Chronos-2: A Replication and Extension Study

Hassan Maatouk[1], Owais Ali[1], Hassine El Ghazel[1], Arash Rahmani[1], Koen Van Den Berk[1,2]
[1]*Politecnico Di Torino,* [2]*Eindhoven University of Technology*
S343952, S338975, S346265, S343938, S350800

***Abstract*—Chronos-2 is a state-of-the-art framework for universal time series forecasting. This report evaluates four targeted extensions aimed at improving computational efficiency and forecast accuracy through inference-time modifications.**

We evaluate: (1) sparse windowed temporal attention to reduce computational cost, (2) augmented inference with engineered covariates or variables, (3) soft similarity-weighted group masking to enable cross-group learning, and (4) semantic cross-learning through similarity-based context construction. Our experiments on standard benchmarks reveal that sparse attention preserves accuracy while discarding substantial attention mass, though runtime gains are not realized without optimized kernels. Augmented inference degrades performance, likely due to distribution mismatch. Soft masking shows no significant improvement over baseline hard masking, suggesting the model's pretrained representations already capture relevant cross-series dependencies. Semantic cross-learning demonstrates directional improvements on two-thirds of tasks, with statistical significance under non-parametric testing.

Code

I. Introduction

Time series forecasting is fundamental to decision-making across finance, energy systems, supply chain management, and environmental monitoring. Traditional forecasting methods are typically trained on specific datasets, limiting their ability to generalize across domains. Inspired by the success of foundation models in natural language processing, recent research has pursued universal time series forecasting through large-scale pretrained transformer architectures. Chronos-2 [**ansari2025chronos2**] represents a state-of-the-art approach in this direction, reformulating forecasting as a sequence modeling problem by discretizing time series values into tokens. This design enables unified treatment of univariate and multivariate settings with probabilistic predictions via quantile estimation, demonstrating strong zero-shot performance across diverse benchmarks. While Chronos-2's pretrained representations are robust, opportunities remain to enhance performance through targeted inference-time modifications that preserve model compatibility and require no retraining. This work investigates whether inference-time interventions targeting attention mechanisms, input construction, and batch composition can meaningfully improve forecasting accuracy or computational efficiency.

II. Problem Statement

This work investigates four inference-time modifications to the pretrained Chronos-2 forecasting framework. Each extension addresses a specific limitation or opportunity while maintaining compatibility with the pretrained model. We formalize the problem setting for each extension by specifying the expected input, the addressed task, and the expected output. All extensions operate exclusively during inference using the frozen Chronos-2 model, requiring no retraining or architectural modifications.

A. Sparse Time Attention

Chronos-2 performs dense temporal self-attention over the context window, which scales quadratically with the number of tokens and can become a computational bottleneck on long horizons [**vaswani2017attention**]. The problem is to determine whether Chronos-2 can retain zero-shot forecast performance when context-to-context temporal attention is sparsified using a local window of radius r,[**beltagy2020longformer**, **child2019sparse**] while keeping attention for the future-tokens dense and leaving all model weights unchanged.

B. Augmented Inference

This extension attempts to improve the pre-trained Chronos-2 model by adding extra information during inference. Inspiration for this extension is recent literature establishing that repeating a prompt in a causal, non-reasoning LLM enhances performance [**leviathan2025prompttrepetitionimprovesnonreasoning**]. Similarly, Chronos-2 could improve if the "prompt" is augmented. Chronos-2 is given an original univariate input sequence in a zero-shot context and additional inputs are engineered from the original input. The additional inputs are added as covariates or as variables to be predicted. The output is a univariate prediction plus the corresponding quantiles.

C. Soft Group Masking

Chronos-2's group attention mechanism uses binary masking: time series with identical group IDs can attend to each other, while series in different groups cannot exchange information. This prevents leakage across unrelated series but may block useful information when related series are assigned to different groups due to batching constraints. The objective is to investigate whether relaxing this hard constraint through similarity-weighted cross-group attention can improve forecast accuracy when correlated time series co-occur in the same batch but belong to different groups. Given a batch of B time series $\{\mathbf{x}_1, \ldots, \mathbf{x}_B\}$ with group IDs, we aim to produce quantile forecasts $\{\hat{\mathbf{y}}_i^{(q)}\}$ that leverage pairwise similarity to modulate cross-group information flow without compromising the model's ability to prevent interference between unrelated series.

D. Semantic Cross-Learning

Chronos-2 [**ansari2025chronos2**] supports inference-time information sharing via *cross-learning* (ICL) by assigning multiple series in a batch the same group ID, enabling cross-series context without retraining or changing weights. However, arbitrary batching may yield incoherent helpers, which is problematic on heterogeneous benchmarks like FEV [**shchur2025fev**] (diverse domains, frequencies, seasonalities, noise, and covariates), making performance sensitive to batch composition. We therefore propose a pre-inference grouping strategy that selects helpers by similarity while keeping Chronos-2 inference code and weights unchanged.

III. Proposed Extensions

A. *First extension: Sparse Time Attention*

One extension assesses replacing dense temporal attention mechanisms with sparse attention, while keeping the tokenization, group attention and quantile forecasting pipeline unchanged. Concretely, we implemented a **windowed** temporal attention scheme for the past history in which each token attends to a local neighborhood of radius r "local_radius", while the future attends to all previous tokens in a dense way. Attention computation is performed in chunks to reduce memory pressure and enable efficient kernels when available. The extension is integrated into the existing pipeline via configuration flags (attention type, backend, radius, chunk size), allowing controlled comparisons against the original full-attention baseline under identical preprocessing and evaluation settings. Sparse attention theoretically reduces computational complexity by focusing on the most relevant time steps, potentially improving efficiency without compromising accuracy[**zaheer2020bigbird**].

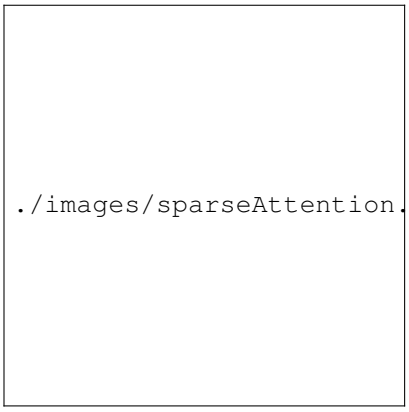

Fig. 1: Sparse time attention: windowed context (radius r) and global future attention.

To quantify how much attention is discarded by sparsification, we run the pretrained model with full temporal attention and extract per-layer attention weights. For a given radius r, we compute the fraction of the attention probability mass that falls inside the sparse mask. This yields a model dependent 'mass retention' curve, complementary to the purely geometric edge-retention fraction. All experiments are performed in a zero-shot setting using the pre-trained Chronos-2 weights.

Experimental Setup: To quantify "how much attention is discarded" by sparse time attention, we run chronos-2 with full time attention and extract the encoder time self-attention weights. For each radius r, we construct the sparse mask matching our implementation and compute the fraction of full-attention probability mass that lies inside the allowed pattern (kept mass), restricted to context queries. We also report the purely structural fraction of retained query-key edges (kept edges). Since Chronos-2 uses patching, we additionally stratify results by the effective token length S. We evaluate the pretrained Chronos-2 pipeline on Chronos Benchmark II tasks[**ansari2024chronos**]. We compare full time attention against sparse time attention with radii $r \in 8, 16, 32, 64, 128$ and compute per-task performance delatas $\triangle MASE = MASE_{sparse} - MASE_{full}$ following the benchmark's build-in evaluation. Statistical significance is assessed using a paired t-test over tasks. Inference time is measured as median over N repeats per task, using CUDA synchronization before and after each run, reporting the task-level inference time and speedup ratio (t_{sparse}/t_{full}). We also record peak GPU allocated and researved memory per task (median across repeats).

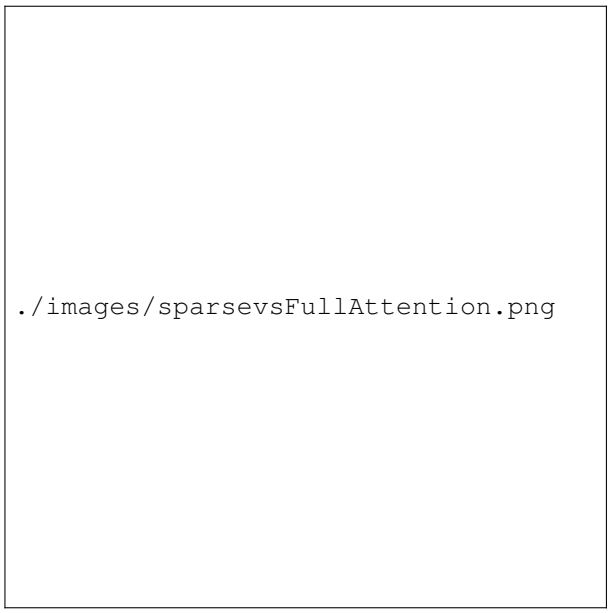

Fig. 2: Attention mass retained vs radius (context queries only)

Results: Because patching yields very different effective token lengths S, we stratify by S. In long-token windows ($S \geq 128$), small radii retain only a small fraction (3.3%) of the full-attention mass for context queries, evident from the Figure 2. Across radii, the retained mass closely tracks the retained edge fraction, indicating that aggregated context-query attention is broadly distributed across context keys rather than sharply concentrated locally.

Radius	Median ΔMASE	Median speedup	p-value
8	-0.000086	0.965846	0.107531
16	-0.000091	0.959966	0.887393
32	0.000009	0.960674	0.288954
64	-0.000117	0.961477	0.221018
128	-0.000032	0.963810	0.219545

TABLE I: Sparse time attention vs. full attention (all contexts pooled). ΔMASE (negative is better). We report medians to reduce sensitivity to outlier tasks. Speedup values < 1 indicate sparse is slower).

Despite discarding most of this mass at small radii, performance remains essentially unchanged, suggesting that the dropped long-range attention mass is not critical for CBII accuracy in this regime. Across the evaluated CBII tasks evaluated across different radii, sparse time attention achieves near-pairty with full attention. The average $\triangle MASE(sparse - full)$ across the radii stay close to zero, and paired t-tests over the task-level deltas does not reject the null hypothesis of equal mean performance (all ($p > 0.1$). In our current implementation, sparse attention does not yield runtime gains; speedup ratios, computed from median inference time over repeats, are less than 1, indicating overhead dominates under this token regime.

Results Analysis: Chronos-2's architectural step of patching reduces the practical benefit of sparsifying the attention matrix[**nie2022patchtst**]. Secondly, although aggregated full-attention weights over context queries appear broadly distributed, CBII accuracy is largely insensitive to removing long-range context links, suggesting redundancy in how historical information is repeated (e.g. via patching, residuals and summary tokens). Consequently, sparse attention can match full-attention accuracy but does not improve runtime without a more optimized sparse attention kernal[**dao2023flashattention2**].

B. Second extension: Augmented inference

The extension consists of two related but different approaches. The first approach involves injecting transformed versions of an input time series $\mathbf{x}$ as covariates. Specifically, let

$$f_1(\mathbf{x}) = \mathbf{x}^3, \quad f_2(\mathbf{x}) = \exp \mathbf{x}, \quad f_3(\mathbf{x}) = -\mathbf{x}, \tag{1}$$

$$f_4(\mathbf{x}) = -\mathbf{x}^3, \; f_5(\mathbf{x}) = -\exp \mathbf{x}, \tag{2}$$

where operations are element-wise. To ensure significant transformation and prevent impractically large values, we do not transform $\mathbf{x}$ directly, but transform a normalized version. So, the covariates introduced are

$$f_i\left(\frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right), \; i = 1, 2, 3, 4, 5.$$

The idea is that these covariates could help Chronos-2 recognize patterns that would otherwise remain unnoticed. A tempting strategy, given the prompt-repetition result for LLMS is to simply include $\mathbf{x}$ as a covariate. However, by construction of Chronos-2, attention to the original time series is equivalent to attention to the covariate. Hence, adding $\mathbf{x}$ as a covariate has zero effect. In addition to the f_i's, an experiment is also conducted with covariates

$$g_1(\mathbf{x}) = \text{MovingAverage}(\mathbf{x}, 9), \quad g_2(\mathbf{x}) = \mathbf{x} - g_1(\mathbf{x}). \tag{3}$$

So Chronos-2 is given a smoothed version of $\mathbf{x}$ and the residual of the smoothed version. A sliding window of 9 was chosen based on the patch size of 16. Edges are handled by repeating the outer values as padding.

The second approach involves considering these covariates as time series to be predicted. These additional predictions are then combined with the original prediction. Observe that f_i are invertible functions. Under the assumption that Chronos-2 is truly a zero-shot model, there is no reason to expect the prediction Chronos-2$(\mathbf{x})$ to be better than f^{-1}(Chronos-2$(f(\mathbf{x}))$. After all, if Chronos-2 made perfect predictions, both vectors would be equal. In practice, the performance of Chronos-2 may be biased due to the nature of the training data. Therefore, combining the predictions f_i^{-1}(Chronos-2$(f_i(\mathbf{x}))$ could be more robust. The final prediction is the unweighted average over all predictions (original and transformed). The same is done for quantiles. Some important notes on this approach:

- Input is normalized before transformation. Hence, predictions need to be normalized. Precisely, the additional predictions for $i = 1, \ldots, 5$.

$$\mu(\mathbf{x}) + \sigma(\mathbf{x}) f_i^{-1}\left(\text{Chronos-2}\left(f\left(\frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right)\right)\right). \tag{4}$$

- Functions f_2 and f_5 are only invertible if the prediction is positive. A perfect predictor would always produce such values. An ad-hoc solution is to clip small or negative values to some small constant ε.
- Decreasing transformation functions, like f_3, f_4 and f_5, will yield predicted quantiles in flipped order. So a predicted q-quantile will correspond to a $(1-q)$-quantile of $\mathbf{x}$.

Experimental Setup

The experiment targets zero-shot performance on univariate datasets. All four variants are evaluated on the Chronos-Zeroshot collection. Each variant is applied to both Chronos-2 and Chronos-2-Synth (Chronos-2 trained only on synthetic data). This gives a total of 8 experiments. The win rate, MASE and WQL is analyzed in each experiment. These are then compared with the baseline, being Chronos-2 and Chronos-2-Synth, respectively. A statistically significant improvement in any of the three metrics is considered promising.

Results

None of the four experiments produce a promising result: the win rate is always lower with respect to the baseline. Mean MASE and WQL is also lower for each experiment. Inference time is 6 times as high when using f_i's and 3 times as high for when using g_i's. As expected, the inference time is directly proportional to the number of input sequences. The full table can be found in Table II.

Base model	Variant	Win rate	MASE	WQL	Inference time (s)
Chronos-2	Baseline	0.76	1.37	0.14	21
	Covariates f_i	0.51	1.44	0.14	118
	Covariates g_i	0.58	1.42	0.15	60
	Copredict f_i	0.31	1.60	0.17	123
	Copredict g_i	0.44	1.45	0.15	63
Chronos-2-Synth	Baseline	0.69	1.41	0.14	21
	Covariates f_i	0.55	1.43	0.14	119
	Covariates g_i	0.62	1.41	0.14	61
	Copredict f_i	0.41	1.66	0.19	120
	Copredict g_i	0.48	1.45	0.15	61

TABLE II: Win rate, mean MASE, mean WQL and mean inference time for variants of the augmented inference extension. Evaluated on Chronos-Zeroshot benchmark

Interpretation

Augmented inference does not result in out-of-the-box improvement of Chronos-2. In fact, performance declines. Further research may explore first fine-tuning using augmented inputs. Note that adding inputs increases inference time. As a result, augmented inference may not be an attractive candidate for further experimentation.

C. *Third extension: Soft Group Masking Extension*

a) Motivation and Hypothesis: Chronos-2 employs hard group masking to enable batch-level cross-learning, where time series with identical group IDs can attend to each other while series in different groups cannot exchange information [**ansari2025chronos2**]. This binary attention constraint is expressed as a hard mask $\mathbf{M}_{\text{hard}} \in \{0,1\}^{B\times B}$ where element $[i,j]$ equals 1 if time series i and j belong to the same group, and 0 otherwise. While this prevents information leakage across unrelated series, it may inadvertently block useful information exchange when related time series are assigned to different groups due to random batching. **We hypothesize** that replacing hard masking with similarity-weighted soft masking at inference time will improve forecast accuracy when related series are co-located in the same batch but assigned to different groups, particularly for datasets with correlated time series.

b) Technical Approach: Our extension replaces the binary hard mask with a continuous soft mask $\mathbf{M}_{\text{soft}} \in [0,1]^{B\times B}$ that encodes pairwise similarity between time series:

$$\mathbf{M}_{\text{soft}}[i,j] = \begin{cases} 1.0 & \text{if group\_id}[i] = \text{group\_id}[j] \\ \text{sim}(\mathbf{x}_i, \mathbf{x}_j) & \text{otherwise} \end{cases} \tag{5}$$

where $\text{sim}(\cdot,\cdot) : \mathbb{R}^T \times \mathbb{R}^T \to [0,1]$ measures similarity between input context sequences. This soft mask is integrated into Chronos-2's self-attention via an additive attention bias:

$$\text{attention\_bias}[i,j] = \log(\mathbf{M}_{\text{soft}}[i,j]) \times \tau \tag{6}$$

where $\tau = 5.0$ is a temperature parameter balancing cross-group information flow and training distribution fidelity. The logarithm ensures zero similarity results in $-\infty$ bias (complete masking), while high similarity approaches zero bias (full attention).

We employ Pearson correlation to measure linear relationships between standardized time series $\hat{\mathbf{x}} = (\mathbf{x} - \mu)/\sigma$:

$$\text{sim}_{\text{corr}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}\left(1 + \frac{\sum_{t=1}^{T}(\hat{x}_{i,t})(\hat{x}_{j,t})}{\sqrt{\sum_{t=1}^{T}\hat{x}_{i,t}^2 \sum_{t=1}^{T}\hat{x}_{j,t}^2}}\right) \tag{7}$$

mapping correlation from $[-1,1]$ to $[0,1]$.

c) Implementation Details: The soft masking extension is implemented with minimal modifications to the Chronos-2 codebase. We add a similarity computation module that calculates pairwise relationships between input time series based on their raw context values. Critically, this is an **inference-only** approach requiring no model retraining. Users can enable soft masking through optional parameters in the prediction interface, allowing seamless switching between hard and soft masking modes while maintaining full backward compatibility.

d) Results: We evaluated soft group masking against baseline hard masking across the Chronos Benchmark II datasets using MASE (Mean Absolute Scaled Error) and WQL (Weighted Quantile Loss). Table III summarizes the aggregate performance.

Metric	Baseline	Soft Masking	Improvement
MASE	2.7686	2.6811	+2.8%
WQL	0.1514	0.1505	+0.39%

TABLE III: Baseline (hard masking) vs. soft masking performance on Chronos Benchmark II. Lower is better for both metrics.

While soft masking shows modest numerical improvements, paired t-tests reveal these differences are not statistically significant (MASE: $p = 0.211$; WQL: $p = 0.765$). We conclude that the **hypothesis is not supported**: soft masking does not provide meaningful improvements over standard hard masking in Chronos-2.

To further investigate whether correlation structure influences the effectiveness of soft masking, we stratified datasets by their expected cross-series dependencies. For correlated datasets (ETTh1, Weather, where sensors or temporal patterns exhibit strong dependencies), soft masking showed a stronger directional trend toward significance (paired t-test: $p = 0.124$) compared to weakly correlated datasets such as Walmart time series ($p = 0.266$). While both remain above the significance threshold, the directional pattern suggests that soft masking may provide marginal benefits in highly correlated settings, though the effect is too small to be reliable under current inference-only application.

We attribute this null result to three factors: (1) **training-inference mismatch**—the model was trained exclusively with hard masking, creating distribution shift when soft masking is applied only at inference; (2) **sufficiency of learned representations**—Chronos-2's pretraining likely already captured

relevant cross-series dependencies through group attention, leaving limited room for similarity-based refinement; and (3) **similarity metric limitations**—raw Pearson correlation may not align with the latent similarity structures the model internally relies upon for cross-learning.

D. Fourth extension: Context Construction and Batched Inference

Chronos-2 [**ansari2025chronos2**] controls inference-time information sharing via *group IDs* within each batch. With cross-learning *off*, each series gets its own group ID, yielding standard univariate inference with independent predictions. With cross-learning *on*, multiple items can share a group ID so the model can use shared context during the (unchanged) forward pass. Our extension modifies only *batch composition*: for each target series, we optionally attach a small set of auxiliary *helper* series and assign them the same group ID, creating an **augmented cross-learning context** while keeping Chronos-2 tokenization, parameters, and inference path unchanged. We evaluate three modes differing only in group formation: **Baseline** (cross-learning off; no helpers), **Random cross-learning** (helpers sampled uniformly at random), and **Semantic cross-learning** (helpers retrieved by cosine similarity between lightweight *series signatures* computed from history windows, then packed with a deterministic policy: Top-K first, weak fill, and a global fallback to reduce fragmentation). In all cases, only grouping and batching change; the Chronos-2 forward pass is identical.

Fig. 3: Chronos-2 inference pipeline with the proposed extension layer. The Chronos-2 forward pass is unchanged; the extension operates strictly *before* inference by constructing grouped contexts and assembling batches.

*1) **Experiments and evaluations**:* Experiments are run under a single, fixed evaluation configuration; Table IV summarizes the benchmarks, metrics, shared inference settings, and the semantic extension defaults used in our implementation. Unless otherwise stated, only *grouping and batch construction* change; the Chronos-2 model forward pass remains identical across modes.

Parameter / setting	Value
Benchmarks	`fev-bench` (*80 tasks over 100 successfully tested*)
Metrics	`MASE`, `WQL`
Modes compared	Baseline (*cross-learning off*); Random cross-learning (Chronos-2 ICL); Semantic cross-learning (ours)
Model (`model\_id`)	`amazon/chronos-2`
Precision (`dtype`)	`float32`
Batch size	`32`
Semantic grouping	`neighbors`
Top-K	`64`
Neighbor threshold	`0.20`
Alt. clustering params (if used)	`num\_clusters=50`, `kmeans\_iters=25`

TABLE IV: Evaluation configuration and semantic extension defaults.

We compare three inference-time modes on FEV [**shchur2025fev**] tasks: *Baseline* (cross-learning off), *Random* cross-learning, and *Semantic* cross-learning. For FEV, **we *flatten* each task into a set of univariate forecasting instances: regardless of whether a task is originally univariate, covariate, or multivariate**, evaluation is performed by forecasting a *single target series* per instance. Lower is better for both metrics (**MASE**, **WQL**). All Random vs Semantic statistics are *paired across tasks*. We report mean (median) metric values, the mean percent gain of Semantic over Random, and the fraction of tasks where Semantic strictly outperforms Random. Significance is assessed with a **paired t-test (two-sided)** and a **Wilcoxon signed-rank test** (one-sided, H_1: Semantic $<$ Random).

Metric	n	Base (mean/med)	Rand (mean/med)	Sem (mean/med)	$\Delta(\%)$ & $\mathrm{WR}_{S<R}$	p_t/p_W
MASE	80	1.373 (0.834)	1.335 (0.791)	**1.325** (0.795)	+0.78% WR=66.2%	0.137 0.0007
WQL	80	0.1789 (0.1149)	0.1744 (0.1138)	**0.1738** (0.1132)	+0.69% WR=67.5%	0.201 0.0029

TABLE V: Mean/median (lower is better). $\Delta(\%)$: mean percent gain (Semantic vs Random); $\mathrm{WR}_{S<R}$: fraction where Semantic beats Random. p_t: paired t-test (two-sided); p_W: Wilcoxon (one-sided, H_1: Sem<Rand). "Exploded" MASE tasks (MASE>10 in any mode) are removed, yielding n=80.

*2) **Interpretation**:* Semantic cross-learning shows a consistent *directional* advantage over Random Table V: it wins on about two-thirds of tasks ($\mathrm{WR}_{S<R} \approx 66$–$68\%$) for both **MASE** and **WQL**. This is reflected by the Wilcoxon signed-rank test, which is significant under standard thresholds ($p_W < 0.05$ for both metrics), indicating the improvements

are systematic across tasks. In contrast, the paired t-test is not significant ($p_t > 0.05$) because the *mean* gains are small and task-level differences are heavy-tailed, so a few large deviations can mask many small wins in the average. When retrieved neighbors closely match the target dynamics, gains can be substantial, e.g., `bizitobs_l2c_1H` (**+7.69%** MASE, **+5.19%** WQL), `world_tourism` (**+1.65%** MASE, **+4.77%** WQL), `m5_1W` (**+10.63%** MASE, **+8.55%** WQL), and `rohlik_orders_1W` (**+2.31%** MASE, **+10.19%** WQL). Degradations occur on a minority of datasets (e.g., `walmart`: **-7.24%** MASE, **-2.93%** WQL; `rohlik_orders_1D`: **-0.55%** MASE, **-5.21%** WQL), consistent with occasional neighbor mismatch introducing incompatible structure into the shared context.

Finally, statistics are computed on the tasks that completed successfully: after excluding "exploded" MASE cases (MASE>10 in any mode) and removing 17/100 tasks that could not be evaluated due to limited computational capabilities, paired comparisons use n=80 tasks.