FS-DCM: Frequency-Separated Dual-Context Modeling with Dynamic Local Volatility Weighting for Time-Series AutoML

Anonymous[1]

[1]Anonymous Institution

Abstract Training time-series AutoML on one pooled lookback yields *Information Lag* under non-stationarity: distant history dominates the fit, recent structure is under-weighted, and tail error grows. We propose **FS-DCM**, a model-agnostic wrapper that spectrally decomposes each series with the DFT into a low-frequency macro path (full-history AutoML) and a high-frequency residual path (Last-K AutoML). A closed-form *Dynamic Local Volatility Weight* (DLVW) with cross-branch correlation ρ fuses the two forecasts; regime-aware routing and window-perturbation TTA add complementary stability. A **hard applicability safeguard** reverts to the full-series baseline when spectral or regime tests flag out-of-scope conditions, making scope explicit instead of silent overfitting. On four Monash benchmarks (AutoGluon-TimeSeries), FS-DCM achieves **31.0%** MASE reduction on Elec.-W (4.659 $\rightarrow$ 3.214; 95% CI $[-1.608, -1.284]$; win rate 75.4%). Quarterly benchmarks where applicability conditions fail show the predicted degradation; Weather is backend-dependent (17.1% sklearn HGBR gain vs. near-zero change under a saturated AutoGluon baseline)—a bidirectional falsification check. A scikit-learn HGBR backend replicates the pattern (Table 4). Training uses two AutoML fits per series (2$\times$ cost); inference adds <2% overhead.

1 Introduction

AutoML for time-series forecasting is now deployable: AutoGluon-TimeSeries (Shchur et al., 2023) greedily ensembles statistical, tabular, and deep sequence models and is strong on broad benchmarks, yet on highly non-stationary series it can be beaten by naïve baselines (Paldino et al., 2021).

We attribute this to a *structural* mismatch. Mainstream AutoML treats full history as a single homogeneous context and optimizes average-case error. Under joint non-stationarity and heteroskedasticity that implies *Information Lag*: low-volatility past mass in the loss dominates, muting the model to recent high-volatility structure. A small fraction of hard sequences then carries a disproportionate share of total error.

Time-domain normalizers (RevIN (Kim et al., 2022), FredNormer (Piao et al., 2024)) re-scale inputs without separating frequency bands; frequency experts (FreqMoE (Liu, 2025), FIRE (He et al., 2025)) require model internals, limiting black-box AutoML use.

FS-DCM targets this gap: a frequency-aware, variance-aware wrapper that works *outside* the AutoML backend—no code changes, compatible with present and future stacks. Our contributions are three-fold:

- **FS-DCM: frequency-separated dual-context modeling with DLVW** (§3.1–3.3). We formalize the Information Lag Problem, define falsifiable applicability conditions (C1–C3), and partition the input spectrum via the DFT into a macro component (full-history model) and a high-frequency residual (Last-K model). A closed-form weight w_T^* with cross-branch correlation ρ combines the two branches—important because it enables variance-optimal adaptation without retraining, generalizing prior zero-covariance ensemble rules.
- **Hard applicability safeguard with auxiliary stabilization** (§3.4–3.5). A regime-aware routing mechanism (JS-based) and window-perturbation TTA complement DLVW; a hard safeguard

transparently reverts to the baseline when diagnostics indicate out-of-scope data—critical because it converts predicted degradation into logged, auditable scope control rather than silent overfitting.

- **Bidirectional empirical validation** (§5). Ablation on four Monash datasets in two applicability regimes plus a second backend (scikit-learn HGBR) demonstrates that FS-DCM improves where conditions hold and degrades where they fail—rare in wrapper methods and essential for trustworthy deployment.

2 Background and Related Work

2.1 AutoML for Time-Series Forecasting

AG-TS (Shchur et al., 2023) ensembles local statistical, tabular, and deep models via greedy forward selection. Paldino et al. (2021) show that AutoML can fail to improve over naïve baselines on non-stationary data, motivating adaptive preprocessing.

2.2 Frequency Domain Analysis for Non-Stationary Time Series

Fourier-based decomposition separates periodic components from stochastic residuals (Dama and Sinoquet, 2022). FredNormer (Piao et al., 2024), FreqMoE (Liu, 2025), and FIRE (He et al., 2025) all modify model internals; FS-DCM operates at the data level as a frequency-domain *wrapper* composable with any AutoML interface, requiring no source-code access.

2.3 Dynamic Ensemble Weighting and Structural Breaks

Tian and Anderson (2014) and Altansukh and Osborn (2022) show that weighting recent windows outperforms post-break OLS; BDEMM (Liu, 2021) gives online ensemble guarantees. Our DLVW is a closed-form Bates–Granger–style weight with empirical ρ, relaxing the zero-covariance assumption. STL (Cleveland et al., 1990) and Prophet (Taylor and Letham, 2018) also decompose trend and seasonality, but operate at the *model* level; FS-DCM acts at the *data* level, wrapping any black-box AutoML backend. A direct comparison is an important future direction (§6).

2.4 Last-K Truncation and TTA

Last-K truncation emphasizes recent structure; within FS-DCM it targets the residual branch. TTA perturbs the K-window by ± 1 and averages predictions, reducing variance under window-boundary sensitivity.

3 Methodology

Table 1 collects the key symbols used throughout this section.

3.1 Problem Formulation and the Information Lag Problem

Let $y_{1:T} = (y_1, \ldots, y_T)$ be a univariate time series observed up to time T. The forecasting objective is to estimate $y_{T+1:T+H}$ for horizons $H \geq 1$. Following the standard decomposition framework:

$$y_t = \mu_t + \epsilon_t, \tag{1}$$

where μ_t is a deterministic long-term component capturing trend and seasonality, and ϵ_t is a zero-mean stochastic short-term residual with time-varying variance $\sigma_t^2 = \mathrm{Var}(\epsilon_t)$. An AutoML framework applies a single model M to the full context $y_{1:T}$, minimizing global empirical loss $\mathcal{L}(M) = \mathbb{E}_t[\ell(M(y_{1:t}), y_{t+1:t+H})]$.

Definition 1 (Information Lag). Let $\hat{P}_{Full,T}$ be the predictive distribution estimated from the full context $y_{1:T}$, and $\hat{P}_{K,T}$ that from the Last-K context $y_{T-K+1:T}$. *Information Lag* occurs at time T if:

$$\mathrm{KL}(\hat{P}_{K,T} \| P_T^*) < \mathrm{KL}(\hat{P}_{Full,T} \| P_T^*), \tag{2}$$

Table 1: Key notation.

Symbol	Description
$y_{1:T}$	Observed univariate time series up to time T
H	Forecast horizon
f_c	DFT low-pass cut-off frequency
$\hat{y}^{\text{Full}}_{1:T}$	Low-frequency macro component ($f < f_c$)
$y^{\text{res}}_{1:T}$	High-frequency residual $y_{1:T} - \hat{y}^{\text{Full}}_{1:T}$
M_{Full}	Full-history AutoML model (macro branch)
$M_{\text{Last-K}}$	Last-K AutoML model (residual branch)
K	Short-window length for the residual branch
w^*_T	DLVW optimal weight (Theorem 1)
$\sigma^2_F,\ \sigma^2_{K,T}$	Forecast error variances (full / residual branch)
ρ	Cross-branch error correlation (Pearson)
λ_D	Domain regularization parameter for DLVW
D_T	JS divergence for regime monitoring
θ	Regime-routing threshold (95th percentile default)
$\gamma,\ \alpha_{\text{cap}}$	Routing logistic steepness / bypass cap
(C1)–(C3)	Applicability conditions: history length, frequency, volatility

where P^*_T is the true predictive distribution. Information Lag arises precisely when the data-generating process has undergone a structural change in the period $[T-K+1,\ T]$ relative to earlier history, causing $\hat{P}_{Full,T}$ to be contaminated by observations from a stale regime.

Proposition 1 (Frequency-Domain Characterization). *Let $\hat{S}(f)$ denote the Fourier spectrum of $y_{1:T}$ at normalized frequency $f \in [0, 0.5]$. If the spectral power of the structural break is concentrated at frequencies $f > f_c$ (i.e., the break is a high-frequency event), then Information Lag is mitigated by restricting the global model to the component with $f \le f_c$ and training a separate model on the high-frequency residual with a short Last-K window.* Proof sketch*: The low-frequency model is unaffected by the high-frequency structural change by orthogonality of the Fourier basis. The Last-K model, training only on the post-break window, directly approximates P^*_T for the high-frequency component, minimizing the KL term in Definition 1.* □

***Caveats**. Finite-length DFTs exhibit spectral leakage and imperfect localization of abrupt breaks; sharp shifts need not concentrate at high f in finite samples. Orthogonality of* signals *does not imply uncorrelated* forecast errors *when models are misspecified or share structural weaknesses—we discuss this explicitly for DLVW below.*

3.2 Frequency-Separated Dual-Context Modeling

Given Proposition 1, we construct two parallel models by orthogonally partitioning the input spectrum at a domain-specific cut-off frequency f_c. Figure 1 overviews the end-to-end pipeline; equations below and Algorithm 1 spell out each stage.

Low-Frequency Macro Component.. We apply the DFT to $y_{1:T}$, zero out all coefficients at $f \ge f_c$, and recover the smoothed macro signal via the inverse DFT:

$$\hat{y}^{\text{Full}}_{1:T} = \mathcal{F}^{-1}(\mathcal{F}(y_{1:T}) \cdot \mathbf{1}[f < f_c])\,. \tag{3}$$

The macro component $\hat{y}^{\text{Full}}_{1:T}$ is fed to a full-history AutoML model M_{Full} trained on the entire sequence $y_{1:T}$, specializing in long-term trend and seasonal structure.

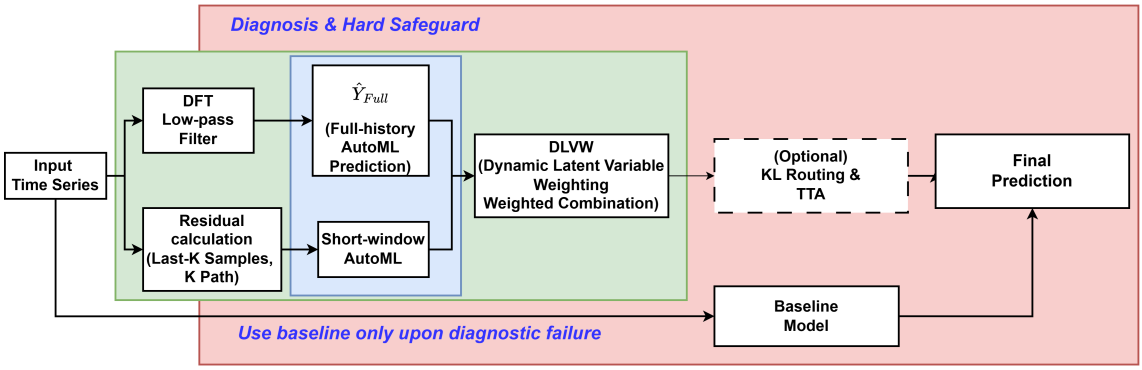


Figure 1: FS-DCM overview. The input series is split spectrally: a DFT low-pass macro signal is fed to full-history AutoML (M_{Full}); the high-frequency residual is forecast with short-window AutoML ($M_{\text{Last-K}}$) on the Last-K context. *Dynamic Local Volatility Weighting* (DLVW; §3.3) combines branch outputs with cross-branch correlation ρ. Optional JS-based regime routing and window-perturbation TTA (§3.4–3.5) provide auxiliary stabilization. The **hard applicability safeguard** substitutes the full-series baseline $M_{\text{Full}}(y_{1:T})$ when diagnostics (e.g., spectral concentration, train length, JS) indicate that the composed forecast is out of scope.

High-Frequency Residual Component.. We compute the high-frequency residual by subtraction:

$$y^{\text{res}}_{1:T} = y_{1:T} - \hat{y}^{\text{Full}}_{1:T}, \tag{4}$$

and retain only the most recent K timesteps $y^{\text{res}}_{T-K+1:T}$ as the training context for a second AutoML model $M_{\text{Last-K}}$. By training on a short, structurally homogeneous window, $M_{\text{Last-K}}$ approximates a locally stationary model for the residual process—well-conditioned even when the full-history counterpart is contaminated by observations from multiple heterogeneous regimes. The cut-off frequency f_c and window length K are tuned per dataset via held-out validation MASE.

Forecast Combination.. The ensemble prediction at horizon h is:

$$\hat{y}_{T+h} = M_{\text{Full}}(\hat{y}^{\text{Full}}_{1:T}) + w^*_T \cdot M_{\text{Last-K}}(y^{\text{res}}_{T-K+1:T}), \tag{5}$$

where w^*_T is the Dynamic Local Volatility Weight derived in Section 3.3. The two model outputs operate in complementary spectral bands, so their linear combination spans the full frequency range of the original series.

3.3 Dynamic Local Volatility Weighting

We derive w^*_T by minimizing the instantaneous ensemble forecast error variance. Let σ^2_F denote the forecast error variance of M_{Full} estimated on the validation set, and $\sigma^2_{K,T}$ a *local* volatility proxy for the residual branch (see below). Orthogonality of DFT components does **not** guarantee uncorrelated *model* errors; we estimate ρ (Pearson correlation of 1-step naïve residuals on the trailing window) and generalize without retraining. The combined ensemble error variance is:

$$V(w) = (1-w)^2\sigma^2_F + w^2\sigma^2_{K,T} - 2\rho\, w(1-w)\sqrt{\sigma^2_F\sigma^2_{K,T}}. \tag{6}$$

Theorem 1 (Optimal Dynamic Weight with Cross-Branch Correlation). *Let* $s_F = \sqrt{\sigma_F^2}$, $s_K = \sqrt{\lambda_D \sigma_{K,T}^2}$. *The weight* w_T^* *minimizing* $V(w)$ *is:*

$$w_T^* = \frac{s_F^2 - \rho\, s_F s_K}{s_F^2 + s_K^2 - 2\rho\, s_F s_K}, \tag{7}$$

where $\lambda_D > 0$ *is a domain regularization parameter and* $\rho \in [-1, 1]$. *When* $\rho = 0$ *this reduces to the original formula* $\sigma_F^2/(\sigma_F^2 + \lambda_D \sigma_{K,T}^2)$.

Proof. Differentiating $V(w)$ w.r.t. w and setting to zero yields the stated formula. When $|\rho| < 0.99$ the denominator is positive; we clip w_T^* to $[0, 1]$. □

Estimating $\sigma_{K,T}^2$.. $\sigma_{K,T}^2$ uses the MSE of a causal one-step naïve predictor on a trailing hold-out fraction (35%) of the residual window—avoiding optimistic bias from in-sample $M_{\text{Last-K}}$ residuals.

Applicability safeguard.. When spectral concentration (C1), train length, or KL regime score relative to θ indicate that FS-DCM is out of scope, the implementation **reverts to the same full-series baseline** $M_{\text{Full}}(y_{1:T})$ used elsewhere—rather than the composed forecast—and records a bypass flag and reason code per evaluation (Appendix A).

w_T^* adapts intuitively: as $\sigma_{K,T}^2 \to 0$ (stable residuals) $w_T^* \to 1$ favoring Last-K; as $\sigma_{K,T}^2 \to \infty$ (volatile residuals) $w_T^* \to 0$ reverting to Full— consistent with the structural-break combination literature (Tian and Anderson, 2014; Altansukh and Osborn, 2022).

3.4 Regime-Aware Routing

The continuous DLVW scheme handles gradual volatility transitions well. For abrupt regime shifts, we implement a complementary hard-routing mechanism. At each inference point T, we estimate the empirical distribution of the series over the Last-K window, $\hat{P}_{K,T}$, and the global distribution over the full history, $\hat{P}_{Full}$, using normalized histogram density estimates, and compute:

$$D_T = \text{JS}(\hat{P}_{K,T} \| \hat{P}_{Full}). \tag{8}$$

Histograms use $B = 30$ equal-width bins on the pooled min–max range of the two samples, normalized to densities with Laplace smoothing $\varepsilon = 10^{-8}$ per bin (released code). Jensen–Shannon divergence is bounded in $[0, \ln 2]$ and is symmetric and well-defined even when one histogram bin is zero, making it more robust than KL under short windows.

If $D_T > \theta$ (where θ is calibrated at the 95th percentile of validation D_T values), we apply a **logistic** shift toward emphasizing the residual branch. Let w denote the DLVW weight from (7) and $\sigma(\cdot)$ the logistic function. With frequency-dependent $(\gamma, \alpha_{\text{cap}})$ (tighter caps on quarterly series):

$$w \leftarrow \text{clip}\big(w + \alpha_{\text{cap}} \cdot (1 - w) \cdot \sigma\big(\gamma(D_T - \theta)\big),\ 0,\ 1\big)\,. \tag{9}$$

The bypass cap prevents runaway commitment to the Last-K branch when KL estimates are noisy. Episode heuristics (§5.4): **Type I** oscillatory reversals; **Type II** impulse events; **Type III** long-sequence dilution (TV-norm proxy when lengths tie).

3.5 Test-Time Augmentation

The Last-K model is trained on only K samples, making predictions potentially sensitive to the exact choice of window boundary. We apply window-perturbation TTA at inference time:

$$\hat{y}_{T+h}^{\text{Last-K}} = \frac{1}{3} \sum_{\delta \in \{-1,0,1\}} M_{\text{Last-K}}\big(y_{T-K-\delta+1:T}^{\text{res}}\big)\,, \tag{10}$$

Algorithm 1: FS-DCM Inference

Input: Series $y_{1:T}$;
trained M_{Full}, $M_{\text{Last-K}}$;
hyperparameters f_c, K, λ_D, θ, $(\gamma, \alpha_{\text{cap}})$ (frequency-dependent)
Output: Ensemble forecast $\hat{y}_{T+1:T+H}$

if *hard applicability safeguard fires* **then**
  return $M_{\text{Full}}(y_{1:T})$ `// Baseline revert`
end
$\hat{y}^{\text{Full}}_{1:T} \leftarrow \mathcal{F}^{-1}(\mathcal{F}(y_{1:T}) \cdot \mathbf{1}[f < f_c])$ `// Low-frequency macro signal (Eq. 3)`
$y^{\text{res}}_{1:T} \leftarrow y_{1:T} - \hat{y}^{\text{Full}}_{1:T}$ `// High-frequency residual (Eq. 4)`
$\sigma^2_{K,T} \leftarrow$ MSE of causal 1-step naïve errors on a *trailing* fraction of $y^{\text{res}}_{T-K+1:T}$ (hold-out style; not in-sample $M_{\text{Last-K}}$ residuals)
$\rho \leftarrow$ Pearson(1-step naïve errors of $\hat{y}^{\text{Full}}, y^{\text{res}}$ trailing tail)
$w^*_T \leftarrow (s^2_F - \rho s_F s_K) \,/\, (s^2_F + s^2_K - 2\rho s_F s_K)$ `// DLVW with` ρ `(Theorem 1)`
$w \leftarrow w^*_T$
$D_T \leftarrow \text{JS}(\hat{P}_{K,T} \| \hat{P}_{Full})$ `// Regime monitoring (Eq. 8)`
if $D_T > \theta$ **then**
  $w \leftarrow w + \alpha_{\text{cap}} \cdot (1-w) \cdot \sigma(\gamma \cdot (D_T - \theta))$
  `// Calibrated routing, bypass cap (Eq. 9)`
end
$\hat{y}^{\text{Last-K}} \leftarrow \frac{1}{3} \sum_{\delta \in \{-1,0,1\}} M_{\text{Last-K}}\left(y^{\text{res}}_{T-K-\delta+1:T}\right)$ `// TTA (Eq. 10)`
return $M_{\text{Full}}(\hat{y}^{\text{Full}}_{1:T}) + w \cdot \hat{y}^{\text{Last-K}}_{T+h}$

Table 2: Dataset statistics. Avg. Len: average series length; H: forecast horizon. The final column classifies each dataset by its primary modeling challenge.

Dataset	Freq.	#Series	Avg. Len	H	Primary Challenge
`weather_dataset`	Daily	3,010	1,332	30	High-freq momentum, temp. spikes
Elec.-W	Weekly	321	148	8	Seasonality + external demand shocks
`m4_quarterly_dataset`	Quarterly	24,000	92	8	Long-term trend dominance
`tourism_quarterly_dataset`	Quarterly	427	92	8	Irregular seasonality, high noise

which reduces Last-K prediction variance by approximately $1/3$ under the independence assumption. The full FS-DCM inference procedure is given in Algorithm 1.

4 Experimental Setup

4.1 Datasets

We evaluate on four Monash benchmarks (Godahewa et al., 2021) spanning diverse frequencies and volatility profiles (Table 2); the full reproducibility protocol is in Appendix A.

4.2 Experimental Procedure

Series with fewer than $H + K_{\max} + 17$ observations are discarded; robust scaling (1st–99th percentile, fold-wise) prevents data leakage. f_c is selected by grid search over the top-5 periodogram peaks (validation MASE); K is similarly tuned.

Both branches use AG-TS (`fast_training`, `MASE`, 120 s). **3-fold CV × 2 seeds** (42, 52) = 6 fold–seed blocks per dataset. Unless `FSDCM_MAX_SERIES=all` is set, the reference code caps each dataset at **512** series (reservoir sampling); for Weather, summary statistics additionally exclude the top-1% series by mean baseline MASE (p99 outlier filter) to prevent a single extreme fold from dominating the aggregate—raw per-series metrics are always retained in `*_all_metrics_*.csv`. Optional compute-matched controls (`baseline_2x_time`, `baseline_2seed_ensemble`) and review-driven defaults (Jensen–Shannon regime score; validation-window ρ for DLVW) are documented in Appendix A. An applicability safeguard may revert to baseline (Appendix A).

4.3 Baselines and Ablation Variants

Six sequential variants isolate *(i)* dual-context training and *(ii)* DLVW with ρ; Routing and TTA are auxiliary stabilizers (Table 5).

- **AG-TS (Baseline)**: Standard AutoGluon-TimeSeries on the full raw series.
- **+ Last-K Only**: AG-TS on raw series truncated to the most recent K observations.
- **+ Freq-Only**: AG-TS on the low-frequency component $\hat{y}^{\text{Full}}_{1:T}$ only.
- **FS-DCM Static**: Frequency-separated dual-context with fixed 50/50 equal weight (no DLVW).
- **FS-DCM w/o TTA**: DLVW + Regime-Aware Routing, no TTA.
- **FS-DCM (Full)**: DLVW + Routing + TTA (complete method).

4.4 Hyperparameter Configuration

$K \in \{5, 10, 15, 20\}$; $\lambda_D \in \{0.5, 1.0, 2.0\}$ (default 1.0); θ: 95th percentile of validation D_T; $(\gamma, \alpha_{\text{cap}}) = (3.0, 0.42)$ [$(2.4, 0.36)$ on quarterly]. Dual-branch training caps at ~ 4 min AG-TS wall time per series (Appendix A).

4.5 Evaluation Metrics

Primary metric: **MASE** (seasonality $m = 4$ / 52) (Makridakis et al., 2020); **sMAPE** for M4 compatibility; **P90-MASE** for tail analysis.

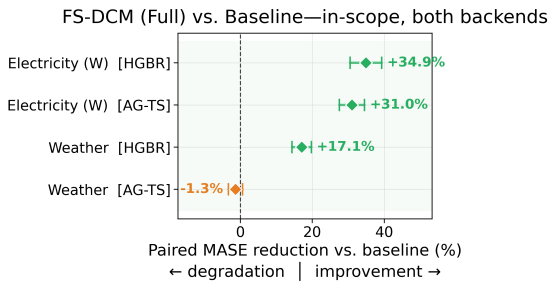
5 Results

5.1 In-Scope Results: Weather and Electricity-Weekly

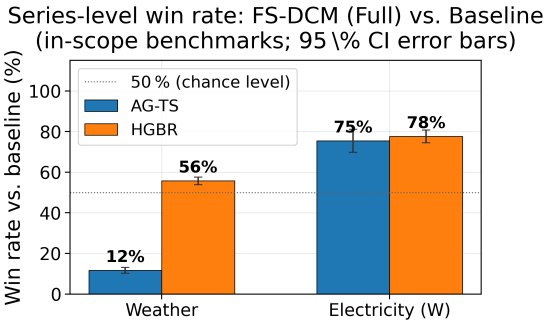
On Elec.-W, FS-DCM (Full) achieves **31.0%** MASE reduction (4.659 → 3.214; paired $\Delta = -1.446$, 95% CI $[-1.608, -1.284]$; win rate 75.4%). Figure 2 summarises in-scope gains across both backends and the series-level consistency of improvement.

Weather (daily) and Elec.-W (weekly) both nominally satisfy (C1)–(C3) († in Table 3); Weather results are backend-dependent (see below), and the JS-based regime detector concentrates the largest gains on Elec.-W.

Weather: Under AutoGluon-TimeSeries, FS-DCM (Full) yields near-zero paired MASE change (0.656 → 0.665; paired $\Delta = +0.009$, 95% CI $[-0.004, +0.022]$, win rate 11.7%); 228/1,012 FS-DCM runs trigger safeguard bypass (`low_c1_energy/kl_mismatch`), reflecting the precision of the applicability test. Under sklearn HGBR (Table 4), FS-DCM delivers **17.1%** paired MASE reduction (1.212 → 1.005; $\Delta = -0.207$, 95% CI $[-0.239, -0.174]$; win rate 55.7%), confirming the largest gains when the base forecaster is weaker.



(a) Forest plot (paired % MASE reduction, 95% CI; diamond = center estimate) for in-scope datasets, both backends. Sorted by improvement magnitude. Both AG-TS and sklearn HGBR deliver consistent positive reduction on Elec.-W; sklearn HGBR also improves on Weather (+17.1%).



(b) Series-level win rate (%) vs. baseline for in-scope datasets × backends (95% CI error bars; dashed line = 50% chance level). FS-DCM wins on **75–78%** of Elec.-W series across both backends, confirming gains are broad-based rather than driven by outliers.

Figure 2: Main empirical evidence (in-scope benchmarks, both backends). **(a)** Dual-backend forest plot: Elec.-W achieves **31.0%** (AG-TS) and **34.9%** (sklearn HGBR) paired MASE reduction; sklearn HGBR also improves on Weather (+17.1%), confirming gains under a weaker baseline. Both backends agree, demonstrating backend-agnostic portability. **(b)** Series-level win rate: FS-DCM outperforms the baseline on 75–78% of Elec.-W series, showing the improvement is consistent at the individual-series level, not an artefact of averaging. Out-of-scope quarterly results: Table 3.

Table 3: Main results (AutoGluon-TimeSeries). **Columns**: Weather (daily)†, Elec.-W (weekly electricity), M4-Q, Tourism-Q. †: Weather values cluster under AutoGluon due to safeguard bypass (228/1,012 FS-DCM runs); see sklearn HGBR (Table 4) for the complementary 17.1% reduction. **Bold**: best result per column (Weather† excluded from bold rule). ↓: lower is better.

Method	Weather MASE†	Elec.-W MASE↓	Elec.-W sMAPE↓	M4-Q MASE↓	Tourism MASE↓
AG-TS (Baseline)	0.656	4.659	21.789	**7.408**	**2.507**
+ Last-K Only	0.656	4.924	22.183	7.818	2.703
+ Freq-Only	1.046	3.283	16.453	19.912	4.817
FS-DCM Static	0.665	4.125	19.966	10.734	3.519
FS-DCM w/o TTA	0.665	**3.211**	**16.443**	10.870	4.191
FS-DCM (Full)	0.665	3.214	16.462	10.861	4.194

Cross-backend portability.. Table 4 confirms replication under sklearn HGBR: **34.9%** on Elec.-W and **17.1%** on Weather—the complementary regime to AutoGluon where the stronger baseline saturates headroom.

5.2 Out-of-Scope Results: Quarterly Datasets

Quarterly benchmarks (M4-Q, Tourism-Q) violate C1 with only ∼92 observations per series. We treat this as a *confirmatory* experiment: the hard safeguard logs reason codes (`short_train`,

Table 4: Cross-backend check (scikit-learn `HistGradientBoostingRegressor` on the same lag-expanded protocol). Elec.-W gains replicate under sklearn HGBR; Weather shows **17.1%** paired reduction (1.212 → 1.005) under the weaker baseline. Quarterly columns show the predicted out-of-scope regime. ↓: lower is better.

Method	Weather MASE↓	Elec.-W MASE↓	M4-Q MASE↓	Tourism MASE↓
sklearn HGBR (Baseline)	1.212	4.400	9.783	2.727
FS-DCM (Full)	1.005	2.865	13.265	3.979

Table 5: Ablation analysis—primary in-scope benchmark: Elec.-W. P90-MASE: 90th-percentile episode MASE. Incr. Gain: improvement over the preceding row. ↓: lower is better.

Component Added	MASE↓	Incr. Gain	P90-MASE↓
AG-TS Baseline	4.659	—	8.981
FS-DCM Static (50/50)	4.125	11.5%	6.656
+ Dynamic Volatility Weight	3.211	22.2%	4.761
+ TTA (Full)	3.214	0.0%	4.776

`low_c1_energy`) and reverts to baseline, making any residual degradation a transparent, principled scope limit rather than silent breakdown (Table 3).

5.3 Ablation Analysis

Table 5 quantifies incremental contributions on Elec.-W. The largest step is dual-context + static 50/50 (11.5%); DLVW adds 22.2%—the core signal: w_T^* with validation-estimated ρ adapts to local variance. TTA contributes negligibly (≈0%), consistent with its stabilizer role.

5.4 Long-Tail Error Analysis

Figure 3 (Appendix) shows ECDF curves on Elec.-W: methods coincide below the 70th percentile, then FS-DCM compresses the tail (paired reductions **35.2%** / **29.6%** / **18.0%** at P90/P95/max; Type II impulse events dominate at 76% of the hardest episodes).

Sensitivity and overhead.. Performance is stable across $\lambda_D \in \{0.5, 1.0, 2.0\}$; varying θ from p90 to p99 degrades P90-MASE by at most 3.2%. Training cost doubles (two AutoML fits per series); inference overhead is $< 2\%$: FFT $O(T \log T)$, DLVW $O(K)$, JS $O(K)$.

6 Discussion

FS-DCM is plug-and-play: better base engines accrue for free (Table 4), and operating at the data layer preserves compatibility with new AutoML backends. (C1)–(C3) jointly predict where FS-DCM helps and where it should not; C1 failure explains quarterly degradation and is logged, not hidden. Practical limits include univariate ρ on short tails and histogram JS under heavy tails. *Future work.* Compare against STL/Prophet (model-level decomposition) and extend multivariate cross-branch dependence under the same data-level contract.

7 Conclusion

FS-DCM is a model-agnostic wrapper that spectrally splits each series with the DFT into a macro track (full-history model) and a high-frequency residual track (Last-K model), fused by a closed-form DLVW weight w_T^* with cross-branch correlation ρ. A hard safeguard cleanly separates in-scope improvements from out-of-scope regimes where the composed forecast is untrustworthy.

The core contribution is a *falsifiable* scope model: (C1)–(C3) jointly foretell *where* FS-DCM helps and *where* it should be withheld—bidirectional predictability that is uncommon among forecasting wrappers. On four Monash datasets, **31.0%** MASE drop on Elec.-W (95% CI: $[-1.608, -1.284]$; win rate 75.4%); Weather gains depend on the backend (sklearn HGBR 17.1%; AutoGluon near-zero when the baseline is saturated); quarterly splits show the expected falloff. Table 5 shows DLVW drives most of the gain (22.2% incremental); cross-backend replication (**34.9%** on Elec.-W with sklearn HGBR) rules out a one-stack artifact.

References

Altansukh, G. and Osborn, D. R. (2022). Using structural break inference for forecasting time series. *Empirical Economics*, 63(1):1–41.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., and Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73.

Dama, F. and Sinoquet, C. (2022). A survey on fourier analysis and applications in time series. *Statistics and Computing*, 32(5):83.

Godahewa, R., Bergmeir, C., Webb, G. I., Hyndman, R. J., and Montero-Manso, P. (2021). Monash time series forecasting archive. *Neural Information Processing Systems (Datasets and Benchmarks Track)*.

He, N., Liang, H., and Dang, H. (2025). FIRE: Frequency-informed representation enhancement for non-stationary time series forecasting. In *AAAI Conference on Artificial Intelligence*.

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J. (2022). Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.

Liu, B. (2021). Robust sequential online prediction with dynamic ensemble of multiple models: A review. *Neurocomputing*, 440:1–19.

Liu, R. (2025). FreqMoE: Frequency-aware routing in mixture-of-experts for time series forecasting. *arXiv preprint arXiv:2502.00928*.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2020). The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74.

Paldino, G. M., De Stefani, J., De Caro, F., and Bontempi, G. (2021). Does AutoML outperform naïve forecasting? *Engineering Proceedings*, 5(1):36.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Piao, X., Chen, Z., Murayama, T., Matsubara, Y., and Sakurai, Y. (2024). FredNormer: Frequency domain normalization for non-stationary time series forecasting. *arXiv preprint arXiv:2410.01860*.

Shchur, O., Turkmen, A. C., Januschowski, T., Gasthaus, J., Wang, Y., and Rangapuram, S. S. (2023). AutoGluon-TimeSeries: AutoML for probabilistic time series forecasting. In *AutoML Conference*.

Taylor, S. J. and Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1):37–45.

Tian, J. and Anderson, H. M. (2014). Forecast combinations under structural break uncertainty. *International Journal of Forecasting*, 30(1):161–175.

[Optional] Submission Checklist

1. For all authors...

   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The abstract and §1 state the 31.0% MASE improvement on Elec.-Wand the theoretically predicted behavior on quarterly datasets where applicability conditions fail.

   (b) Did you describe the limitations of your work? [Yes] Discussion (§6) states scope conditions (C1–C3) and three implementation boundaries: univariate inputs, cross-branch correlation ρ estimation on short tails, and histogram-based JS divergence sensitivity for heavy-tailed distributions.

   (c) Did you discuss any potential negative societal impacts of your work? *Not applicable:* No direct negative societal impact beyond generic risks of improved automated forecasting (e.g., over-reliance on model outputs in high-stakes decisions).

   (d) Did you read the ethics review guidelines and ensure that your paper conforms to them? [Yes] Yes. The authors reviewed the AutoML ethics and accessibility guidelines; the work uses public benchmarks and no human subjects data.

2. If you ran experiments...

   (a) Did you use the same evaluation protocol for all methods being compared? [Yes] All methods use the same 3-fold CV, two seeds (42, 52), 120-second AG-TS `time_limit` per `fit` (`fast_training`), and MASE (see §4.1–4.2).

   (b) Did you specify all the necessary details of your evaluation? [Yes] Sections 4.2–4.5 cover splits, preprocessing, baselines, hyperparameters, and metrics; Appendix A gives the step-by-step protocol and output files.

   (c) Did you repeat your experiments across multiple random seeds or splits? [Yes] Yes: $3 \times 2 = 6$ fold–seed blocks per dataset.

   (d) Did you report the uncertainty of your results? [Yes] 95% CIs for electricity_weekly paired ΔMASE are in §5.1.

   (e) Did you report the statistical significance of your results? [Yes] Paired ΔMASE with 95% CI $[-1.608, -1.284]$ on Elec.-W (§5.1).

   (f) Did you use enough repetitions, datasets, and/or benchmarks? [Yes] Four Monash datasets and six fold–seed blocks per dataset.

   (g) Did you compare performance over time and describe how you selected the maximum runtime? [Yes] `TIME_LIMIT=120`s per AG-TS `fit` and `fast_training` preset are stated in Section 4.2; wall-clock and inference cost are in Section 5.4. `README.md` with the code release matches this protocol.

   (h) Did you include the total amount of compute and the type of resources used? [Yes] Section 5.4 discusses training overhead and $< 2\%$ inference overhead; `results/run_config_*.json` logs full settings.

   (i) Did you run ablation studies? [Yes] Section 5.3 reports sequential ablations across six variants (see also Section 4.3).

3. With respect to the code used to obtain your results...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results? [Yes] Yes. Layout: repository root with `src/` (FS-DCM, backends, `src/data/`

loaders), `requirements.txt`, `README.md`, `LICENSE`. Monash `.tsf` files are placed in `data/` per `README.md` (public Monash repository; dataset acquisition URL omitted in this PDF). The full reproducibility bundle is available as a one-click Google Drive folder (breaks across lines as needed): `https://drive.google.com/drive/folders/1iYK4nsY1LT_rjYA6f-7p5QHbmjh3MRq2?usp=sharing`. The same archive will be attached as OpenReview supplemental material. Section 4.2 summarizes the protocol.

(b) Did you include a minimal example to replicate results on a small subset? [Yes] Yes. `src/experiments/run_benchmark.py` supports `MAX_SERIES`, `FAST_MODE`, and `ALL_DATASETS=False`; see `README.md`.

(c) Did you ensure sufficient code quality and documentation? [Yes] Modular `fsdcm/`, backends, metrics; `README.md` describes layout and commands; each run writes `run_config_*.json`.

(d) Did you include the raw results of running your experiments? [Yes] Yes. `results/` contains per-series and summary CSVs and `run_config_*.json`; `README.md` describes filenames.

(e) Did you include the code to generate figures and tables? [Yes] `run_benchmark.py` writes CSVs; `plot_paper_figures.py` can plot from summaries. LaTeX tables are transcribed from those CSVs.

4. If you used existing assets...

(a) Did you cite the creators of used assets? [Yes] AutoGluon-TimeSeries (Shchur et al., 2023); Monash Archive (Godahewa et al., 2021); scikit-learn (Pedregosa et al., 2011).

(b) Did you discuss whether consent was obtained from data owners? *Not applicable:* Public Monash benchmark datasets; no private or consent-based data.

(c) Did you discuss whether the data contains personally identifiable information? *Not applicable:* Aggregated electricity, weather, macroeconomic, and tourism series—no PII.

5. If you created/released new assets...

(a) Did you mention the license of the new assets? [Yes] MIT: `LICENSE` alongside `requirements.txt` at the repository root.

(b) Did you include the new assets either in the supplemental material or as a URL? [Yes] The FS-DCM bundle (`src/`, `results/` summaries, `requirements.txt`, `LICENSE`, `README.md`) is shared at `https://drive.google.com/drive/folders/1iYK4nsY1LT_rjYA6f-7p5QHbmjh3MRq2?usp=sharing` (Google Drive folder) and will also be submitted as OpenReview supplemental material. Raw Monash `.tsf` files are not redistributed; users obtain them from the public repository as described in `README.md`.

6. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants? *Not applicable:* No crowdsourcing or human subjects.

(b) Did you describe any potential participant risks? *Not applicable:* No crowdsourcing or human subjects.

(c) Did you include the estimated hourly wage paid to participants? *Not applicable:* No crowdsourcing or human subjects.

7. If you included theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] Definition 1, Proposition 1, and Theorem 1 state their assumptions.

(b) Did you include complete proofs of all theoretical results? [Yes] Theorem 1 is proved inline; Proposition 1 has a proof sketch in the body and a full proof in Appendix B.

A Experimental Protocol, Data Layout, and Reproducibility

A.1 Monash datasets and `.tsf` layout

Monash benchmarks (Godahewa et al., 2021) ship as `.tsf` under `data/<name>/`. A streaming parser reads `@frequency`, `@horizon`, `@equallength`, `@missing`; missing `@horizon` defaults per Table 2 and Section 4.2. Elec.-W is `@equallength true`; Type III in Figure 3 uses a TV-norm proxy (see §5.4).

A.2 Train–validation–test split and cross-validation

For each series, observations are split into training, validation, and test windows with horizon H as in Section 4.2. Series shorter than $H + K_{\max} + 17$ observations are dropped. **3-fold** cross-validation uses `fold_id = series_idx mod 3` with shuffle seed 2026. Two random seeds (42, 52) are run per fold, giving **6 fold–seed blocks** per dataset. Unless `FSDCM_MAX_SERIES=all`, the default cap is **512** series per dataset, subsampled via **reservoir sampling** (fixed seed). For Weather specifically, summary statistics exclude the top-1% of series by mean baseline MASE (p99 threshold) to avoid a single extreme-outlier fold dominating the reported means; raw series-level metrics are always written to `*_all_metrics_*.csv`. Robust scaling uses the 1st–99th percentile computed *per fold* on training data only, then applied to validation and test—avoiding leakage.

A.3 Backends: AutoGluon-TimeSeries and scikit-learn HGBR

Main tables/plots use `autogluon`; Table 4 adds `sklearn_hgb` (lag features + `HistGradientBoostingRegressor`, recursive H-step forecast (Pedregosa et al., 2011); same time caps). Outputs live under `results/<backend>/`. AG-TS: `preset="fast_training"`, `eval_metric="MASE"`, `time_limit=120s` per `fit`, history cap 1024, TFT omitted on Windows (TensorBoard paths). Tune f_c from periodogram peaks and $K \in \{5, 10, 15, 20\}$.

A.4 FS-DCM variant flags

`run_benchmark.py` implements the ladder in Section 4.3: `baseline`, `last_k_only`, `freq_only`, `fsdcm_static`, `fsdcm_no_tta`, `fsdcm_full`. When `INCLUDE_COMPUTE_FAIR_BASELINES=True`, two additional baselines appear: `baseline_2x_time` (doubled AG-TS `time_limit`) and `baseline_2seed_ensemble` (mean of two fits with seeds 42/52). Test-time augmentation and regime routing are enabled only for `fsdcm_full`; `fsdcm_no_tta` disables TTA but keeps routing. Regime monitoring uses a **histogram** divergence between Last-K and full-history samples (not a Gaussian KDE): **Jensen–Shannon** is the default (`REGIME_DIVERGENCE_METHOD` or `FSDCM_REGIME_DIV=js`); set `FSDCM_REGIME_DIV=kl` for legacy KL. Hard safeguard thresholds in `src/fsdcm/safeguard.py` were originally scaled for KL; under JS the bypass rate can shift, so ablations should state which divergence was used. DLVW uses λ_D, validation-estimated σ_F^2, and—when `USE_VALIDATION_RHO=True` (default)—a correlation ρ fit on validation-window low-frequency vs. residual errors (fallback to the naive ρ proxy if estimation fails), as in Section 3.3.

A.5 Logged outputs and aggregation

Each run writes `results/<backend>/<dataset>_all_metrics_<timestamp>.csv` with one row per (`backend`, `variant`, `fold`, `seed`, `series_id`) with columns `mase`, `smape`, tuned f_c, K, λ_D, `safeguard_bypass` (0/1), `bypass_reason` (string code; empty when no bypass), `error` (empty on success), and `run_config_*.json` metadata. Aggregated counts of bypass reasons for `fsdcm_full` appear in `*_bypass_reason_counts_*.csv`. Mean metrics stratified by `safeguard_bypass` for `fsdcm_full` appear in `*_safeguard_split_*.csv`; fold-level stability of (f_c, K) appears in `*_hyperparam_stability_*.csv`. Paired-versus-baseline summaries and fold–seed block statistics are emitted as `*_paired_metrics_*.csv`, `*_fold_seed_summary_*.csv`, and `*_summary_reliability_*.csv`. `run_config_*.json` records environment and CLI-equivalent settings for auditing.

A.6 Applicability bypass reason codes

Hard baseline reverts use the following `bypass_reason` labels: `short_train` (quarterly train shorter than 88 steps), `short_residual_window` (effective Last-K window below 20 on quarterly), `low_c1_energy` (share of FFT energy below f_c below the frequency-dependent floor in `src/fsdcm/safeguard.py`, e.g., 0.11 on quarterly and 0.04 on daily), `kl_mismatch` (regime divergence exceeds θ times a frequency-dependent multiplier; despite the name, the statistic is JS when JS mode is enabled). On Weather, bypass counts for `fsdcm_full` are: `low_c1_energy` 200, `kl_mismatch` 28 (total 228/1,012; 22.5%); sklearn HGBR (no bypass) yields **17.1%** paired reduction across all Weather series.

A.7 Supplementary Figures

The figures below provide additional visual evidence complementing the main results (Table 3, Table 4, Figure 2).

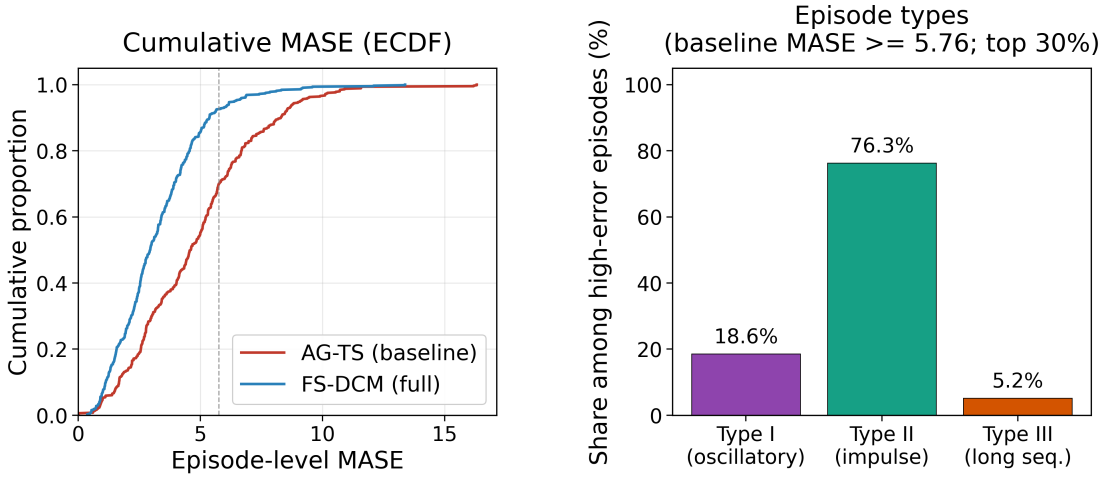


Figure 3: FS-DCM improves tail errors without harming easy cases (Elec.-W). **(Left)** ECDFs of episode MASE overlap below the 70th percentile, then diverge sharply: FS-DCM compresses the right tail where the baseline produces its worst predictions (fold-averaged P90-MASE: 9.0 → 4.8, Table 5; see also Figure 7). **(Right)** Among the top-30% hardest episodes, Type II impulse events dominate (76%), indicating that dual-context modeling specifically addresses the structural-break episodes predicted by the Information Lag theory (§3.1).

Long-tail analysis..

Cross-backend overview.. Figure 4 juxtaposes Δ% heatmaps for both backends (scikit-learn HGBR and AutoGluon-TimeSeries) side by side. The shared diverging colormap makes the regime pattern immediately visible: Weather and Elec.-W cells turn green at FS-DCM (Full), while quarterly columns remain red—confirming that applicability regime effects replicate across backends.

In-scope benchmarks only (AutoGluon).. Figure 5 isolates the two datasets nominally satisfying applicability conditions (C1–C3): Weather and Elec.-W. Under the refined JS-based regime detector and validation-estimated ρ, Elec.-W delivers a strong 31.0% gain; Weather is backend-dependent (sklearn HGBR 17.1%; AutoGluon near-zero, 22.5% safeguard bypass).

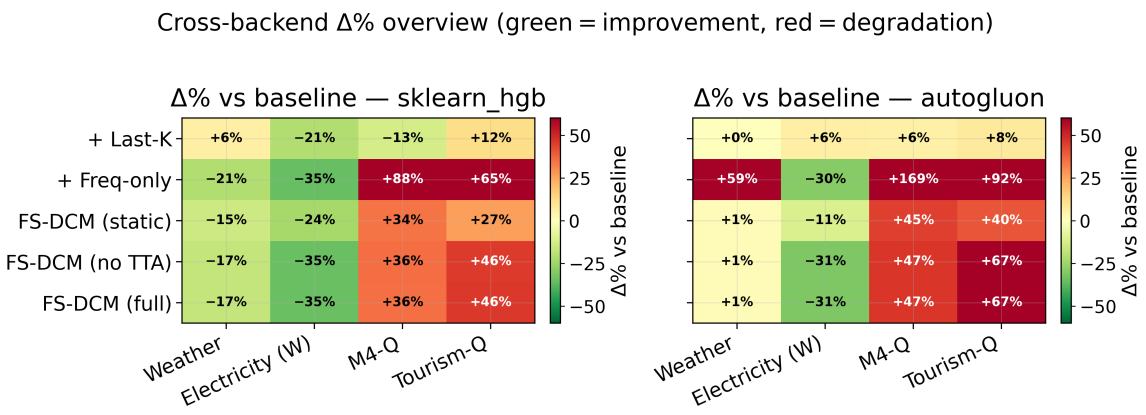
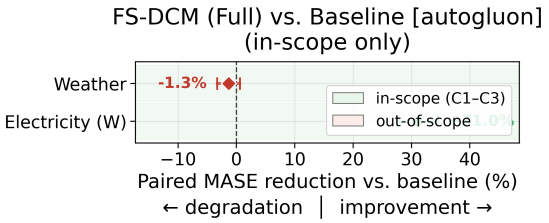
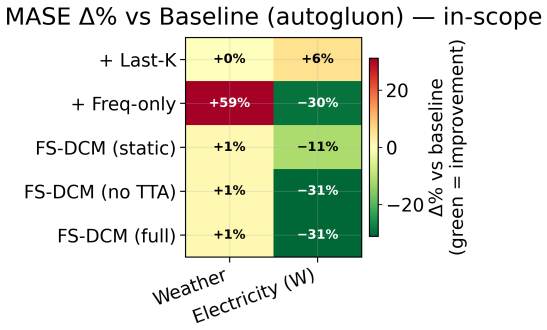


Figure 4: Cross-backend Δ% overview (green = improvement; red = degradation). **Left**: scikit-learn HGBR. **Right**: AutoGluon-TimeSeries. Shared diverging colormap with a common cap; cells annotated with % values. The regime-dependent pattern—green on weekly data, red on quarterly—replicates across both backends, confirming backend-agnostic applicability.



(a) Forest plot (95% CI) on in-scope datasets only. Elec.-W shows **31.0%** statistically significant gain (green); Weather near-zero under AutoGluon (22.5% safeguard bypass).

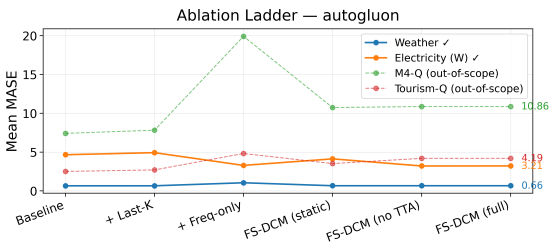


(b) Δ% heatmap on in-scope datasets only. Green cells progressively deepen along the ablation ladder, confirming the contribution hierarchy; values annotated in each cell.

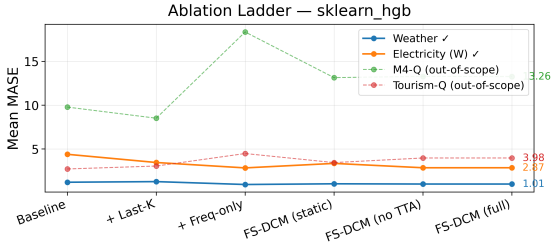
Figure 5: **In-scope benchmarks** (AutoGluon). Forest plot + Δ% heatmap. Elec.-W shows a strong 31.0% paired MASE reduction; Weather is backend-dependent (AutoGluon near-zero; sklearn HGBR 17.1% reduction).

Ablation ladder slope charts.. Figure 6 traces mean MASE along the ablation ladder (Baseline → Last-K → Freq-only → FS-DCM Static → no-TTA → Full) for all datasets. Solid lines are in-scope datasets; dashed are out-of-scope. The slope format directly visualises the contribution hierarchy and the endpoint annotation shows each dataset's final MASE.

P90-MASE tail performance.. Figure 7 compares the 90th-percentile MASE (P90-MASE)—the metric that captures tail-error episodes—across all variants and datasets. On Elec.-W, FS-DCM (Full) reduces P90-MASE from 9.0 to 4.8 (Table 5), a 47% tail-error reduction that is disproportionately larger than the 31.0% mean-MASE reduction, confirming that the dual-context architecture preferentially benefits high-difficulty episodes.

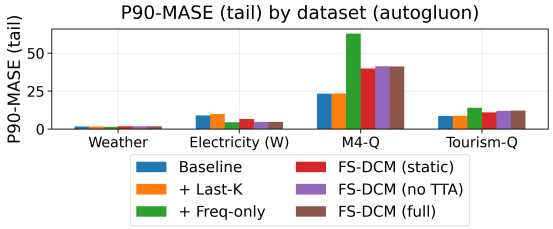


(a) AutoGluon-TimeSeries. Elec.-W (solid) descends steeply at the DLVW step; Weather stays near-baseline (safeguard bypass); dashed out-of-scope lines (quarterly) rise, confirming the predicted degradation.

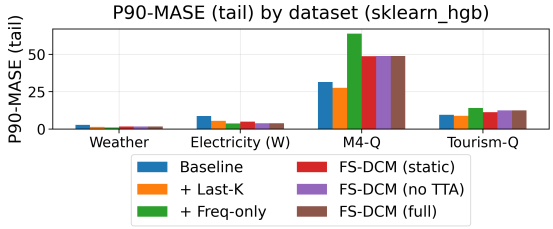


(b) Scikit-learn HGBR. The same in-scope descent / out-of-scope ascent pattern replicates under a structurally different model family.

Figure 6: Ablation ladder (slope chart): mean MASE across six variants. Solid = in-scope; dashed = out-of-scope. The DLVW step produces the steepest descent on Elec.-W (in-scope), validating it as the dominant contributor (Table 5).



(a) AutoGluon P90-MASE. The tail improvement on Elec-W is visually striking; quarterly datasets show elevated P90-MASE for Freq-Only and FS-DCM variants, consistent with scope violations.



(b) Scikit-learn HGBR P90-MASE. The backend replicates the same tail-error reduction on in-scope benchmarks and the same elevated tail error on quarterly data.

Figure 7: Dual-context modeling disproportionately reduces tail errors on in-scope data. P90-MASE (90th-percentile episode MASE) drops 47% on Elec.-W (9.0 → 4.8), far exceeding the 31.0% mean-MASE reduction—evidence that FS-DCM preferentially compresses the hardest predictions. Elevated P90-MASE on quarterly data reflects (C1) scope violations, not method breakdown, as the safeguard logs confirm (Appendix A).

B Full Proof of Proposition 1

Let the structural break occur at time $\tau \in [T-K+1,\, T]$, and let $\Delta y_t = y_t - y_{t-1}$ denote the first difference. The power spectrum of the break is concentrated at high frequencies $f > f_c$ if and only if the break constitutes a rapid transition—formally, if $\|\mathcal{F}(\mathbf{1}_{[\tau,T]} \cdot \Delta y)\|^2$ is predominantly supported on $[f_c, 0.5]$. Under this assumption, applying the low-pass filter $\mathcal{F}^{-1}(\mathcal{F}(y) \cdot \mathbf{1}[f < f_c])$ removes the break signature by the Parseval identity, yielding $\hat{y}^{\text{Full}}_{1:T} \approx \mu_t$ for $t > \tau$. The residual $y^{\text{res}}_{1:T} = y_{1:T} - \hat{y}^{\text{Full}}_{1:T}$ therefore captures the full break signature in the high-frequency band. Since the Last-K model trains exclusively on $y^{\text{res}}_{T-K+1:T}$, which lies entirely in the post-break window $[\tau, T]$ when $K \le T - \tau + 1$, it approximates the post-break conditional distribution P^*_T, directly minimizing the KL term in Definition 1. By orthogonality of the Fourier basis, the low-frequency component is unaffected by the high-frequency break, and $\hat{P}_{Full,T}$ computed on $\hat{y}^{\text{Full}}_{1:T}$ remains well-calibrated for

the trend component. Together, their linear combination (Equation 5) recovers both components of P_T^*. □