STRUCTURE AT THE TOP, NOISE AT THE LEAVES: HIERARCHY-AWARE RESIDUAL ADAPTATION OF FROZEN TIME-SERIES FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series foundation models (TSFMs) forecast each series on its own and ignore the aggregation structure that ties many real systems together. We ask whether an explicit hierarchy can improve a *frozen* TSFM without touching its weights. This work continues the hierarchical residential load forecasting study of Sucaldito & Ostia (2025), which found that per-series AutoARIMA and LightGBM models forecast aggregates well but degrade sharply on zero-inflated appliance series, and which called for hybrid, hierarchy-aware models. We introduce such a hybrid, the *Hierarchy-Aware Residual Adapter* (HAA), a 19k-parameter MLP that sits on top of frozen TimesFM 2.5 forecasts. It predicts leaf-level residuals from structural context: bottom-up category, household and system aggregates, the leaf's share of each, recent temporal statistics, and category identity. Bottom-up aggregation of the corrected leaves keeps every level exactly coherent. On a four-level residential energy hierarchy (Total $\rightarrow$ 25 households $\rightarrow$ 105 categories $\rightarrow$ 218 sub-metered appliances, Pecan Street New York, 15-minute resolution), we evaluate a 24-hour forecast against naïve, seasonal-naïve, Chronos and TimesFM baselines. HAA lowers pooled RMSE relative to frozen TimesFM at every level: by 2.0% (appliance), 1.1% (category), 5.1% (household) and 20.2% (system total; Diebold–Mariano $p < 10^{-6}$). A closer analysis gives a more cautious picture. HAA beats TimesFM on only 24%, 30% and 36% of individual appliance, category and household series. Series-bootstrap confidence intervals for the lower-level gains include zero. Almost all of the gain at the total level comes from removing a systematic under-forecast (the bias share of TimesFM's total-level MSE falls from 73% to 47%), while the error variance actually grows. The adapter also produces negative loads in 31% of leaf predictions and worsens MAE and MAPE at the leaves. Projecting leaf predictions onto the non-negative orthant before aggregation raises the total-level RMSE gain to 29.4%. We argue that structural context is most useful as an *aggregation-amplified bias corrector*. We also release a protocol (rolling origins, household hold-out, hierarchy permutation, cross-city transfer) for testing whether such adapters show real *structural generalization*.

1 INTRODUCTION

Pretrained time-series foundation models such as Chronos (Ansari et al., 2024), TimesFM (Das et al., 2024), Moirai (Woo et al., 2024) and MOMENT (Goswami et al., 2024) now give competitive zero-shot forecasts across many domains. They share one design choice: each input series is treated as an isolated sequence. Many operational systems, however, are *hierarchical*. An electricity total is the sum of households, a household is the sum of its end-use categories, and a category is the sum of its appliances. The classical hierarchical forecasting literature (Hyndman et al., 2011; Wickramasuriya et al., 2019; Athanasopoulos et al., 2024) uses this structure mostly *after* forecasting, through reconciliation. The structure is rarely used as an *input* that could change the forecasts themselves.

Continuation of prior work. This paper continues the line of work begun by Sucaldito & Ostia (2025). That study organized the Pecan Street New York circuit data (Pecan Street Inc., 2019) into

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

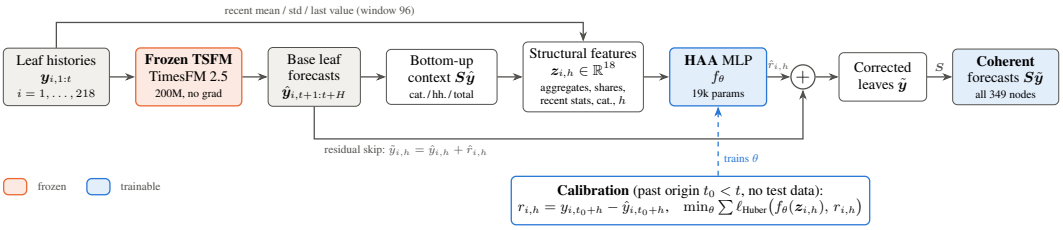


Figure 1: **Overview of hierarchy-aware residual adaptation.** A frozen foundation model forecasts every leaf independently. The base forecasts are aggregated through the summation matrix $\boldsymbol{S}$ to build category, household and system context. Together with recent temporal statistics and category identity, this context forms an 18-dimensional feature vector per leaf and horizon. A small MLP predicts a residual that is added to the base forecast. The corrected leaves are re-aggregated through $\boldsymbol{S}$, so all 349 nodes are coherent by construction. The adapter is trained only on residuals from a historical calibration origin (dashed).

a household–category–appliance hierarchy and trained one AutoARIMA (Hyndman & Khandakar, 2008) and one LightGBM (Ke et al., 2017) model per series, reconciled with MinT (Wickramasuriya et al., 2019). It reported two findings that motivate the present work (Figure 8 in Section A). First, aggregate levels are comparatively easy while appliance-level forecasts are hard for both models: symmetric MAPE rose from 0.12% (AutoARIMA, total) to 40–46% at the appliance level because the signals are sporadic and zero-inflated. Second, it recommended *hybrid* and *hierarchy-aware* models to close that gap. We take up both points. We replace per-series trained models with *frozen, pretrained* foundation models that need no per-series fitting. We also turn the hierarchy from a post-processing constraint into an *input* of a learned hybrid correction. The experiments reuse that study's households, observation period and five-category taxonomy, so the two studies form one continuous research thread on the same system.

This paper asks a narrow question: *Can explicit hierarchy information improve a frozen time-series foundation model, and if so, where in the hierarchy and through what mechanism?* We keep the backbone frozen for three reasons: it isolates the contribution of structure, it keeps adaptation cheap enough for utilities and researchers without large compute, and it matches how TSFMs are actually deployed as black-box forecasting services.

We propose the Hierarchy-Aware Residual Adapter (HAA; Figure 1). The frozen model forecasts every leaf. We aggregate these forecasts bottom-up through the summation matrix $\boldsymbol{S}$ to build hierarchy context for each leaf. A small MLP, trained on historical forecast errors, then predicts a residual correction per leaf. Coherence does not depend on anything the adapter learns: corrected leaves are re-aggregated through $\boldsymbol{S}$, so every level is coherent by construction (Proposition 1).

Contributions.

1. **Method.** A parameter-efficient, backbone-agnostic adapter that conditions frozen TSFM forecasts on multi-level structure while keeping coherence exact. It adds 19,073 parameters, about 0.01% of the 200M-parameter backbone (Section 4).

2. **Evidence.** A four-level, 349-node residential energy hierarchy with 218 sub-metered appliance series, built on the taxonomy of Sucaldito & Ostia (2025). Here HAA improves pooled RMSE over frozen TimesFM at every level, most at the system total (Section 6).

3. **Diagnosis.** A series-level, horizon-level and bias–variance analysis showing *why* the gains appear. Most of the improvement is sign-consistent bias removal that aggregation amplifies. At the leaves the adapter is often harmful, and physically invalid (negative) outputs are common (Section 7). A zero-cost non-negativity projection recovers part of this loss.

4. **Protocol.** A pre-registered set of rolling-origin, household-hold-out, hierarchy-permutation and cross-city experiments. These are needed before any claim of *structural generalization*, and we state clearly which parts are not yet done (Section 8).

2 Related Work

Time-series foundation models. Large pretrained forecasters either tokenize real values (Chronos; Ansari et al., 2024), use patched decoder-only transformers (TimesFM; Das et al., 2024), or train any-variate encoders (Moirai; Woo et al., 2024). Others build on or reprogram language models (Gruver et al., 2023; Jin et al., 2024; Zhou et al., 2023; Rasul et al., 2023; Garza & Mergenthaler-Canseco, 2023). Benchmarks such as GIFT-Eval (Aksu et al., 2024) evaluate these models one series at a time. Cross-series structure, when used at all, is usually limited to multivariate channel mixing, not *known* aggregation constraints. Our work is complementary: we keep the TSFM frozen and add structure externally.

Hierarchical forecasting and reconciliation. Bottom-up, top-down and optimal-combination methods (Hyndman et al., 2011), trace-minimization (MinT; Wickramasuriya et al., 2019) and their geometric interpretation (Panagiotelis et al., 2021) project incoherent base forecasts onto the coherent subspace; see Athanasopoulos et al. (2024) for a review. Learning-based variants regularize or relax unbiasedness (Ben Taieb & Koo, 2019), or train deep models end-to-end under coherence constraints (Rangapuram et al., 2021; Han et al., 2021; Paria et al., 2021; Kamarthi et al., 2022). Ben Taieb et al. (2021) study hierarchical probabilistic forecasting for smart-meter electricity demand. These methods either fit the base model jointly with the hierarchy or reconcile fixed forecasts linearly. HAA does neither. It learns a *nonlinear, structure-conditioned* correction for a frozen model and delegates coherence to bottom-up aggregation.

Parameter-efficient adaptation and stacking. Adapters (Houlsby et al., 2019) and low-rank updates (Hu et al., 2022) adapt large models with few parameters, but they sit *inside* the network and need gradient access. Our adapter works purely on outputs, like stacked generalization (Wolpert, 1992), so it applies to closed-weight or API-served forecasters.

Residential load forecasting. Appliance-level and household-level load is intermittent and heavy-tailed, which makes it hard to forecast (Hong & Fan, 2016; Kong et al., 2019). Public sub-metered corpora such as REDD (Kolter & Johnson, 2011), UK-DALE (Kelly & Knottenbelt, 2015) and Pecan Street Dataport (Pecan Street Inc., 2019) make it possible to build natural multi-level hierarchies. We use the last of these. On exactly this data, Sucaldito & Ostia (2025) compared a statistical model (AutoARIMA) with gradient-boosted trees (LightGBM) under MinT reconciliation. AutoARIMA was better at every level, and both models struggled at the appliance level. Our study is the direct successor of that work, moving from trained per-series models to frozen foundation models with a structure-conditioned adapter.

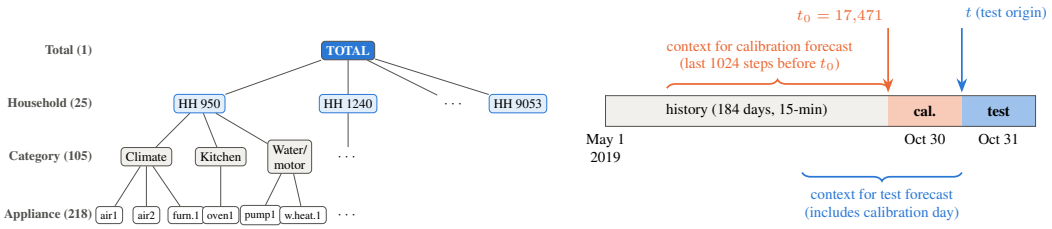
3 Problem Setup

Let $\boldsymbol{y}_t \in \mathbb{R}^m$ collect the m bottom-level (leaf) series at time t, and let $\boldsymbol{s}_t \in \mathbb{R}^n$ collect all $n \geq m$ nodes of the hierarchy. A *summation matrix* $\boldsymbol{S} \in \{0,1\}^{n \times m}$ satisfies $\boldsymbol{s}_t = \boldsymbol{S}\boldsymbol{y}_t$, where $S_{ji} = 1$ iff leaf i is a descendant of (or equal to) node j. Our hierarchy has four levels (Figure 2a):

$$i \;\rightarrow\; c(i) \;\rightarrow\; h(i) \;\rightarrow\; T, \tag{1}$$

where $c(i)$ is the household-specific end-use category of appliance i, $h(i)$ is its household, and T is the system total. With $m = 218$ leaves, 105 category nodes, 25 households and one total, $\boldsymbol{S} \in \{0,1\}^{349 \times 218}$ has 872 non-zeros (Section C).

Given history up to t, we want H-step forecasts $\tilde{\boldsymbol{s}}_{t+h}$, $h = 1, \ldots, H$, that are *accurate* at every level and *coherent*, meaning $\tilde{\boldsymbol{s}}_{t+h} \in \operatorname{span}(\boldsymbol{S})$. A foundation model g_ϕ with frozen weights ϕ produces leaf forecasts $\hat{\boldsymbol{y}}_{i,t+1:t+H} = g_\phi(\boldsymbol{y}_{i,1:t})$ independently for each i.



(a) Four-level hierarchy (excerpt). (b) Temporal protocol (not to scale).

Figure 2: **Structure and protocol.** (a) Each of the 218 active appliance circuits belongs to exactly one household-specific category node. (b) The adapter is fit on residuals of a frozen forecast made at a historical origin t_0 (one day before the test origin). The final 96 steps (Oct 31, 2019) are never seen during calibration.

4 Hierarchy-Aware Residual Adaptation

Structural context. Given base leaf forecasts $\hat{y}_{i,h}$ (we drop t for brevity), we compute bottom-up aggregates of the *forecasts*, not of observed data:

$$\hat{y}^{c}_{i,h} = \sum_{j:\, c(j)=c(i)} \hat{y}_{j,h}, \qquad \hat{y}^{hh}_{i,h} = \sum_{j:\, h(j)=h(i)} \hat{y}_{j,h}, \qquad \hat{y}^{T}_{h} = \sum_{j=1}^{m} \hat{y}_{j,h}, \tag{2}$$

together with the leaf's structural shares $\rho^{c}_{i,h} = \hat{y}_{i,h}/(|\hat{y}^{c}_{i,h}| + \epsilon)$, $\rho^{hh}_{i,h} = \hat{y}_{i,h}/(|\hat{y}^{hh}_{i,h}| + \epsilon)$ and $\rho^{T}_{i,h} = \hat{y}_{i,h}/(|\hat{y}^{T}_{h}| + \epsilon)$, with $\epsilon = 10^{-6}$. Temporal context comes from the last $L = 96$ observed values $\boldsymbol{x}_i$: mean, standard deviation, last value, and the mean and standard deviation normalized by $|\boldsymbol{x}_i|$. The feature vector is

$$\boldsymbol{z}_{i,h} = \big[\, \hat{y}_{i,h},\ \hat{y}^{c}_{i,h},\ \hat{y}^{hh}_{i,h},\ \hat{y}^{T}_{h},\ \rho^{c}_{i,h},\ \rho^{hh}_{i,h},\ \rho^{T}_{i,h},\ \mathrm{stats}(\boldsymbol{x}_i),\ h,\ \mathrm{onehot}(\mathrm{cat}(i)) \,\big] \in \mathbb{R}^{18}. \tag{3}$$

Household *identity* is deliberately left out. The adapter sees only *relational* information about a leaf's position in the hierarchy, so the same f_θ could in principle be applied to households, and hierarchies, that it never saw during training (Section 8).

Residual correction. The adapter $f_\theta : \mathbb{R}^{18} \to \mathbb{R}$ is a two-hidden-layer MLP (width 128, ReLU, dropout 0.1). The corrected forecast is

$$\tilde{y}_{i,h} = \hat{y}_{i,h} + f_\theta(\boldsymbol{z}_{i,h}), \tag{4}$$

so $f_\theta \equiv 0$ recovers the frozen backbone exactly. Because of this residual form, the adapter only has to learn *what the backbone systematically misses given the structure*.

Calibration without test leakage. We fix a historical origin $t_0 < t$ and run the frozen backbone on $\boldsymbol{y}_{i,1:t_0}$ to get $\hat{y}_{i,t_0+h}$. We build $\boldsymbol{z}_{i,h}$ from these forecasts and from $\boldsymbol{y}_{i,t_0-L+1:t_0}$, and form targets $r_{i,h} = y_{i,t_0+h} - \hat{y}_{i,t_0+h}$. We then minimize the Huber loss (Huber, 1964) $\mathcal{L}(\theta) = \frac{1}{mH}\sum_{i,h} \ell_\delta\big(f_\theta(\boldsymbol{z}_{i,h}) - r_{i,h}\big)$ with $\delta = 1$ using AdamW (Loshchilov & Hutter, 2019). At test time (Algorithm 1), the only inputs are the frozen test forecasts and the history up to t.

Proposition 1 (Coherence is independent of the adapter)**.** *For any θ, $\tilde{\boldsymbol{s}}_{t+h} := \boldsymbol{S}\tilde{\boldsymbol{y}}_{t+h}$ is coherent. Its error at any node j is the sum of the corrected errors of j's leaf descendants: $s_{j,t+h} - \tilde{s}_{j,t+h} = \sum_i S_{ji}\big(e_{i,h} - \hat{r}_{i,h}\big)$, where $e_{i,h} = y_{i,t+h} - \hat{y}_{i,h}$.*

Proof. Coherence follows because $\tilde{\boldsymbol{s}} \in \mathrm{span}(\boldsymbol{S})$ by definition. The error identity follows from linearity of $\boldsymbol{S}$ and $\boldsymbol{s} = \boldsymbol{S}\boldsymbol{y}$. □

Proposition 1 has a consequence that shapes all of our results. At an aggregate node with $n_j = \sum_i S_{ji}$ descendants, the *mean* of the corrections adds up linearly (a scale of n_j). Idiosyncratic leaf errors that are uncorrelated grow only like $\sqrt{n_j}$. A small, sign-consistent correction that barely matters at a single appliance can therefore dominate at the total ($n_T = 218$). Section 7 shows that this is exactly what happens.

Algorithm 1 HAA inference (one forecast origin t)

Require: leaf histories $\{\boldsymbol{y}_{i,1:t}\}_{i=1}^{m}$, summation matrix $\boldsymbol{S}$, frozen g_ϕ, trained f_θ
1: $\hat{\boldsymbol{y}}_{i,t+1:t+H} \leftarrow g_\phi(\boldsymbol{y}_{i,t-1023:t})$ for all i ▷ one batched call; no gradients
2: compute $\hat{y}^c, \hat{y}^{hh}, \hat{y}^T$ by summing $\hat{\boldsymbol{y}}$ along $\boldsymbol{S}$; compute shares ρ
3: compute stats$(\boldsymbol{y}_{i,t-95:t})$; assemble $\boldsymbol{z}_{i,h}$ ▷ non-finite values set to 0
4: $\tilde{y}_{i,h} \leftarrow \hat{y}_{i,h} + f_\theta(\boldsymbol{z}_{i,h})$ ▷ optional: $\tilde{y}_{i,h} \leftarrow \max(0, \tilde{y}_{i,h})$
5: **return** $\tilde{\boldsymbol{s}}_{t+h} = \boldsymbol{S}\,\tilde{\boldsymbol{y}}_{t+h}$ for $h = 1..H$ ▷ coherent at all 349 nodes

5 Experimental Setup

Data. Following Sucaldito & Ostia (2025), we use the Pecan Street Dataport New York sample (Pecan Street Inc., 2019): 25 households with circuit-level average power (kW) at 15-minute resolution, from May 1 to Oct 31, 2019 (17,664 steps per household; 441,599 rows in total). We remove non-consumption and rarely-populated channels (grid, solar, EV, battery, voltage legs, pool, and duplicate room circuits; 38 columns), fill missing values with zero, and map the remaining 39 appliance channels to the five end-use categories defined in that study (we additionally assign the dishwasher channel to kitchen appliances): climate control, kitchen appliances, laundry, lighting/fan/wall outlets, and water/motor loads (Table 7). Unlike the nominal 25×38=950-leaf layout of the earlier study, we keep only household–appliance pairs with non-zero total consumption, which gives $m = 218$ leaves. In the test window, 26.4% of leaf observations are exactly zero.

Split. The final $H = 96$ steps (24 h, Oct 31) form the test window. The adapter is calibrated at a single origin $t_0 = 17{,}471$, the day before the test day, which yields $218 \times 96 = 20{,}928$ training rows (Figure 2b).

Models. (i) *Naïve*: repeat the last value. (ii) *Seasonal naïve*: repeat the last 96 steps. (iii) *Chronos-T5-Base* (Ansari et al., 2024): median of 20 sample paths, per-series inference. Chronos recommends horizons of ≤ 64; we keep $H = 96$ for comparability and note this as a limitation. (iv) *TimesFM 2.5 (200M)* (Das et al., 2024): batched inference with context 1024, input normalization, continuous quantile head, flip invariance and positivity inference enabled. (v) *TimesFM* + HAA (ours): 19,073 parameters, 30 epochs, batch 512, learning rate 10^{-3}, weight decay 10^{-4}, seed 42. (vi) *TimesFM* + HAA + *clip*: (v) followed by the projection $\max(0, \cdot)$ at the leaves. This is a post-hoc analysis variant, added after we inspected (v) (Section 7.1). All methods are reconciled bottom-up, so every level is exactly coherent (we verified numerically that the coherence error is 0).

Metrics and inference. We report pooled RMSE, MAE, weighted absolute percentage error $\mathrm{WAPE} = \sum |e| / \sum |y|$ and R^2 at each level. MAPE over non-zero actuals and MASE with a 96-step seasonal scale (Hyndman & Koehler, 2006) are listed in Section E. For the three lower levels we report the fraction of series on which HAA beats TimesFM (*win rate*), a Wilcoxon signed-rank test on per-series RMSE, and a 95% percentile bootstrap CI for the pooled RMSE gain obtained by resampling *series* (2,000 draws). The total level is a single series of 96 correlated errors. There we use a Diebold–Mariano test with a HAC variance (lag 8; Diebold & Mariano, 1995) and a moving-block bootstrap (block 8; Künsch, 1989).

6 Results

Headline accuracy. Table 1 and Figure 4 summarize the results. Among the foundation models, TimesFM has lower RMSE than Chronos at every level. Chronos has the lowest leaf MAE and WAPE, and is worse by 9.8% RMSE at the total. Adding HAA to TimesFM lowers RMSE at every level: -2.0% (appliance), -1.1% (category), -5.1% (household) and -20.2% (total), and gives the best R^2 at the three lower levels. The gain grows with aggregation depth, as Proposition 1 predicts. At the total level, the improvement over TimesFM is significant (DM $= 5.06$, $p = 4.3 \times 10^{-7}$; block-bootstrap 95% CI for the RMSE gain $[0.91, 1.71]$ kW). Still, HAA does *not* beat seasonal naïve there (DM $= -0.22$, $p = 0.83$). The frozen foundation models are simply poor at the system level on this day (Figure 3); every model has negative R^2 at the total.

Table 1: **Accuracy at each hierarchy level** (24-h test window, bottom-up coherent). Lower is better except R^2. **Bold**: best among (i)–(v). Row (vi) is a post-hoc variant and is shown in gray for reference.

	Appliance (n=218)				**Category** (n=105)			
Model	RMSE	MAE	WAPE%	R^2	RMSE	MAE	WAPE%	R^2
(i) Naïve	0.2222	0.0450	85.6	−0.007	0.3235	0.0885	81.1	0.012
(ii) Seasonal naïve	0.2864	0.0583	110.8	−0.673	0.4171	0.1150	105.4	−0.642
(iii) Chronos	0.2075	**0.0377**	**71.7**	0.122	0.3045	0.0761	69.8	0.125
(iv) TimesFM	0.2053	0.0380	72.3	0.140	0.3004	**0.0755**	**69.3**	0.148
(v) TimesFM + HAA	**0.2013**	0.0518	98.5	**0.174**	**0.2970**	0.0978	89.7	**0.168**
(vi) + clip	0.2010	0.0487	92.6	0.176	0.2955	0.0937	85.9	0.176
	Household (n=25)				**Total** (n=1)			
Model	RMSE	MAE	WAPE%	R^2	RMSE	MAE	WAPE%	R^2
(i) Naïve	0.7274	0.3057	66.7	−0.087	5.445	4.490	39.2	−2.10
(ii) Seasonal naïve	0.8444	0.3855	84.2	−0.464	**4.608**	**3.512**	**30.7**	**−1.22**
(iii) Chronos	0.6800	0.2908	63.5	0.051	6.540	5.635	49.2	−3.48
(iv) TimesFM	0.6659	**0.2735**	**59.7**	0.090	5.959	5.106	44.6	−2.72
(v) TimesFM + HAA	**0.6321**	0.2950	64.4	**0.180**	4.755	3.750	32.8	−1.37
(vi) + clip	0.6254	0.2854	62.3	0.197	4.209	3.217	28.1	−0.85

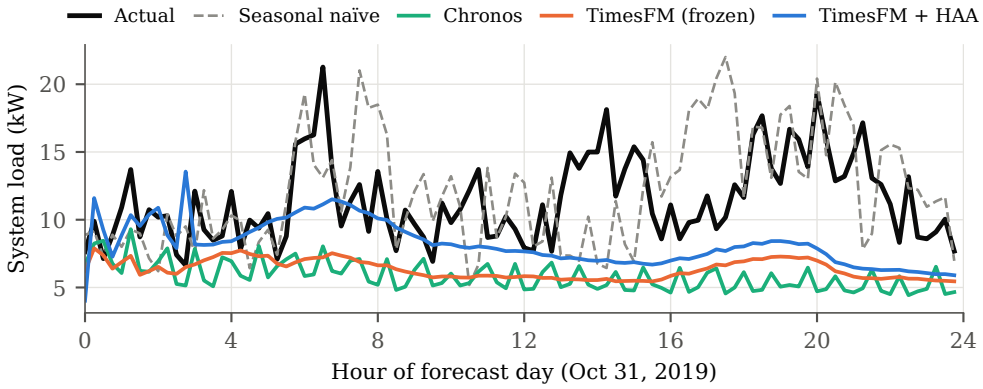


Figure 3: **System-total forecasts on the test day.** Both foundation models badly under-forecast the evening load. HAA lifts the level of the TimesFM forecast, especially in the first half of the day, but does not recover the demand peaks. Seasonal naïve tracks the peak timing because the previous day had a similar shape.

7 Analysis: Where Do the Gains Come From?

Pooled gains hide heterogeneity. Pooled RMSE is dominated by high-variance series. At the series level (Figure 5, Table 2), HAA beats TimesFM on only 24% of appliances, 30% of categories and 36% of households. Median per-series RMSE *increases* (e.g. 0.033→0.036 kW for appliances), and the Wilcoxon test favors TimesFM at the appliance ($p < 10^{-15}$) and category ($p = 5 \times 10^{-5}$) levels. Bootstrap CIs over series for the pooled gain include zero at all three lower levels. At the household level, one high-load household (950) accounts for most of the gain; its RMSE drops from 2.12 to 1.87 kW (Figure 11 in Section F). The lower-level improvements should therefore be read as *not yet established*.

The adapter is mainly a bias corrector. TimesFM under-forecasts at every level. At the total, its mean bias is −5.10 kW, which is 44.5% of the mean test load of 11.45 kW (Figure 6a). The mean learned correction is small per leaf (+0.0085 kW), but it is sign-consistent. Summed over 218 leaves it shifts the total by $218 \times 0.0085 \approx 1.85$ kW, which matches the observed bias change

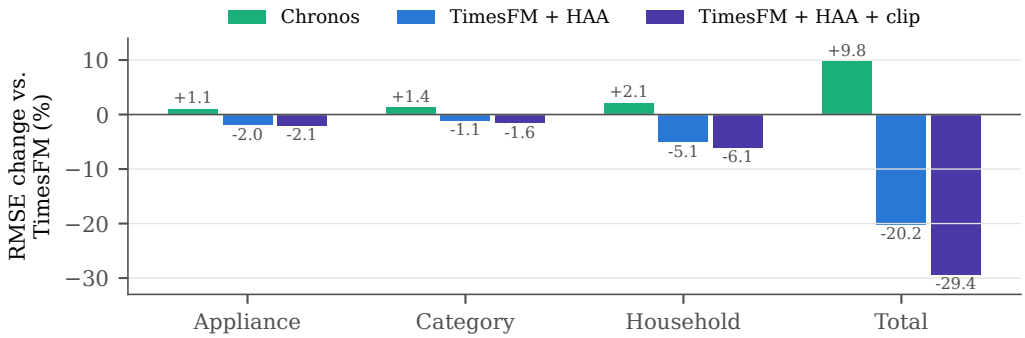


Figure 4: **Relative RMSE change versus frozen TimesFM** (negative is better). The benefit of hierarchy-aware correction grows with aggregation depth.

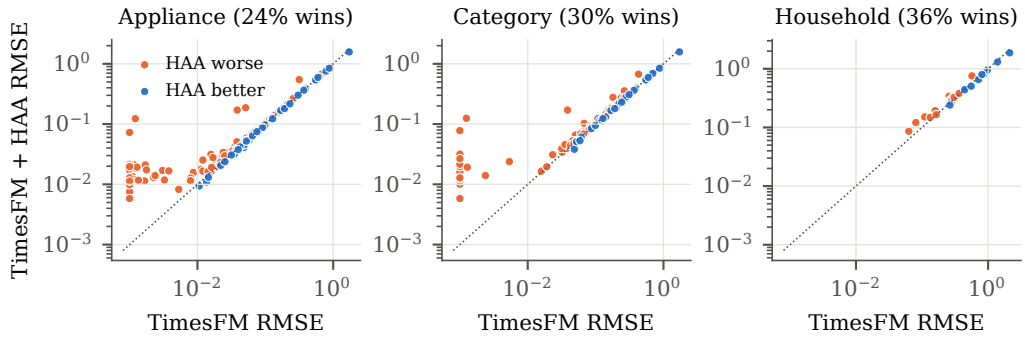


Figure 5: **Per-series RMSE, TimesFM vs. TimesFM + HAA** (log scale; values below 10^{-3} are clipped for display). Points above the diagonal are series where the adapter hurts. Pooled gains are driven by a minority of high-load series, while many near-idle series get worse.

($-5.10 \rightarrow -3.25$ kW) exactly. Table 3 decomposes total-level MSE into squared bias and error variance. *All* of HAA's improvement comes from the bias term (26.0→10.5); the variance term actually *grows* (9.5→12.1). Seasonal naïve has almost no bias but high variance. The corrections also differ by category (Figure 6b): they are largest and most positive for water/motor loads (well pumps, water heaters) and slightly negative for lighting and outlets.

Leaf-level side effects. The positive shift that helps aggregates hurts intermittent leaves. On zero-load observations (26% of leaf targets), the adapter still adds $+0.0195$ kW on average. As a result, leaf MAE increases by 36% and MAPE on non-zero actuals nearly triples (Table 4). The adapter also produces negative loads in 31% of leaf predictions, compared with 5.5% for TimesFM, because it adds residuals of either sign to near-zero base forecasts. RMSE and MAE therefore give *different* answers at the leaves. RMSE rewards the adapter for shrinking large errors on a few high-load circuits; MAE penalizes it for adding small errors on the many idle ones.

Horizon profile. Figure 7 splits the 24-hour window into 4-hour blocks. At the three lower levels, HAA is *worse* than TimesFM in the first block (00:00–04:00), when the backbone's short-range forecast is already good, and better in every later block. At the total level, HAA beats TimesFM in all six blocks, but it beats seasonal naïve in only three of them (04:00–12:00 and 16:00–20:00).

7.1 A Zero-Cost Fix: Non-Negativity Projection

Loads are non-negative, so we project corrected leaves onto $\mathbb{R}_{\geq 0}$ *before* aggregation, which preserves coherence (Algorithm 1, line 4). This post-hoc variant needs no retraining. It lowers RMSE at every level (total: $4.755 \rightarrow 4.209$ kW, a 29.4% gain over TimesFM, which now also beats sea-

Table 2: **Statistical assessment of TimesFM + HAA vs. TimesFM.** Positive gain means HAA is better. Lower levels: series-level tests. Total: temporal tests on 96 errors.

Level	Pooled RMSE gain	95% CI	Win rate	Median RMSE TFM → HAA	Test (p-value)
Appliance	+0.0040	[−0.0066, 0.0131]	24% (52/218)	0.033 → 0.036	W: 2×10^{-16} †
Category	+0.0034	[−0.0144, 0.0181]	30% (32/105)	0.077 → 0.088	W: 5×10^{-5} †
Household	+0.0338	[−0.0206, 0.0807]	36% (9/25)	0.297 → 0.327	W: 0.56
Total	+1.2032	[0.906, 1.706]	—	—	DM: 4×10^{-7}

W: Wilcoxon signed-rank; DM: Diebold–Mariano. † Significant in favor of *TimesFM* (lower median per-series RMSE).

Table 3: **Total-level MSE decomposition** (kW^2). MSE = bias^2 + error variance.

Model	MSE	Bias^2	Var	Bias %
Seasonal naïve	21.23	0.44	20.79	2%
Chronos	42.78	31.44	11.33	73%
TimesFM	35.50	26.01	**9.49**	73%
+ HAA	22.61	10.54	12.07	47%
+ HAA + clip	**17.71**	6.60	11.11	37%

Table 4: **Leaf-level side effects.** Share of negative leaf predictions and mean correction on zero-load observations.

Quantity	TimesFM	+ HAA
Negative leaf predictions	5.5%	31.3%
Mean $\lvert\hat{r}\rvert$ (kW)	—	0.0195
Mean $\hat{r}$ when y=0 (kW)	—	+0.0195
MAPE$_{\neq0}$, appliance	270%	771%
Median MASE, appliance	0.55	0.89

sonal naïve's 4.608), recovers part of the leaf-level MAE loss, and raises total R^2 from -1.37 to -0.85 (Table 1, row vi). The same projection barely changes TimesFM ($5.959 \rightarrow 5.948$), which confirms that the benefit comes from interacting with the adapter's corrections. Because we designed this variant after seeing test results, it should be validated under the protocol in Section 8 before it is treated as a method component. A natural next step is to build non-negativity into f_θ, for example by predicting $\max(-\hat{y}_{i,h}, \hat{r}_{i,h})$.

8 Limitations and Pre-Registered Protocol

We state the scope of our evidence plainly. **(1) Single origin.** Both the calibration and the test use one day each; a single atypical day, such as a Halloween evening, could drive the results. **(2) Single dataset.** One city, one season (May–Oct), 25 homes. **(3) No ablations yet.** We do not know which feature group (aggregates, shares, temporal statistics or category) carries the signal, and a structure-free adapter might do as well by learning a global bias. **(4) No structural generalization test.** The adapter was trained and tested on the same households. **(5) Point forecasts only**, and Chronos is run beyond its recommended horizon. **(6) Baselines.** We do not yet re-run the AutoARIMA and LightGBM pipeline of Sucaldito & Ostia (2025) under the present split and metrics. Its published numbers use a different horizon, series set and metrics, so they are not directly comparable to ours. We also do not compare against MinT reconciliation (Wickramasuriya et al., 2019), fine-tuned TSFMs, or trained deep forecasters (Oreshkin et al., 2020; Nie et al., 2023; Zeng et al., 2023).

We therefore frame *structural generalization* as a hypothesis, not a result, and pre-register the experiments needed to test it (Table 6 in Section B). The most important controls are E8 and E9. If a structure-free, bias-only adapter (E8) or an adapter trained on a *randomly permuted* hierarchy (E9) matches HAA, the gains should be attributed to calibration rather than to structure. E10–E12 test whether a household-identity-free adapter transfers to unseen households, unseen category compositions and a different city (e.g. Pecan Street Austin). All results should be reported as mean $\pm$ standard deviation over rolling origins (Tashman, 2000).

9 Conclusion

Explicit hierarchy information can improve a frozen time-series foundation model cheaply and without breaking coherence. In our residential energy hierarchy, a 19k-parameter residual adapter cut system-level RMSE by 20%, and by 29% with a non-negativity projection. The mechanism, how-

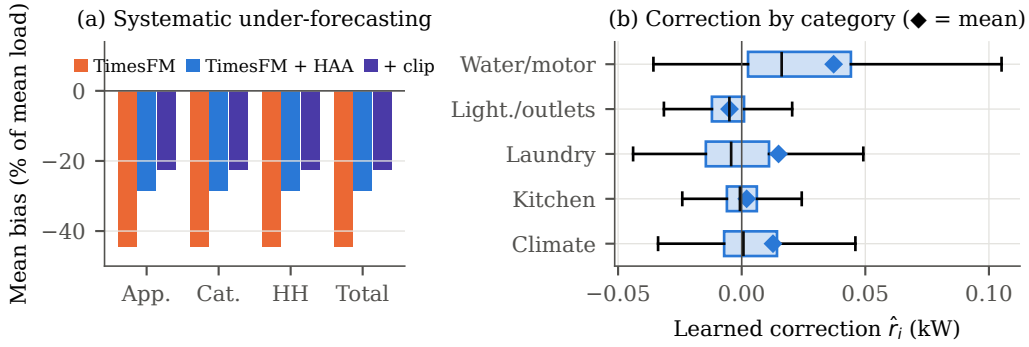


Figure 6: **What the adapter learns.** (a) Mean bias as a percentage of mean load: TimesFM under-forecasts by ≈45% at every level, and HAA removes roughly a third of this. (b) Distribution of the learned leaf corrections by category (whiskers: 1.5 IQR).

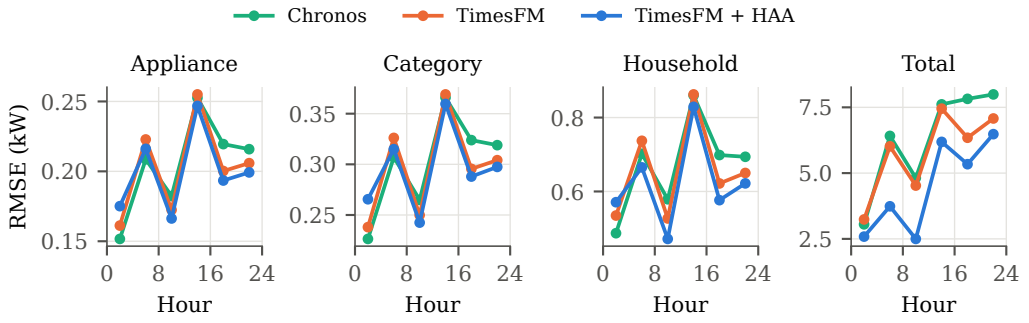


Figure 7: **RMSE by 4-hour block of the forecast horizon.** The adapter trades a small loss at short lead times for gains at longer lead times.

ever, is simpler than "learning structure". The adapter learns small, sign-consistent corrections to a systematically under-forecasting backbone, and bottom-up aggregation amplifies them. At the leaves the same corrections often hurt. This suggests a design principle for structure-aware TSFM adaptation: *corrections should be judged at the level where they are consumed, and leaf-level validity (non-negativity, intermittency) must be built in.* The results also echo the earlier finding of Sucaldito & Ostia (2025) that aggregates are easy and appliances are hard. A frozen foundation model with a structural adapter moves the aggregate levels forward, but the zero-inflated leaves remain the open problem in this research thread. Whether hierarchy-aware adapters show true structural generalization across households and cities is still open, and our protocol is designed to answer it.

Reproducibility Statement

All preprocessing steps, the category mapping (Table 7), hyperparameters (Section 5, Section D) and the random seed (42) are specified. We will release the notebook, the saved frozen forecasts for both backbones, the trained adapter checkpoint and all per-node evaluation files. The frozen backbones are public checkpoints (`amazon/chronos-t5-base`, `google/timesfm-2.5-200m-pytorch`). The dataset is available from Pecan Street Dataport under its academic license, and the preprocessing and category taxonomy follow Sucaldito & Ostia (2025). All statistics in Section 7 are computed directly from the released evaluation files.

Ethics Statement

Circuit-level household energy data can reveal occupancy and daily routines. We use anonymized household identifiers only as provided by the data licensor and report no information beyond aggregate forecasting accuracy. Better load forecasts can support grid efficiency and demand response. Systematic under-forecasting, which we document, could lead to under-provisioning if such models were deployed without the checks we recommend.

Use of Large Language Models

An LLM-based assistant was used extensively in preparing this submission. It (i) drafted and edited all sections of the manuscript; (ii) identified related work and compiled the references, which the authors checked; (iii) produced the figures and diagrams; (iv) ran the post-hoc statistical analyses in Section 7 (bootstrap intervals, Wilcoxon and Diebold–Mariano tests, bias–variance decomposition) from the saved evaluation files, and proposed the non-negativity projection variant (Section 7.1) and the pre-registered protocol (Section B); and (v) wrote Proposition 1 and its proof. The authors ran all model training and inference, checked the reported numbers against the code outputs, and take full responsibility for the content. No synthetic data were generated; all experiments use the real Pecan Street measurements.

References

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-Eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

George Athanasopoulos, Rob J. Hyndman, Nikolaos Kourentzes, and Anastasios Panagiotelis. Forecast reconciliation: A review. *International Journal of Forecasting*, 40(2):430–456, 2024.

Souhaib Ben Taieb and Bonsoo Koo. Regularized regression for hierarchical forecasting without unbiasedness conditions. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1337–1347, 2019.

Souhaib Ben Taieb, James W. Taylor, and Rob J. Hyndman. Hierarchical probabilistic forecasting of electricity demand with smart meter data. *Journal of the American Statistical Association*, 116 (533):27–43, 2021.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Francis X. Diebold and Roberto S. Mariano. Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3):253–263, 1995.

Azul Garza and Max Mergenthaler-Canseco. TimeGPT-1. *arXiv preprint arXiv:2310.03589*, 2023.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Xing Han, Sambarta Dasgupta, and Joydeep Ghosh. Simultaneously reconciled quantile forecasting of hierarchically related time series. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 190–198, 2021.

Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 2790–2799, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

Rob J. Hyndman and Yeasmin Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27(3):1–22, 2008.

Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

Rob J. Hyndman, Roman A. Ahmed, George Athanasopoulos, and Han Lin Shang. Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis*, 55(9): 2579–2589, 2011.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Harshavardhan Kamarthi, Lingkai Kong, Alexander Rodríguez, Chao Zhang, and B. Aditya Prakash. PROFHiT: Probabilistic robust forecasting for hierarchical time-series. *arXiv preprint arXiv:2206.07940*, 2022.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jack Kelly and William Knottenbelt. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific Data*, 2:150007, 2015.

J. Zico Kolter and Matthew J. Johnson. REDD: A public data set for energy disaggregation research. In *Workshop on Data Mining Applications in Sustainability (SIGKDD)*, 2011.

Weicong Kong, Zhao Yang Dong, Youwei Jia, David J. Hill, Yan Xu, and Yuan Zhang. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Transactions on Smart Grid*, 10(1):841–851, 2019.

Hans R. Künsch. The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, 17(3):1217–1241, 1989.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations (ICLR)*, 2020.

Anastasios Panagiotelis, George Athanasopoulos, Puwasala Gamakumara, and Rob J. Hyndman. Forecast reconciliation: A geometric view with new insights on bias correction. *International Journal of Forecasting*, 37(1):343–359, 2021.

Biswajit Paria, Rajat Sen, Amr Ahmed, and Abhimanyu Das. Hierarchically regularized deep forecasting. *arXiv preprint arXiv:2106.07630*, 2021.

Pecan Street Inc. Pecan street dataport. `https://www.pecanstreet.org/dataport/`, 2019. 15-minute circuit-level data, New York sample.

Syama Sundar Rangapuram, Lucien D. Werner, Konstantinos Benidis, Pedro Mercado, Jan Gasthaus, and Tim Januschowski. End-to-end learning of coherent probabilistic forecasts for hierarchical time series. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8832–8843, 2021.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.

Michel Angelo R. Sucaldito and Conrado F. Ostia, Jr. Hierarchical forecasting of load consumption from residential dwellings with AutoARIMA and LightGBM approach. In *2025 12th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp. 66–71. IEEE, 2025. doi: 10.1109/EECSI67060.2025.11290142.

Leonard J. Tashman. Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*, 16(4):437–450, 2000.

Shanika L. Wickramasuriya, George Athanasopoulos, and Rob J. Hyndman. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526):804–819, 2019.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

A Relation to the Preceding Study

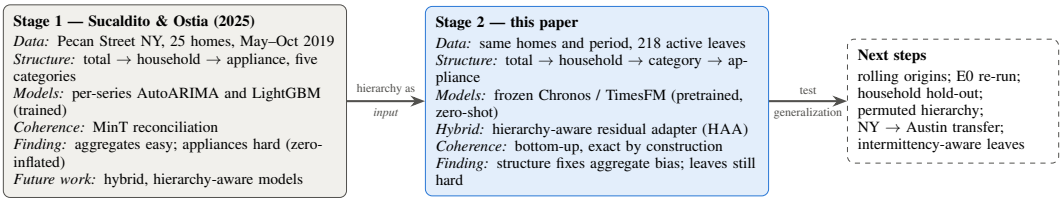


Figure 8: **Research lineage.** The present study takes up the hybrid, hierarchy-aware direction recommended by Sucaldito & Ostia (2025) on the same residential system, and sets out the experiments that remain.

Table 5: **Setup of the preceding study and of this paper.** Reported numbers are *not* directly comparable because horizon, series set, reconciliation and metrics differ; experiment E0 (Table 6) is planned to align them.

	Sucaldito & Ostia (2025)	This paper
Households / period	25 / May 1–Oct 31, 2019	25 / May 1–Oct 31, 2019
Resolution	15 minutes	15 minutes
Leaf series	950 (25 homes × 38 channels)	218 (active pairs only)
Hierarchy	total, household, appliance; five categories	total, 25 households, 105 categories, 218 appliances
Forecasters	AutoARIMA, LightGBM (trained per series)	Chronos, TimesFM (frozen) + HAA (19k params)
Coherence	MinT reconciliation	bottom-up (exact)
Metrics	RMSE, sMAPE	RMSE, MAE, WAPE, R^2, $\text{MAPE}_{\neq 0}$, MASE
Key finding	AutoARIMA best at all levels; appliance sMAPE 40–46%	HAA cuts total RMSE 20% (29% with clip); leaf gains not significant

B Pre-Registered Experimental Protocol

Table 6: **Experimental protocol and status.** ✓: reported in this paper; ○: pre-registered, not yet run.

ID	Backbone	Structure given to adapter	Purpose	Status
E0	AutoARIMA / LightGBM	MinT (as in prior study)	continuity with prior work	○
E1–E3	S. naïve/Chronos/TimesFM	none	baselines	✓
E4	TimesFM	bottom-up only	coherent baseline	✓
E5	TimesFM + HAA	full hierarchy	proposed	✓
E6	TimesFM + HAA	category / household / total only	level ablation	○
E7	TimesFM + HAA	no temporal stats; no shares	feature ablation	○
E8	TimesFM + plain MLP	none (leaf + temporal only)	structure-free control	○
E9	TimesFM + HAA	randomly permuted $\boldsymbol{S}$	structural control	○
E10	TimesFM + HAA	leave-households-out	unseen households	○
E11	TimesFM + HAA	held-out category compositions	unseen structure	○
E12	TimesFM + HAA	train NY → test Austin	cross-city transfer	○
All	—	3–5 rolling origins; MinT; Chronos-Bolt	robustness	○

C HIERARCHY DETAILS

Table 7: **Category mapping** (39 retained appliance channels) and the number of active leaf series per category.

Category	Appliance channels	Leaves
Climate control	air1, air2, airwindowunit1, circpump1, furnace1, furnace2, heater1–3	43
Lighting, fans & outlets	bathroom1, bedroom1, diningroom1, garage1, housefan1, kitchen1, kitchenapp1–2, lights_plugs1–4, livingroom1, office1, utilityroom1, venthood1	71
Laundry	clotheswasher1, clotheswasher_dryg1, drye1	36
Kitchen appliances	freezer1, microwave1, oven1, refrigerator1, dishwasher1	40
Water / motor loads	jacuzzi1, pump1, sewerpump1, sumppump1, waterheater1, wellpump1	28
Total		218

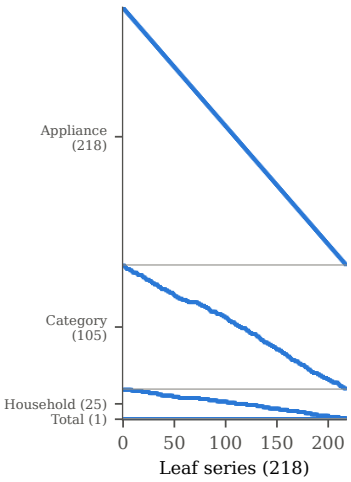


Figure 9: **Sparsity pattern of the summation matrix $\boldsymbol{S} \in \{0,1\}^{349\times 218}$** (872 non-zeros). Rows are ordered appliance (identity block), category, household and total. Households contribute between 4 and 20 leaves (median 8), and category nodes between 1 and 9 leaves (median 2).

D ADAPTER DETAILS

Input: 13 numeric features (base forecast; category, household and total forecast aggregates; three shares; recent mean, standard deviation and last value; normalized mean and standard deviation; horizon index $h \in \{1,\dots,96\}$) plus a 5-way category one-hot, for 18 inputs in total. Non-finite values are replaced by 0. Features are *not* standardized, which is a likely source of optimization difficulty given the scale gap between leaf values ($\sim$0.01 kW) and total aggregates ($\sim$10 kW) and should be fixed in future versions. Architecture: Linear(18,128)–ReLU–Dropout(0.1)–Linear(128,128)–ReLU–Linear(128,1), with $18\cdot128+128+128\cdot128+128+128+1 = 19{,}073$ parameters. Training: AdamW, learning rate 10^{-3}, weight decay 10^{-4}, batch 512, 30 epochs, Huber loss ($\delta = 1$), one NVIDIA GPU. Training takes seconds. Chronos inference took about 70 minutes for 218 series ($\approx$20 s per series with 20 sample paths); TimesFM inference is a single batched call.

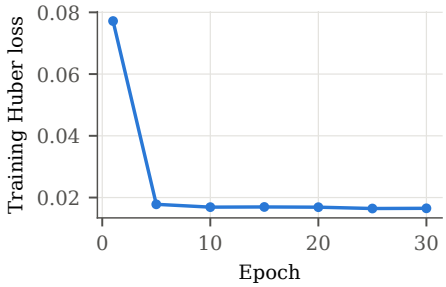


Figure 10: **Adapter training loss** at the logged epochs (1, 5, 10, …, 30). The loss falls from 0.0772 to 0.0178 within five epochs and levels off at about 0.0165, which suggests limited capacity to fit calibration residuals beyond a level shift.

E Additional Metrics

Table 8: **Scale-free metrics.** MAPE over non-zero actuals (%) and MASE with a 96-step seasonal-naïve scale computed on the training history. We report the median MASE over series because a few near-idle series with tiny seasonal scale make the mean unstable (mean appliance MASE for HAA is 257).

	$MAPE_{\neq 0}$ (%)				Median MASE			
Model	App.	Cat.	HH	Total	App.	Cat.	HH	Total
Naïve	648	698	249	35.0	0.60	0.70	0.68	1.25
Seasonal naïve	787	1014	203	31.2	0.92	0.97	0.82	**0.98**
Chronos	**246**	283	**76**	45.5	**0.53**	0.60	0.63	1.57
TimesFM	270	**272**	86	40.9	0.55	**0.53**	**0.61**	1.42
TimesFM + HAA	771	958	175	**29.8**	0.89	0.82	0.68	1.04

F Additional Qualitative Results

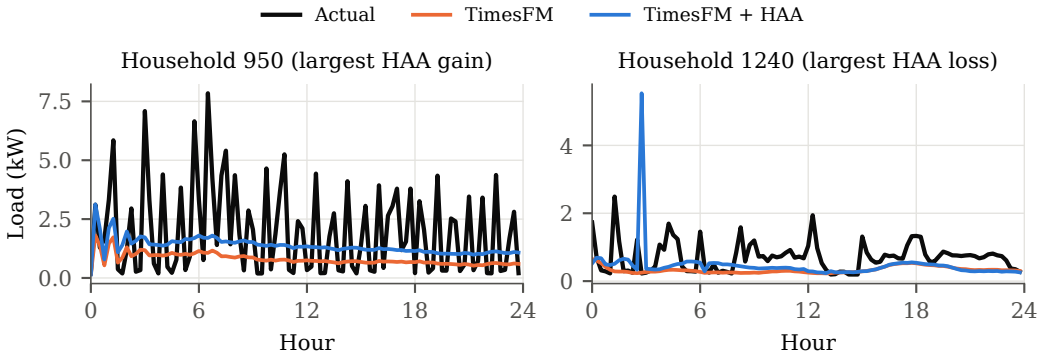


Figure 11: **Households with the largest gain (950) and the largest loss (1240) from HAA.** In household 950 the adapter raises a flat, under-forecasting TimesFM trajectory toward the mean of a highly cyclic load. In household 1240 it introduces a spurious 5 kW spike near hour 3, a failure mode of an unconstrained, unstandardized residual model.