%	15.32	23.57
Healthcare_Hog	Base	−7.2%	24.40	46.27	+3.1%	32.32	42.57	−12.1%	30.03	57.02
	ResCP	+2.3%	28.51	**40.95**	+3.5%	30.87	40.83	+3.5%	32.74	**40.62**
	CQR	−1.6%	26.07	40.96	+0.4%	26.48	**39.35**	−3.0%	29.68	43.50
Healthcare_Rat	Base	−10.4%	17.79	33.75	−2.7%	18.08	30.31	−1.1%	20.79	30.82
	ResCP	+0.6%	23.16	**30.56**	+2.9%	22.09	**29.19**	+2.0%	22.76	**29.87**
	CQR	−1.4%	21.23	30.85	+0.5%	20.32	30.10	−0.1%	22.40	31.53

Dataset	Method	TiDE			iTransformer			TSMixer		
		Δ Cov.	MIW	IWS	Δ Cov.	MIW	IWS	Δ Cov.	MIW	IWS
Industry_Hog	Base	−2.7%	53.56	76.70	−7.2%	50.86	77.69	−17.6%	44.74	99.70
	ResCP	+6.5%	68.31	75.22	+6.3%	68.29	74.53	+4.9%	66.11	73.38
	CQR	+1.2%	55.76	**71.42**	+1.2%	54.59	**68.32**	+0.0%	55.69	**69.55**
Industry_Hog_idx2	Base	−6.0%	85.70	141.75	+4.3%	109.46	120.37	+4.7%	114.29	127.92
	ResCP	+0.9%	102.09	**127.79**	+5.1%	106.68	116.25	+3.6%	104.88	**121.67**
	CQR	−1.8%	94.15	133.57	+0.7%	90.54	**113.13**	−1.6%	96.56	127.41
Industry_Hog_idx3	Base	−17.5%	4.30	10.97	−14.5%	6.34	16.01	−41.8%	4.46	17.52
	ResCP	−2.0%	5.61	8.25	−2.0%	7.84	10.41	−3.4%	7.10	9.80
	CQR	−2.5%	5.73	**8.23**	−2.4%	6.96	**9.67**	−2.1%	5.66	**7.94**
Office_Eagle	Base	−5.4%	29.20	52.05	−3.2%	28.25	47.79	−12.4%	21.95	53.50
	ResCP	+0.7%	34.32	**47.72**	−0.7%	28.71	**44.89**	−2.2%	29.31	**45.10**
	CQR	−1.0%	32.47	49.64	−2.3%	29.25	47.99	−1.7%	30.09	46.18
Office_Fox	Base	−10.2%	35.20	73.38	+2.3%	57.02	73.49	−14.2%	37.50	80.35
	ResCP	+1.5%	42.89	64.49	+1.9%	46.77	63.66	+0.8%	44.80	**62.61**
	CQR	−1.3%	40.46	**62.90**	−2.8%	40.83	**63.40**	−3.3%	40.49	64.86
Office_Hog	Base	−6.3%	16.96	27.74	−0.3%	14.82	23.35	−4.2%	19.00	28.78
	ResCP	+1.1%	19.19	**24.92**	+2.6%	16.58	**23.26**	+1.1%	19.74	**25.96**
	CQR	−1.7%	17.74	26.25	−0.3%	15.27	24.54	−1.8%	19.40	28.30
Office_Peacock	Base	−9.7%	27.76	54.41	−0.9%	29.52	46.48	−9.2%	30.84	56.93
	ResCP	−1.3%	33.16	**49.29**	−0.4%	29.94	**45.70**	−2.3%	33.11	**50.42**
	CQR	−3.3%	30.19	49.64	−1.5%	28.60	47.10	−3.6%	33.04	52.95
Office_Rat	Base	−31.8%	154.39	852.42	−11.1%	187.09	453.45	−46.2%	147.57	1746.17
	ResCP	−7.8%	244.87	508.92	−1.7%	224.03	**403.55**	−7.9%	337.31	558.54
	CQR	−2.1%	262.44	**490.92**	−1.5%	218.85	428.18	−2.9%	300.99	**523.06**
Office_Robin	Base	−7.7%	10.50	22.42	−1.5%	11.27	19.04	−11.0%	9.42	22.58
	ResCP	+2.1%	13.48	**19.46**	−0.8%	10.92	**17.74**	+3.3%	14.70	**19.54**
	CQR	−1.7%	12.77	20.58	−0.2%	11.90	18.51	−1.5%	13.88	21.43
Office_Wolf	Base	−8.4%	15.37	29.22	−7.1%	14.73	26.87	−16.6%	9.55	28.36
	ResCP	+0.5%	18.91	**25.17**	−0.2%	17.78	**25.40**	−0.4%	15.51	**24.67**
	CQR	−2.2%	16.71	29.24	−2.4%	15.21	28.08	−2.0%	13.95	27.03
solar_poli	Base	−5.5%	132.89	249.33	+2.8%	156.29	219.90	−0.2%	142.56	222.23
	ResCP	+3.7%	165.95	**234.94**	+4.2%	166.53	**219.48**	+3.7%	161.94	**217.35**
	CQR	−0.5%	145.14	263.36	+1.0%	152.73	236.02	−0.1%	144.94	236.89
zurich_ne5	Base	−22.1%	8882.77	37107.82	−18.5%	11203.79	41581.71	−4.0%	12502.46	20429.55
	ResCP	−1.1%	13477.69	**21683.88**	+2.0%	16416.20	**22893.13**	+1.4%	14182.25	**19167.47**
	CQR	−2.5%	13395.15	22515.28	−2.0%	15713.97	25029.95	−1.7%	13648.68	20237.36
zurich_ne7	Base	−37.0%	9914.21	66072.96	−9.0%	23929.30	55796.97	−7.1%	21345.59	37543.83
	ResCP	−3.6%	21993.33	40966.47	−2.9%	29551.29	51771.38	−0.3%	26739.79	37173.80
	CQR	−3.6%	20966.17	**38830.22**	−2.6%	27117.07	**42099.41**	−2.3%	22820.19	**35571.70**

Appendix : Point-Forecast Benchmark: Deep Models vs. Gradient Boosting Baseline,

B.1 Motivation and Scope

The core contributions of this thesis, presented in Chapter 3, Chapter 4, and Chapter 6, concern the calibration of prediction intervals and are evaluated accordingly, through coverage and interval-width metrics. This appendix instead reports a complementary benchmark conducted as part of the author's R&D internship at Modelway, comparing the point-forecast accuracy of the deep learning architectures adopted throughout this work (TSMixer, iTransformer, TiDE) against XGBoost, the gradient-boosting baseline currently used in production.

This benchmark differs from the main body of the thesis along three axes. First, the evaluation is purely point-based (MAE, RMSE, MAPE), rather than interval-based, since the goal here is to assess raw forecasting accuracy rather than calibration quality. Second, the datasets and preprocessing pipeline differ from those used in Chapters 4-6: the data here undergo a distinct processing pipeline. Third, rather than training separate model instances for each dataset, the deep learning architectures are evaluated under a globally-shared weights regime (a single network trained jointly on all data), reflecting the production deployment scenario. Crucially, within this shared-model regime, we investigate different scaling granularities (per-dataset versus per-category) to evaluate the trade-off between raw predictive accuracy and the transferability required by production environments. Because this comparison originates from a business requirement — justifying the adoption of deep learning architectures over the company's existing baseline — rather than from the research questions driving the main contributions of this thesis, it is reported

separately here rather than integrated into the main results. Its purpose is to provide evidence, in an applied setting, of the accuracy trade-offs and deployment considerations (e.g., scaling granularity, generalization across building categories) associated with each modeling choice.

B.2 Experimental Setup

To ensure a fair comparison between the deep learning models proposed in this work and the production baseline currently adopted at Modelway, both approaches are evaluated under an identical black-box protocol: models are trained to forecast a horizon of $H = 8$ steps but evaluated only on the first $H = 4$ steps, a strategy adopted to encourage better generalization during training. This protocol is applied identically to the baseline and to all proposed models, so that the input-output data exposure is held constant regardless of the differences in internal preprocessing described below. All models receive an input window of 96 past observations, corresponding to four days of history at hourly granularity.

Baseline. The production baseline is a single XGBoost model, globally fit across all datasets jointly. Preprocessing consists of differencing followed by a globally-fit StandardScaler, consistent with the pipeline currently deployed in production.

Proposed models. TSMixer, iTransformer, and TiDE are evaluated without differencing, as differencing was found to substantially degrade performance for all three architectures during preliminary experimentation; this motivated a switch to a purely scaling-based preprocessing pipeline. Two scaling configurations are compared, both using a single fitted model (i.e., architecture and weights are shared; only the input scaling granularity changes):

- **Experiment 1 (per-dataset scaling):** one scaler is fit per individual dataset (N scalers total), all feeding into a single model instance.
- **Experiment 2 (per-category scaling):** one scaler is fit per building category (School, Hospital, Industry, Office; C scalers total), again feeding into a single model instance.

The per-category configuration is motivated by deployment considerations: a scaler defined at the category level supports a more plug-and-play integration across new datasets belonging to a known category, at the cost of some predictive accuracy relative to per-dataset scaling, as discussed in Appendix B.4.

All models are trained for a maximum of 100 epochs, with early stopping triggered after 20 epochs without improvement on the validation set. Performance is evaluated using MAE, RMSE, and MAPE.

B.3 Results

Table 19 reports point-forecast accuracy (MAE, RMSE, MAPE) for the baseline XGBoost model and the three proposed deep learning architectures under the per-dataset scaling configuration (Experiment 1), while Table 20 reports the same metrics under the per-category scaling configuration (Experiment 2). All metrics are computed on the evaluation horizon $H = 4$, as described in Appendix B.2. In both tables, each row corresponds to an individual dataset; the best result for each metric is highlighted in bold. Lower values indicate better performance for all three metrics.

Table 21 extends the evaluation to a zero-shot generalisation scenario: the five datasets reported therein belong to building categories already present in the training split (Industry, K-12 School, Office), but the individual meters were never seen during training. Models are evaluated as-is, without any fine-tuning on the target series. The results reveal a sharp divergence in generalisation ability. iTransformer transfers consistently well, outperforming XGBoost on every dataset and metric by substantial margins (mean MAE -22.4%, MAPE -25.5%), confirming that the attention-based cross-variate representation it learns is robust to distribution shift at the meter level. TiDE also generalises, but its advantage over XGBoost essentially vanishes in this setting: it remains slightly worse than the baseline on average ($+5.1\%$ MAE), suggesting that its gains in the seen-dataset experiments stem partly from in-distribution fitting rather than genuine transferability. TSMixer, by contrast, fails entirely in the zero-shot regime, degrading to more than three times the baseline error on average ($+223.5\%$ MAE), which confirms that the channel-mixing mechanism that benefits it under per-dataset scaling cannot compensate for the distributional mismatch introduced by unseen meters.

B.4 Discussion

B.4.1 Overall Ranking

Across both scaling configurations and all three metrics, iTransformer consistently outperforms the XGBoost baseline by a substantial margin, achieving mean error

Dataset	MAE				RMSE				MAPE (%)			
	XGB	TSM	TiDE	iTr	XGB	TSM	TiDE	iTr	XGB	TSM	TiDE	iTr
Hospital												
Hog_Hisako	6.53 -	4.96 (-24.1%)	5.86 (-10.3%)	**4.89** (-25.2%)	9.74 -	**7.73** (-20.6%)	8.96 (-8.0%)	7.86 (-19.3%)	3.32 -	2.54 (-23.3%)	2.97 (-10.6%)	**2.49** (-24.8%)
Hog_Jenny	8.57 -	6.57 (-23.4%)	7.62 (-11.1%)	**5.61** (-34.6%)	14.98 -	9.85 (-34.3%)	11.41 (-23.9%)	**9.05** (-39.6%)	4.95 -	3.74 (-24.4%)	4.36 (-12.0%)	**3.31** (-33.2%)
Hog_Kesha	7.59 -	4.97 (-34.5%)	6.10 (-19.6%)	**4.71** (-38.0%)	12.95 -	**7.17** (-44.6%)	9.43 (-27.1%)	8.05 (-37.8%)	3.18 -	**2.04** (-35.9%)	2.52 (-20.7%)	2.05 (-35.4%)
Rat_Gaye	4.90 -	3.46 (-29.5%)	3.82 (-22.0%)	**3.09** (-37.0%)	6.46 -	4.70 (-27.3%)	5.41 (-16.4%)	**4.40** (-31.8%)	12.69 -	8.25 (-35.0%)	9.39 (-26.0%)	**7.68** (-39.4%)
Industry												
LG 1	16.34 -	11.77 (-27.9%)	16.99 (+4.0%)	**8.89** (-45.6%)	28.36 -	21.77 (-23.2%)	27.82 (-1.9%)	**18.47** (-34.9%)	38.55 -	23.17 (-39.9%)	39.60 (+2.7%)	**16.13** (-58.2%)
LG 12	6.13 -	4.16 (-32.1%)	5.46 (-11.0%)	**3.52** (-42.6%)	9.36 -	7.41 (-20.8%)	8.46 (-9.7%)	**7.07** (-24.5%)	16.35 -	10.09 (-38.3%)	14.08 (-13.9%)	**8.30** (-49.3%)
LG 14	20.69 -	18.60 (-10.1%)	21.97 (+6.2%)	**15.16** (-26.7%)	29.17 -	28.13 (-3.5%)	31.45 (+7.8%)	**23.37** (-19.9%)	17.93 -	15.52 (-13.4%)	19.39 (+8.2%)	**12.01** (-33.0%)
LG 18	38.98 -	36.27 (-6.9%)	40.22 (+3.2%)	**31.39** (-19.5%)	56.18 -	54.93 (-2.2%)	57.75 (+2.8%)	**49.25** (-12.3%)	58.18 -	55.71 (-4.3%)	66.11 (+13.6%)	**42.35** (-27.2%)
LG 2	73.67 -	67.02 (-9.0%)	72.42 (-1.7%)	**53.53** (-27.3%)	108.20 -	105.68 (-2.3%)	112.76 (+4.2%)	**82.70** (-23.6%)	11.70 -	10.22 (-12.6%)	11.52 (-1.6%)	**7.84** (-33.0%)
LG 5	7.01 -	5.46 (-22.1%)	6.42 (-8.4%)	**4.76** (-32.1%)	10.41 -	9.23 (-11.4%)	10.07 (-3.3%)	**8.41** (-19.2%)	25.45 -	17.85 (-29.9%)	23.75 (-6.7%)	**14.24** (-44.0%)
LG 9	51.52 -	46.10 (-10.5%)	62.08 (+20.5%)	**39.28** (-23.8%)	84.76 -	79.80 (-5.8%)	100.27 (+18.3%)	**71.91** (-15.2%)	30.67 -	23.93 (-22.0%)	38.75 (+26.4%)	**22.17** (-27.7%)
K-12 School												
Rat_Adell	21.15 -	20.82 (-1.6%)	23.38 (+10.6%)	**17.89** (-15.4%)	29.43 -	29.79 (+1.2%)	31.80 (+8.1%)	**25.96** (-11.8%)	14.88 -	13.92 (-6.5%)	17.17 (+15.3%)	**12.01** (-19.3%)
Rat_Adrian	7.85 -	7.83 (-0.2%)	7.63 (-2.8%)	**5.00** (-36.3%)	11.96 -	11.98 (+0.2%)	11.76 (-1.6%)	**8.28** (-30.8%)	11.32 -	9.09 (-19.7%)	11.56 (+2.1%)	**6.63** (-41.4%)
Rat_Beverly	100.22 -	100.44 (+0.2%)	103.04 (+2.8%)	**64.78** (-35.4%)	151.96 -	141.14 (-7.1%)	146.89 (-3.3%)	**96.55** (-36.5%)	13.44 -	12.17 (-9.5%)	13.54 (+0.7%)	**8.46** (-37.0%)
Rat_Carmela	11.71 -	10.17 (-13.1%)	11.93 (+1.9%)	**8.94** (-23.6%)	16.25 -	14.54 (-10.5%)	16.54 (+1.8%)	**12.95** (-20.3%)	13.86 -	10.63 (-23.3%)	13.18 (-4.9%)	**9.80** (-29.3%)
Rat_Keith	8.97 -	8.31 (-7.4%)	9.02 (+0.5%)	**7.17** (-20.1%)	12.54 -	11.54 (-8.0%)	13.06 (+4.1%)	**10.26** (-18.2%)	7.39 -	6.48 (-12.3%)	7.15 (-3.2%)	**5.80** (-21.5%)
Rat_Marianna	8.67 -	9.41 (+8.6%)	8.39 (-3.2%)	**6.01** (-30.6%)	12.73 -	13.73 (+7.9%)	12.47 (-2.0%)	**9.15** (-28.1%)	9.89 -	8.54 (-13.7%)	9.23 (-6.7%)	**6.41** (-35.2%)
Rat_Nellie	44.21 -	42.02 (-4.9%)	43.66 (-1.2%)	**34.71** (-21.5%)	64.37 -	56.92 (-11.6%)	60.96 (-5.3%)	**51.00** (-20.8%)	9.67 -	8.95 (-7.5%)	9.48 (-2.0%)	**7.69** (-20.4%)
Rat_Wilmer	9.65 -	8.09 (-16.2%)	9.79 (+1.4%)	**7.50** (-22.3%)	13.77 -	11.78 (-14.5%)	13.62 (-1.1%)	**11.19** (-18.7%)	9.62 -	7.84 (-18.4%)	9.43 (-2.0%)	**7.29** (-24.2%)
Office												
Eagle_Isidro	5.35 -	3.01 (-43.8%)	3.57 (-33.2%)	**2.49** (-53.5%)	7.70 -	4.85 (-37.0%)	5.45 (-29.2%)	**4.42** (-42.6%)	29.60 -	16.00 (-46.0%)	20.77 (-29.8%)	**12.40** (-58.1%)
Eagle_Patrice	8.55 -	6.19 (-27.6%)	6.91 (-19.2%)	**5.72** (-33.1%)	11.86 -	9.10 (-23.3%)	9.91 (-16.5%)	**8.47** (-28.5%)	9.62 -	6.74 (-29.9%)	7.73 (-19.7%)	**6.27** (-34.9%)
Fox_Alice	10.44 -	8.50 (-18.5%)	8.49 (-18.7%)	**6.93** (-33.7%)	21.36 -	14.73 (-31.1%)	16.60 (-22.3%)	**13.50** (-36.8%)	3.18 -	2.61 (-18.0%)	2.65 (-16.8%)	**2.18** (-31.5%)
Fox_Carson	4.55 -	2.83 (-37.8%)	2.98 (-34.4%)	**2.54** (-44.1%)	6.39 -	4.45 (-30.4%)	4.51 (-29.4%)	**4.17** (-34.7%)	10.16 -	6.29 (-38.2%)	6.77 (-33.4%)	**5.49** (-46.0%)
Hog_Josefina	6.82 -	7.09 (+4.0%)	6.60 (-3.2%)	**5.50** (-19.3%)	9.17 -	9.63 (+5.0%)	9.07 (-1.1%)	**8.15** (-11.2%)	3.78 -	3.81 (+0.9%)	3.66 (-3.1%)	**2.99** (-20.9%)
Hog_Shon	5.87 -	5.46 (-7.0%)	5.09 (-13.3%)	**3.95** (-32.6%)	7.99 -	8.01 (+0.2%)	7.41 (-7.3%)	**5.74** (-28.2%)	7.56 -	6.69 (-11.5%)	6.48 (-14.3%)	**4.86** (-35.7%)
Peacock_Effie	10.46 -	7.98 (-23.7%)	8.87 (-15.2%)	**7.40** (-29.2%)	14.04 -	10.53 (-25.0%)	11.98 (-14.7%)	**10.22** (-27.2%)	13.36 -	10.12 (-24.2%)	11.40 (-14.7%)	**9.30** (-30.4%)
Peacock_Elton	6.19 -	5.31 (-14.3%)	5.98 (-3.5%)	**5.16** (-16.8%)	8.08 -	7.05 (-12.8%)	7.82 (-3.3%)	**6.80** (-15.9%)	3.75 -	3.19 (-15.1%)	3.65 (-2.7%)	**3.11** (-17.1%)
Mean	18.98 -	17.14 (-16.1%)	19.05 (-6.7%)	**13.57** (-30.4%)	28.53 -	25.78 (-14.6%)	28.28 (-6.7%)	**21.38** (-25.5%)	14.59 -	11.34 (-21.2%)	14.31 (-6.5%)	**9.23** (-33.6%)

Table 19: Point-forecast accuracy (MAE, RMSE, MAPE) for XGBoost and proposed deep learning models under the per-dataset scaling configuration (Experiment 1). Best results per metric are in bold.

Dataset	MAE				RMSE				MAPE (%)			
	XGB	TSM	TiDE	iTr	XGB	TSM	TiDE	iTr	XGB	TSM	TiDE	iTr
Hospital												
Hog_Hisako	6.53 -	6.11 (-6.5%)	5.81 (-11.1%)	**5.02** (-23.2%)	9.74 -	9.22 (-5.3%)	8.88 (-8.8%)	**8.09** (-16.9%)	3.32 -	3.09 (-6.9%)	2.94 (-11.4%)	**2.56** (-22.9%)
Hog_Jenny	8.57 -	7.46 (-12.9%)	7.56 (-11.8%)	**5.77** (-32.7%)	14.98 -	11.12 (-25.8%)	11.24 (-25.0%)	**9.13** (-39.1%)	4.95 -	4.25 (-14.2%)	4.33 (-12.5%)	**3.40** (-31.3%)
Hog_Kesha	7.59 -	6.97 (-8.2%)	6.07 (-20.0%)	**4.80** (-36.7%)	12.95 -	9.61 (-25.8%)	9.35 (-27.8%)	**8.02** (-38.0%)	3.18 -	2.89 (-9.2%)	2.51 (-21.0%)	**2.10** (-34.1%)
Rat_Gaye	4.90 -	8.71 (+77.7%)	4.26 (-13.1%)	**3.33** (-32.0%)	6.46 -	10.15 (+57.1%)	5.97 (-7.6%)	**4.64** (-28.2%)	12.69 -	20.73 (+63.4%)	10.33 (-18.6%)	**8.53** (-32.7%)
Industry												
LG 1	16.34 -	15.60 (-4.5%)	16.62 (+1.8%)	**10.30** (-36.9%)	28.36 -	24.12 (-14.9%)	28.43 (+0.3%)	**20.71** (-27.0%)	38.55 -	35.10 (-8.9%)	39.38 (+2.1%)	**19.76** (-48.7%)
LG 12	6.13 -	11.45 (+86.8%)	6.16 (+0.5%)	**4.06** (-33.7%)	9.36 -	14.42 (+54.0%)	9.08 (-3.0%)	**7.62** (-18.6%)	16.35 -	28.87 (+76.6%)	15.78 (-3.5%)	**9.83** (-39.9%)
LG 14	20.69 -	22.81 (+10.3%)	22.10 (+6.9%)	**15.71** (-24.1%)	29.17 -	31.14 (+6.8%)	31.95 (+9.6%)	**22.95** (-21.3%)	17.93 -	18.21 (+1.6%)	19.64 (+9.5%)	**12.38** (-31.0%)
LG 18	38.98 -	35.99 (-7.7%)	40.69 (+4.4%)	**33.15** (-14.9%)	56.18 -	52.17 (-7.1%)	59.64 (+6.1%)	**49.72** (-11.5%)	58.18 -	45.57 (-21.7%)	67.11 (+15.4%)	**43.17** (-25.8%)
LG 2	73.67 -	58.67 (-20.4%)	70.05 (-4.9%)	**53.69** (-27.1%)	108.20 -	88.88 (-17.9%)	111.46 (+3.0%)	**85.00** (-21.4%)	11.70 -	9.30 (-20.5%)	10.91 (-6.8%)	**7.98** (-31.8%)
LG 5	7.01 -	9.94 (+41.9%)	7.30 (+4.2%)	**5.09** (-27.4%)	10.41 -	13.97 (+34.2%)	10.97 (+5.4%)	**8.36** (-19.7%)	25.45 -	29.43 (+15.6%)	27.00 (+6.1%)	**15.59** (-38.8%)
LG 9	51.52 -	49.98 (-3.0%)	60.34 (+17.1%)	**38.44** (-25.4%)	84.76 -	79.96 (-5.7%)	100.79 (+18.9%)	**68.92** (-18.7%)	30.67 -	27.51 (-10.3%)	36.28 (+18.3%)	**21.44** (-30.1%)
K-12 School												
Rat_Adell	21.15 -	21.49 (+1.6%)	23.88 (+12.9%)	**18.76** (-11.3%)	29.43 -	30.18 (+2.6%)	32.39 (+10.0%)	**26.82** (-8.9%)	14.88 -	14.37 (-3.4%)	17.23 (+15.8%)	**13.09** (-12.0%)
Rat_Adrian	7.85 -	8.82 (+12.3%)	8.22 (+4.8%)	**5.18** (-34.0%)	11.96 -	12.10 (+1.2%)	12.59 (+5.3%)	**8.62** (-27.9%)	11.32 -	11.52 (+1.8%)	12.58 (+11.1%)	**6.86** (-39.4%)
Rat_Beverly	100.22 -	118.06 (+17.8%)	106.22 (+6.0%)	**64.96** (-35.2%)	151.96 -	165.06 (+8.6%)	151.75 (-0.1%)	**97.65** (-35.7%)	13.44 -	13.46 (+0.1%)	13.86 (+3.1%)	**8.57** (-36.3%)
Rat_Carmela	11.71 -	12.63 (+7.9%)	12.45 (+6.3%)	**9.80** (-16.3%)	16.25 -	16.77 (+3.2%)	17.26 (+6.2%)	**13.79** (-15.1%)	13.86 -	13.71 (-1.1%)	13.40 (-3.4%)	**11.01** (-20.6%)
Rat_Keith	8.97 -	11.15 (+24.3%)	9.69 (+8.1%)	**7.27** (-19.0%)	12.54 -	14.15 (+12.8%)	14.01 (+11.7%)	**10.35** (-17.5%)	7.39 -	9.51 (+28.7%)	7.66 (+3.7%)	**5.97** (-19.2%)
Rat_Marianna	8.67 -	10.50 (+21.2%)	9.00 (+3.8%)	**6.22** (-28.2%)	12.73 -	13.51 (+6.1%)	13.30 (+4.5%)	**9.49** (-25.5%)	9.89 -	12.01 (+21.4%)	9.78 (-1.2%)	**6.76** (-31.7%)
Rat_Nellie	44.21 -	38.83 (-12.2%)	44.51 (+0.7%)	**35.83** (-19.0%)	64.37 -	53.48 (-16.9%)	62.29 (-3.2%)	**52.52** (-18.4%)	9.67 -	8.37 (-13.4%)	9.66 (-0.1%)	**7.97** (-17.6%)
Rat_Wilmer	9.65 -	11.55 (+19.7%)	10.52 (+9.0%)	**7.66** (-20.7%)	13.77 -	15.38 (+11.7%)	14.41 (+4.6%)	**11.14** (-19.1%)	9.62 -	11.33 (+17.8%)	9.98 (+3.7%)	**7.53** (-21.7%)
Office												
Eagle_Isidro	5.35 -	5.08 (-5.1%)	4.11 (-23.2%)	**3.02** (-43.6%)	7.70 -	6.75 (-12.4%)	5.90 (-23.4%)	**4.78** (-37.9%)	29.60 -	29.73 (+0.4%)	24.55 (-17.1%)	**16.47** (-44.4%)
Eagle_Patrice	8.55 -	7.05 (-17.6%)	7.21 (-15.8%)	**5.97** (-30.2%)	11.86 -	9.64 (-18.7%)	10.25 (-13.6%)	**8.60** (-27.5%)	9.62 -	7.84 (-18.6%)	8.11 (-15.7%)	**6.65** (-30.9%)
Fox_Alice	10.44 -	11.41 (+9.3%)	9.39 (-10.0%)	**7.19** (-31.2%)	21.36 -	17.95 (-16.0%)	17.10 (-20.0%)	**13.59** (-36.4%)	3.18 -	3.44 (+8.1%)	2.91 (-8.5%)	**2.26** (-29.1%)
Fox_Carson	4.55 -	5.12 (+12.5%)	3.37 (-25.9%)	**3.05** (-32.9%)	6.39 -	6.94 (+8.6%)	4.90 (-23.3%)	**4.74** (-25.8%)	10.16 -	10.59 (+4.2%)	7.54 (-25.9%)	**6.74** (-33.7%)
Hog_Josefina	6.82 -	7.78 (+14.2%)	6.74 (-1.2%)	**5.69** (-16.6%)	9.17 -	10.29 (+12.2%)	9.21 (+0.4%)	**8.20** (-10.6%)	3.78 -	4.30 (+13.7%)	3.74 (-1.1%)	**3.10** (-17.8%)
Hog_Shon	5.87 -	6.47 (+10.3%)	5.22 (-11.1%)	**4.39** (-25.2%)	7.99 -	9.07 (+13.5%)	7.54 (-5.7%)	**6.20** (-22.4%)	7.56 -	8.14 (+7.6%)	6.60 (-12.8%)	**5.51** (-27.2%)
Peacock_Effie	10.46 -	8.63 (-17.5%)	8.93 (-14.6%)	**7.51** (-28.2%)	14.04 -	11.50 (-18.1%)	12.11 (-13.7%)	**10.41** (-25.8%)	13.36 -	10.82 (-19.1%)	11.44 (-14.4%)	**9.48** (-29.0%)
Peacock_Elton	6.19 -	6.90 (+11.3%)	6.14 (-0.9%)	**5.27** (-15.0%)	8.08 -	8.93 (+10.5%)	8.03 (-0.7%)	**6.92** (-14.4%)	3.75 -	4.20 (+12.1%)	3.76 (+0.1%)	**3.19** (-14.9%)
Mean	18.98 -	19.45 (+9.8%)	19.35 (-2.9%)	**13.97** (-26.7%)	28.53 -	27.65 (+2.2%)	28.92 (-3.3%)	**21.74** (-23.3%)	14.59 -	14.75 (+4.7%)	14.78 (-3.1%)	**9.92** (-29.4%)

Table 20: Point-forecast accuracy (MAE, RMSE, MAPE) for XGBoost and proposed deep learning models under the per-category scaling configuration (Experiment 2). Best results per metric are in bold.

Dataset	MAE				RMSE				MAPE (%)			
	XGB	TSM	TiDE	iTr	XGB	TSM	TiDE	iTr	XGB	TSM	TiDE	iTr
Industry												
LG 16	8.06 -	29.01 (+260.1%)	9.29 (+15.4%)	**6.71** (-16.7%)	12.75 -	50.85 (+298.9%)	13.08 (+2.6%)	**11.87** (-6.9%)	56.35 -	190.20 (+237.5%)	70.85 (+25.7%)	**46.29** (-17.8%)
K-12 School												
Rat_Debra	8.20 -	22.74 (+177.2%)	8.91 (+8.6%)	**5.98** (-27.0%)	12.40 -	33.81 (+172.7%)	13.06 (+5.3%)	**9.86** (-20.5%)	7.95 -	22.98 (+189.0%)	8.52 (+7.1%)	**5.37** (-32.4%)
Rat_Zoe	8.74 -	24.31 (+178.0%)	8.86 (+1.3%)	**7.63** (-12.7%)	12.13 -	35.45 (+192.2%)	12.17 (+0.3%)	**10.86** (-10.5%)	9.80 -	26.94 (+175.0%)	9.99 (+1.9%)	**8.37** (-14.5%)
Office												
Hog_Alexis	4.32 -	8.47 (+96.4%)	4.07 (-5.6%)	**2.68** (-37.9%)	6.03 -	11.16 (+85.2%)	5.69 (-5.6%)	**4.47** (-25.8%)	29.84 -	63.42 (+112.6%)	29.04 (-2.7%)	**16.52** (-44.6%)
Robin_Antonina	13.53 -	68.40 (+405.5%)	14.33 (+5.9%)	**11.13** (-17.8%)	18.64 -	81.78 (+338.6%)	19.78 (+6.1%)	**15.84** (-15.0%)	3.16 -	14.90 (+372.2%)	3.32 (+5.3%)	**2.59** (-17.8%)
Mean	8.57 -	30.59 (+223.5%)	9.09 (+5.1%)	**6.83** (-22.4%)	12.39 -	42.61 (+217.5%)	12.76 (+1.8%)	**10.58** (-15.7%)	21.42 -	63.69 (+217.2%)	24.34 (+7.5%)	**15.83** (-25.5%)

Table 21: Zero-shot generalisation: point-forecast accuracy (MAE, RMSE, MAPE) for XGBoost and proposed deep learning models evaluated on held-out datasets belonging to categories seen during training, under the per-category scaling configuration. Models are applied without any fine-tuning on the target datasets. iTransformer consistently outperforms the XGBoost baseline by a large margin across all datasets and metrics (mean improvement MAE 22.4%, RMSE 15.7%, MAPE 25.5%). TiDE achieves results close to XGBoost but remains slightly worse on average (+5.1% MAE). TSMixer fails to generalise: it degrades dramatically relative to the baseline (+223.5% MAE on average), confirming that its performance does not transfer to unseen meters without per-dataset adaptation. Best results per metric are in bold.

reductions of 27% (MAE), 23% (RMSE), and 29% (MAPE) under per-category scaling, and 30%, 26%, and 34% respectively under per-dataset scaling. Crucially, it is the only model that improves on XGBoost on every single dataset in both experiments, confirming that its accuracy gains are not driven by a handful of favorable instances but are systematic. TiDE ranks second, delivering modest but consistent improvements over the baseline (mean MAE reduction of 3% per-category, 7% per-dataset). TSMixer presents a more nuanced picture and is discussed separately below. These findings are consistent with the results observed in the main body of this thesis, where iTransformer frequently achieves the best forecast accuracy as well, despite the different evaluation setting.

B.4.2 Scaling granularity strongly modulates TSMixer's performance

TSMixer shows a qualitative shift between the two experiments that is absent in the other models. Under per-dataset scaling (Experiment 1) it improves on XGBoost on 24 out of 27 datasets for MAE (mean +16%), broadly competitive with TiDE and in line with the general deep-learning trend. Under per-category scaling (Experiment 2), however, it degrades sharply, performing better on only 11 datasets and yielding a negative mean improvement of −10% for MAE. This behavior is

consistent with TSMixer's architectural design: the model mixes information along both the time axis and the feature (channel) axis within each input window. When a single scaler is fitted over an entire building category, the feature distributions of different meters are pooled together, and the cross-channel mixing amplifies the resulting distributional mismatch. Per-dataset scaling removes this confound, restoring each channel's statistics to a comparable range and allowing TSMixer's mixing mechanism to operate effectively. Neither iTransformer nor TiDE exhibit the same sensitivity, suggesting that their inductive biases interact differently with the normalization level.

B.4.3 Inference latency

Accuracy improvements come at a computational cost. Table 22 summarizes the mean single-sample inference latency measured on CPU and GPU for all four models.

Model	CPU (ms)	CPU slowdown	GPU (ms)	GPU slowdown
XGBoost	1.94	—	1.46	—
TSMixer	32.74	16.9×	33.19	22.7×
TiDE	33.46	17.2×	33.37	22.8×
iTransformer	31.32	16.1×	34.20	23.4×

Table 22: Mean single-sample inference latency.

The three deep learning models are roughly 17× slower than XGBoost on CPU and 23× slower on GPU. Notably, CPU and GPU latencies are nearly identical for all DL architectures (32-34 ms), and GPU inference is even marginally slower than CPU for iTransformer. This is expected given that all benchmarks are conducted at batch size 1, reflecting the production deployment scenario in which each inference call covers a single building: at this scale, the overhead of transferring data to the GPU device becomes the dominant cost, effectively negating any parallelism advantage. In a production setting, the absolute latency of 33 ms per model per call remains well within the internal constraints. Furthermore, while the 17-23× slowdown relative to XGBoost appears significant for isolated single-sample execution, serving a large portfolio of buildings simultaneously would naturally allow for request batching. This operational batching would fully leverage GPU parallelism, effectively amortizing the data transfer overhead and neutralizing the latency gap at scale.

Bibliography

[1] A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, and R. Yu, "Long-term Forecasting with TiDE: Time-series Dense Encoder." [Online]. Available: https://arxiv.org/abs/2304.08424

[2] Y. Liu *et al.*, "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting." [Online]. Available: https://arxiv.org/abs/2310.06625

[3] S.-A. Chen, C.-L. Li, N. Yoder, S. O. Arik, and T. Pfister, "TSMixer: An All-MLP Architecture for Time Series Forecasting." [Online]. Available: https://arxiv.org/abs/2303.06053

[4] M. Stocker, W. Małgorzewicz, M. Fontana, and S. B. Taieb, "A Gentle Introduction to Conformal Time Series Forecasting." [Online]. Available: https://arxiv.org/abs/2511.13608

[5] Y. Romano, E. Patterson, and E. J. Candès, "Conformalized Quantile Regression." [Online]. Available: https://arxiv.org/abs/1905.03222

[6] C. Xu and Y. Xie, "Conformal prediction for time series." [Online]. Available: https://arxiv.org/abs/2010.09107

[7] C. Xu and Y. Xie, "Sequential Predictive Conformal Inference for Time Series." [Online]. Available: https://arxiv.org/abs/2212.03463

[8] A. Auer, M. Gauch, D. Klotz, and S. Hochreiter, "Conformal Prediction for Time Series with Modern Hopfield Networks." [Online]. Available: https://arxiv.org/abs/2303.12783

[9] A. Cini, A. Jenkins, D. Mandic, C. Alippi, and F. M. Bianchi, "Relational Conformal Prediction for Correlated Time Series." [Online]. Available: https://arxiv.org/abs/2502.09443

[10] K. Stankeviciute, A. M. Alaa, and M. van der Schaar, "Conformal Time-series Forecasting," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., Curran Associates, Inc., 2021, pp. 6216–6228. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/312f1ba2a72318edaaa995a67835fad5-Paper.pdf

[11] R. Neglia, A. Cini, M. M. Bronstein, and F. M. Bianchi, "ResCP: Reservoir Conformal Prediction for Time Series Forecasting." [Online]. Available: https://arxiv.org/abs/2510.05060

[12] A. Vaswani *et al.*, "Attention Is All You Need." [Online]. Available: https://arxiv.org/abs/1706.03762

[13] C. Gallicchio and A. Micheli, "Echo State Property of Deep Reservoir Computing Networks," *Cognitive Computation*, vol. 9, no. 3, pp. 337–350, 2017, doi: 10.1007/s12559-017-9461-9.

[14] J. Hallberg Szabadváry, "Adaptive Conformal Inference for Multi-Step Ahead Time-Series Forecasting Online." [Online]. Available: https://arxiv.org/abs/2409.14792

[15] X. Wang and R. J. Hyndman, "Online conformal inference for multi-step time series forecasting." [Online]. Available: https://arxiv.org/abs/2410.13115

[16] H. Jaeger, "The "Echo State" Approach to Analysing and Training Recurrent Neural Networks," *GMD-Report 148, German National Research Institute for Computer Science*, p. , Jan. 2001.

[17] I. Farkas and P. Gergel, "Maximizing Memory Capacity of Echo State Networks with Orthogonalized Reservoirs," May 2017, p. . doi: 10.1109/IJCNN.2017.7966152.

[18] A. Rodan and P. Tino, "Minimum complexity echo state network. IEEE Trans Neural Netw," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 22, pp. 131–144, Nov. 2010, doi: 10.1109/TNN.2010.2089641.

[19] R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani, "Conformal prediction beyond exchangeability." [Online]. Available: https://arxiv.org/abs/2202.13415

[20] A. N. Angelopoulos, S. Bates, A. Fisch, L. Lei, and T. Schuster, "Conformal Risk Control." [Online]. Available: https://arxiv.org/abs/2208.02814

[21] I. Gibbs and E. Candès, "Adaptive Conformal Inference Under Distribution Shift." [Online]. Available: https://arxiv.org/abs/2106.00170

[22] C. Miller *et al.*, "The Building Data Genome Project 2, energy meter data from the ASHRAE Great Energy Predictor III competition," *Scientific Data*, vol. 7, p. 368, Oct. 2020.

[23] J. Herzen *et al.*, "Darts: User-Friendly Modern Machine Learning for Time Series," *Journal of Machine Learning Research*, vol. 23, no. 124, pp. 1–6, 2022, [Online]. Available: http://jmlr.org/papers/v23/21-1177.html

[24] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library." [Online]. Available: https://arxiv.org/abs/1912.01703

[25] L. Biewald, "Experiment Tracking with Weights and Biases." [Online]. Available: https://www.wandb.com/

[26] R. Kuprieiev *et al.*, "DVC: Data Version Control - Git for Data & Models." [Online]. Available: https://doi.org/10.5281/zenodo.5082912

[27] T. Ichinose, M. Yukawa, and R. L. G. Cavalcante, "Online Kernel-Based Quantile Regression Using Huberized Pinball Loss," in *2023 31st European Signal Processing Conference (EUSIPCO)*, 2023, pp. 1803–1807. doi: 10.23919/EUSIPCO58844.2023.10290008.