REQUA: RESIDUAL-GROUNDED QUERY ADAPTATION FOR COVARIATE-AWARE FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent time-series foundation models (TSFMs) increasingly support native multivariate forecasting and have demonstrated substantial gains over univariate modeling on multivariate tasks. However, they always treat all covariates equally, which implies that adapting to downstream tasks, where the focus is on predicting a subset of target covariates, may require fundamentally altering the model's internal structure. It is reasonable to expect that such adaptation cannot be achieved under the conventional fine-tuning paradigm on the pretrained models. Therefore, in this paper, we introduce ReQuA, a lightweight covariate-aware adaptation framework that keeps the TSFM and its original prediction pathway frozen. ReQuA constructs queries conditioned on the target history and forecast horizon to aggregate task-relevant auxiliary information. The backbone's forecast-generating latent states attend to this representation to obtain a residual correction, whose magnitude is controlled by a single coefficient calibrated on held-out data. Experiments on eight widely used time-series datasets demonstrate that ReQuA delivers substantial and consistent improvements across frozen TSFM backbones with few additional parameters and little computational overhead, while global calibration effectively suppresses harmful corrections.

1 INTRODUCTION

Forecasting systems rarely observe a target series in isolation. Energy demand is shaped by weather and calendar effects, traffic evolves jointly across sensors, and industrial measurements interact through shared physical processes. Exploiting such cross-variable structure has therefore become a central objective in modern forecasting. Supervised architectures address this problem through patch-wise temporal representations, variate tokens, or explicit endogenous–exogenous attention (Nie et al., 2023; Liu et al., 2024a; Wang et al., 2024). While effective, these models are typically trained separately for each dataset and deployment setting, limiting their ability to reuse knowledge across domains.

Time-series foundation models (TSFMs) address this limitation by pretraining on large and heterogeneous collections of time series. Models learn reusable temporal priors that can transfer to unseen datasets with little or no task-specific training (Das et al., 2024; Ansari et al., 2024; Woo et al., 2024; Liu et al., 2024b; Goswami et al., 2024). However, most early TSFMs emphasized univariate or channel-independent forecasting, limiting their ability to exploit cross-variable information directly at inference time.

Full fine-tuning can adapt a pretrained forecaster to downstream datasets. However, it requires updating a large parameter set, increasing training costs and potentially disrupting transferable representations. Parameter-efficient adaptation provides a lightweight alternative to full fine-tuning by keeping most or all pretrained parameters frozen (Houlsby et al., 2019; Hu et al., 2022). Existing covariate-aware TSFM adapters mainly focus on introducing external information into frozen forecasters (Dange & Sarawagi, 2025; Han et al., 2026; Qin et al., 2025). These approaches demonstrate the value of auxiliary information, but most address settings in which covariates are introduced outside the backbone's native cross-variable interaction pathway.

Recent models such as Chronos-2 incorporate multivariate and covariate-informed forecasting directly into the pretrained backbone, allowing cross-variable interaction to be learned during large-scale pretraining (Ansari et al., 2025). Because pretraining must accommodate heterogeneous vari-

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.

able sets and different target and covariate assignments across datasets, shared cross-variable interaction is a natural design choice. However, **such shared interaction does not explicitly distinguish the asymmetric downstream roles of prediction targets and auxiliary evidence.** Fine-tuning these shared layers may improve the transfer of information from auxiliary variables to the target, but can also alter other interactions across variables learned during pretraining. Moreover, because part of the auxiliary information is already captured by the backbone, naively injecting it again can produce redundant or harmful corrections. This motivates an adaptation mechanism that selectively transfers complementary auxiliary information to the target without modifying the backbone's shared cross-variable interactions.

We address this gap with **ReQuA**, a residual-grounded query adapter for frozen multivariate TSFMs. ReQuA constructs target- and horizon-conditioned queries to retrieve relevant information from auxiliary-variable states, and uses the resulting fixed-size representation to correct only the forecast-generating latent states of the target. A training-only residual-grounding objective directs the aggregated representation toward errors left by the frozen backbone, while a lightweight calibration stage controls correction strength through a single global coefficient. Our contributions are summarized as follows:

- To our knowledge, ReQuA is the first parameter-efficient adaptation framework explicitly designed for multivariate TSFMs, and formulates adaptation as a complementary latent correction.
- We propose a target- and horizon-conditioned Query Aggregator that separates **target-conditioned retrieval** from auxiliary-variable representation. It supports a variable number of auxiliary series and learns residual-grounded corrections that modify only the target's forecast-generating latent states, while keeping the TSFM and its pretrained cross-variable interactions frozen.
- We introduce lightweight global calibration and demonstrate consistent gains across eight datasets and multiple TSFM backbones. On Chronos-2, ReQuA achieves average improvements of **8.69%** in MAE and **19.50%** in MSE with little additional inference cost, and is the only method with **no negative** MAE or MSE cases in the ten-dataset ablation.

2 Related Work

2.1 Task-Specific Multivariate and Covariate-Aware Forecasting

Task-specific multivariate forecasters primarily differ in how they model temporal structure and cross-variable dependencies. Early approaches focused on improving temporal representations through sparse attention, decomposition, frequency-domain modeling, patching, or simple linear mappings (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022; Nie et al., 2023; Zeng et al., 2023). Later architectures placed greater emphasis on explicit interaction across variables. Crossformer models dependencies across dimensions through cross-dimension attention (Zhang & Yan, 2023), while iTransformer represents individual variates as tokens and performs attention across them (Liu et al., 2024a).

Covariate-aware forecasting further introduces asymmetric interaction between prediction targets and auxiliary variables. TFT performs gated variable selection to control the contribution of different inputs (Lim et al., 2021), TiDE jointly encodes targets and dynamic covariates (Das et al., 2023), and TimeXer retrieves exogenous information through variate-wise cross-attention (Wang et al., 2024). These methods establish selective cross-variable interaction as an important component of multivariate forecasting, but the corresponding interaction mechanisms are learned separately for each downstream dataset.

2.2 Multivariate Time-Series Foundation Models

Time-series foundation models replace dataset-specific training with large-scale pretraining of reusable forecasting representations. Early TSFMs, including Lag-Llama, TimesFM, Chronos, Timer, MOMENT, and Time-MoE, focused on univariate or channel-independent forecasting (Rasul et al., 2023; Das et al., 2024; Ansari et al., 2024; Liu et al., 2024b; Goswami et al., 2024; Shi et al.,

2025). They transferred general temporal forecasting priors across datasets, while cross-variable interaction was typically not part of the pretrained backbone.

Recent TSFMs extend pretraining from temporal modeling to cross-variable interaction. Moirai adopts an any-variate design (Woo et al., 2024), Chronos-2 introduces grouped attention for information exchange among related targets and covariates (Ansari et al., 2025), and TimesFM-3 alternates temporal and variate attention to support multiple targets and covariates (Jain & Sen, 2026). As a result, these models transfer both temporal and cross-variable patterns.

2.3 Parameter-Efficient and Covariate-Aware Adaptation

Parameter-efficient adaptation reduces downstream training cost by updating only a small set of parameters while keeping the pretrained model mostly or entirely frozen. Representative approaches include adapter modules, low-rank updates, and learned prompts (Houlsby et al., 2019; Hu et al., 2022; Li & Liang, 2021; Lester et al., 2021). In time-series forecasting, related approaches adapt pretrained models through restricted fine-tuning, input reprogramming, or low-rank updates (Zhou et al., 2023; Jin et al., 2024; Gupta et al., 2024).

Covariate-aware TSFM adaptation focuses on introducing auxiliary information without updating the pretrained forecasting backbone. TFMAdapter combines covariates with pretrained forecasts (Dange & Sarawagi, 2025); UniCA homogenizes heterogeneous covariates before fusion (Han et al., 2026); CoRA performs causality-oriented weighting before condition injection (Qin et al., 2025); and UniCast incorporates multimodal context into a frozen TSFM (Park et al., 2025).

Existing methods mainly address how external information can be added to a frozen forecaster. For TSFMs with pretrained cross-variable interaction, however, adaptation must also account for information already captured by the backbone. This leaves the complementary adaptation of multivariate TSFMs comparatively underexplored.

2.4 Query-Based Representation Bridging

Learnable queries provide a fixed-size interface for extracting information from variable-size representations. Set Transformer and Perceiver use aggregation queries to summarize large or variable-size inputs (Lee et al., 2019; Jaegle et al., 2021), while DETR and Slot Attention show that learned queries can specialize toward distinct task-relevant structures (Carion et al., 2020; Locatello et al., 2020). Query-based interfaces have also been used to bridge pretrained modules: Flamingo and BLIP-2 compress visual features into compact representations passed to frozen language models (Alayrac et al., 2022; Li et al., 2023). This fixed-capacity design naturally supports multivariate forecasting with a variable number of auxiliary series, motivating a compact query bottleneck before interaction with a frozen forecasting backbone.

3 Preliminaries

3.1 Multivariate Time Series Forecasting

Consider a multivariate time-series panel containing one target variable and multiple auxiliary variables. At forecasting time t, let y denote the target variable and let $\{x^{(k)}\}_{k=1}^{K}$ denote the remaining K variables. The historical context of length C observed by the model is written as

$$\mathbf{U}_{t-C+1:t} = \left[\mathbf{y}_{t-C+1:t}, \mathbf{x}^{(1)}_{t-C+1:t}, \ldots, \mathbf{x}^{(K)}_{t-C+1:t}\right] \in \mathbb{R}^{C\times(K+1)}, \tag{1}$$

where $\mathbf{y}_{t-C+1:t} \in \mathbb{R}^{C}$ denotes the historical observations of the target variable. We further define

$$\mathbf{U}^{\text{aux}}_{t-C+1:t} = \left[\mathbf{x}^{(1)}_{t-C+1:t}, \ldots, \mathbf{x}^{(K)}_{t-C+1:t}\right] \in \mathbb{R}^{C\times K} \tag{2}$$

as the historical observations of the auxiliary variables. Given a forecast horizon h, the objective is to predict the future target sequence

$$\mathbf{y}_{t+1:t+h} = [y_{t+1}, \ldots, y_{t+h}] \in \mathbb{R}^{h}. \tag{3}$$

Accordingly, the forecasting task considered in this work can be written as

$$\hat{\mathbf{y}}_{t+1:t+h} = F_\theta\big(\mathbf{y}_{t-C+1:t}, \mathbf{U}^{\text{aux}}_{t-C+1:t}; h\big)\,. \tag{4}$$

We adopt a **target-centric multivariate forecasting setting**. For a given multivariate panel, different variables may be selected as the target in turn, while the remaining variables are treated as auxiliary information. Only one target variable is predicted at a time. This formulation accommodates datasets with different numbers of variables.

3.2 Multivariate Time-Series Foundation Model Backbone

Let F_θ denote a pretrained multivariate TSFM with backbone hidden dimension D. At an intermediate layer ℓ, we denote the latent states associated with the target history, the k-th auxiliary variable, and the forecast-generating positions as

$$\mathbf{H}^{(\ell)}_y \in \mathbb{R}^{P_y \times D}, \qquad \mathbf{H}^{(\ell)}_{x,k} \in \mathbb{R}^{P_k \times D}, \qquad \mathbf{H}^{(\ell)}_f \in \mathbb{R}^{P_f \times D}, \tag{5}$$

where P_y, P_k, and P_f denote the corresponding numbers of tokens or patches. Their exact organization depends on the native architecture of the underlying TSFM, and all tensor shapes omit the batch dimension.

We use $\mathbf{H}^{(\ell)}$ to denote the complete backbone representation at layer ℓ, which contains these target, auxiliary, and forecast-generating states together with any other backbone-specific latent states. To provide a unified description across different TSFMs, we abstract the forecasting process as

$$\mathbf{H}^{(0)} = E_\theta(\mathbf{U}_{t-C+1:t}; h)\,, \tag{6}$$

$$\mathbf{H}^{(\ell)} = B^{(\ell)}_\theta\Big(\mathbf{H}^{(\ell-1)}\Big)\,, \qquad \ell = 1, \ldots, L, \tag{7}$$

$$\hat{\mathbf{y}}_0 = D_\theta\Big(\mathbf{H}^{(L)}\Big)\,. \tag{8}$$

Here, E_θ denotes the backbone-native input interface, $B^{(\ell)}_\theta$ the ℓ-th backbone layer, and D_θ the native prediction head. The unadapted backbone prediction is denoted by $\hat{\mathbf{y}}_0$.

4 Method

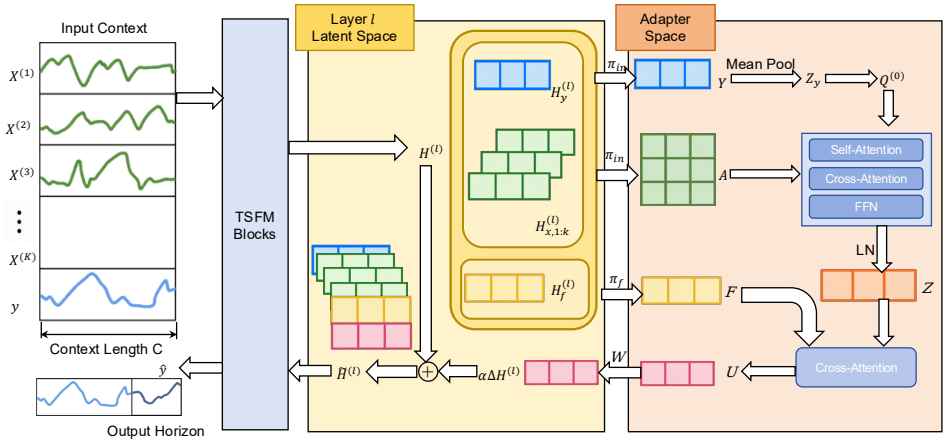


Figure 1: The framework of ReQuA. Target- and horizon-conditioned queries retrieve and compress task-relevant information from auxiliary variables into a fixed-size representation. The target forecast-generating latent states then attend to this representation to produce a calibrated residual correction, while the pretrained TSFM backbone and native prediction head remain frozen.

4.1 Overview

ReQuA is a lightweight latent adapter for frozen multivariate TSFMs. As illustrated in Figure 1, target- and horizon-conditioned queries retrieve task-relevant information from auxiliary-variable

states and compress it into a fixed-size representation Z. The backbone's forecast-generating states then attend to Z to produce a latent residual correction, whose magnitude is controlled by a globally calibrated coefficient α. The corrected states are processed by the remaining frozen backbone and its unchanged prediction head.

4.2 Target-Conditioned Query Aggregation

Shared input projection. Let d denote the adapter dimension used by ReQuA. Target and auxiliary context states share the same input projection from the backbone width D to the adapter width d:

$$\Pi_{\text{in}}(\mathbf{H}) = \text{LN}(\text{LN}(\mathbf{H})\mathbf{W}_a + \mathbf{b}_a), \qquad \mathbf{W}_a \in \mathbb{R}^{D\times d}, \quad \mathbf{b}_a \in \mathbb{R}^d. \tag{9}$$

For the k-th auxiliary variable, we augment the projected patch representations with sinusoidal positional encodings and a variable-identity representation:

$$\mathbf{A}_k = \Pi_{\text{in}}\left(\mathbf{H}_{x,k}^{(\ell)}\right) + \mathbf{P}_k + \mathbf{1}_{P_k}\mathbf{e}_k^\top \in \mathbb{R}^{P_k\times d}, \tag{10}$$

where $\mathbf{P}_k \in \mathbb{R}^{P_k\times d}$ denotes the sinusoidal positional encoding, and $\mathbf{e}_k \in \mathbb{R}^d$ is a variable-identity representation derived from window-level statistics of auxiliary variable k relative to the target and broadcast across its patches.

All valid auxiliary representations are concatenated along the token dimension:

$$\mathbf{A} = \text{Concat}(\mathbf{A}_1, \ldots, \mathbf{A}_K) \in \mathbb{R}^{P_A\times d}, \qquad P_A = \sum_{k=1}^{K} P_k. \tag{11}$$

Because all auxiliary variables share the same projection and attention operates only over currently available memory tokens, ReQuA does not require a fixed number of auxiliary variables.

Conditional queries. The target-history states use the same input projection and positional encoding:

$$\mathbf{Y} = \Pi_{\text{in}}\left(\mathbf{H}_y^{(\ell)}\right) + \mathbf{P}_y \in \mathbb{R}^{P_y\times d}. \tag{12}$$

They are then encoded into a target-conditioning vector:

$$\mathbf{z}_y = \text{MeanPool}(T_\phi(\mathbf{Y})) \in \mathbb{R}^d, \tag{13}$$

where T_ϕ denotes the lightweight target encoder. A learned embedding $\mathbf{e}_h \in \mathbb{R}^d$ encodes the forecast horizon h. Let

$$\mathbf{Q}_b \in \mathbb{R}^{Q\times d} \tag{14}$$

denote Q globally shared learnable base query tokens. A sample-specific query offset is generated from the target state and forecast horizon:

$$\Delta\mathbf{Q} = \text{reshape}(f_q([\mathbf{z}_y; \mathbf{e}_h])) \in \mathbb{R}^{Q\times d}, \qquad f_q : \mathbb{R}^{2d} \to \mathbb{R}^{Qd}. \tag{15}$$

The initial sample-conditioned queries are

$$\mathbf{Q}^{(0)} = \mathbf{Q}_b + \Delta\mathbf{Q} \in \mathbb{R}^{Q\times d}. \tag{16}$$

Thus, $\mathbf{Q}_b$ provides a shared retrieval template across samples, whereas $\Delta\mathbf{Q}$ dynamically adjusts the retrieval direction according to the current target state and forecast horizon.

Query blocks. Each query block contains query self-attention, query-to-auxiliary cross-attention, and a feed-forward network. The blocks use pre-normalization. For $j = 1, \ldots, L_q$, we compute:

$$\overline{\mathbf{Q}}^{(j)} = \mathbf{Q}^{(j-1)} + \text{MHA}\left(\text{LN}(\mathbf{Q}^{(j-1)}), \text{LN}(\mathbf{Q}^{(j-1)}), \text{LN}(\mathbf{Q}^{(j-1)})\right), \tag{17}$$

$$\widetilde{\mathbf{Q}}^{(j)} = \overline{\mathbf{Q}}^{(j)} + \text{MHA}\left(\text{LN}(\overline{\mathbf{Q}}^{(j)}), \mathbf{A}, \mathbf{A}\right), \tag{18}$$

$$\mathbf{Q}^{(j)} = \widetilde{\mathbf{Q}}^{(j)} + \text{FFN}\left(\text{LN}(\widetilde{\mathbf{Q}}^{(j)})\right). \tag{19}$$

After L_q query blocks, the final normalized query representation is

$$\mathbf{Z} = \text{LN}\left(\mathbf{Q}^{(L_q)}\right) \in \mathbb{R}^{Q\times d}. \tag{20}$$

Remark. Throughout this paper, MHA, FFN, and LN denote multi-head attention, feed-forward network, and layer normalization, respectively. For $\mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, the three arguments correspond to the query, key, and value inputs.

4.3 Latent Correction Injection

The query aggregation stage determines what auxiliary information is relevant to the current forecasting task. ReQuA then lets the backbone forecast-generating latent states interact with this fixed-size representation to obtain a state-specific latent correction.

The forecast-generating latent states are mapped into the adapter space using a separate future-state projection:

$$\mathbf{F}_f = \Pi_f\left(\mathbf{H}_f^{(\ell)}\right), \qquad \Pi_f(\mathbf{H}) = \mathrm{LN}(\mathbf{H})\mathbf{W}_f + \mathbf{b}_f, \tag{21}$$

The projected forecast-generating latent states then cross-attend to $\mathbf{Z}$:

$$\mathbf{V} = \mathrm{MHA}(\mathbf{F}_f, \mathbf{Z}, \mathbf{Z}) \in \mathbb{R}^{P_f \times d}. \tag{22}$$

The attended representation is projected back to the original backbone hidden space:

$$\Delta\mathbf{H}_Q = \mathbf{V}\mathbf{W}_o + \mathbf{b}_o, \qquad \mathbf{W}_o \in \mathbb{R}^{d\times D}, \quad \mathbf{b}_o \in \mathbb{R}^D, \quad \Delta\mathbf{H}_Q \in \mathbb{R}^{P_f \times D}. \tag{23}$$

The final latent injection is

$$\widetilde{\mathbf{H}}_f^{(\ell)} = \mathbf{H}_f^{(\ell)} + \alpha\Delta\mathbf{H}_Q, \tag{24}$$

4.4 Residual-Grounded Query Learning

We introduce a residual-grounding objective in Stage 1 to strengthen the training signal by directly supervising the query representation with the frozen backbone's forecast residual.

We first disable the ReQuA latent correction and obtain the corresponding prediction from the same frozen backbone and native decoding pathway used by the adapted model:

$$\hat{\mathbf{y}}_0 = F_\theta(\mathbf{U}_{t-C+1:t}; h) \in \mathbb{R}^h, \tag{25}$$

$$\mathbf{r}_0 = \mathbf{y}_{t+1:t+h} - \hat{\mathbf{y}}_0 \in \mathbb{R}^h. \tag{26}$$

Here, $\mathbf{r}_0$ denotes the forecast residual left by the frozen backbone. The backbone-specific point-readout conventions used during training and evaluation are provided in Appendix B.2.

The query representation is mean-pooled and passed through a lightweight residual supervision head $G_\psi : \mathbb{R}^d \to \mathbb{R}^h$:

$$\bar{\mathbf{z}} = \mathrm{MeanPool}(\mathbf{Z}) \in \mathbb{R}^d, \tag{27}$$

$$\hat{\mathbf{r}} = G_\psi(\bar{\mathbf{z}}) \in \mathbb{R}^h. \tag{28}$$

The corresponding grounding loss is:

$$\mathcal{L}_G = \frac{1}{h}\left\|\hat{\mathbf{r}} - \mathbf{r}_0\right\|_2^2. \tag{29}$$

The residual head G_ψ provides a short supervisory path to the Query Aggregator, encouraging $\mathbf{Z}$ to encode auxiliary information related to the remaining forecast error of the frozen backbone.

During Stage 1, the correction scale is fixed to $\alpha = 1$. The forecasting and overall objectives are

$$\mathcal{L}_{\mathrm{pred}} = \frac{1}{h}\left\|\hat{\mathbf{y}} - \mathbf{y}_{t+1:t+h}\right\|_2^2, \tag{30}$$

$$\mathcal{L}_{\mathrm{stage1}} = \mathcal{L}_{\mathrm{pred}} + \lambda_G\mathcal{L}_G. \tag{31}$$

Here, λ_G controls the weight of the residual-grounding objective. The Query Aggregator, latent correction module, and G_ψ are optimized while the TSFM backbone remains frozen during Stage 1.

4.5 Global Correction Calibration

The correction learned during Stage 1 may not have the same optimal magnitude across datasets. Moreover, auxiliary variables may contain components that do not explain the remaining prediction error of the frozen backbone, in which case applying the full correction may introduce unnecessary noise. ReQuA therefore calibrates a single global scaling coefficient in a second stage:

$$\alpha = \sigma(a), \qquad 0 < \alpha < 1, \tag{32}$$

where $a \in \mathbb{R}$ is the only trainable parameter in Stage 2 and $\sigma(\cdot)$ denotes the sigmoid function.

During Stage 2, the TSFM backbone, the Query Aggregator, and latent correction module are frozen, and only a is optimized:

$$\widetilde{\mathbf{H}}_f^{(\ell)} = \mathbf{H}_f^{(\ell)} + \sigma(a)\Delta\mathbf{H}_Q. \tag{33}$$

The calibration objective remains the forecasting error:

$$\mathcal{L}_{\text{stage2}} = \frac{1}{h} \left\| \hat{\mathbf{y}}_\alpha - \mathbf{y}_{t+1:t+h} \right\|_2^2. \tag{34}$$

The coefficient α is a dataset-level global scalar. Once calibrated, the same α is applied to all test samples under the corresponding experimental setting.

5 Experiments

5.1 Experiment Details

Datasets. We evaluate ReQuA on eight multivariate forecasting datasets: Electricity (Trindade, 2015), PEMS04 (Guo et al., 2019), ETTh1 and ETTm1 (Zhou et al., 2021), Weather (Wu et al., 2021), AQShunyi (Zhang et al., 2017), ZafNoo (Poyatos et al., 2021), and ILI (Centers for Disease Control and Prevention, 2026).

For datasets with at most 21 variables, each variable is evaluated in turn as the target, while the remaining variables are used as auxiliary inputs. For the high-dimensional Electricity and PEMS04 datasets, we select 15 approximately uniformly distributed target variables. Full dataset settings are provided in Appendix A.2.

Backbones and evaluation. We use frozen multivariate Chronos-2 (Ansari et al., 2025) as the primary backbone and its native forecast as the matched reference. The frozen and adapted models use identical targets, context windows, covariates, and forecasting masks; they differ only in whether ReQuA's latent correction is injected. We additionally evaluate ReQuA on TimesFM-3 (Jain & Sen, 2026) and Moirai-1.1 (Woo et al., 2024) under the same protocol, with full results reported in Appendix C.

We report MAE and MSE after normalization using training-set statistics, averaged over all test windows and evaluated target variables.

5.2 Main Results on Chronos-2

Table 1 reports the main results on the Chronos backbones. We include frozen Chronos-1 as an additional pretrained baseline (Ansari et al., 2024), while frozen Chronos-2 serves as the matched reference for ReQuA and for computing all relative improvements. We report both the Stage 1 model without global calibration (**ReQuA w/o** α) and the complete method (**ReQuA**), where the learned correction is modulated by the calibrated coefficient α. All results are averaged over three random seeds.

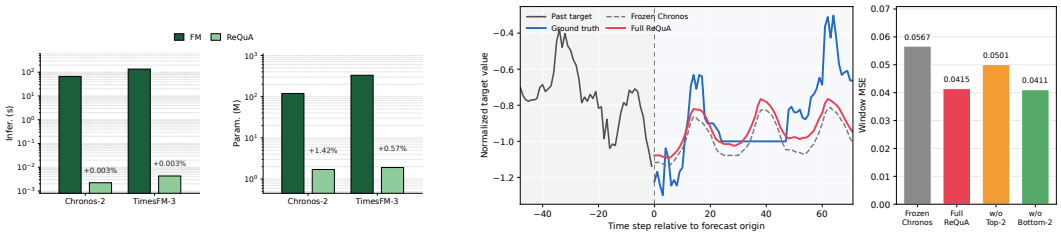
Overall performance. ReQuA w/o α already provides strong gains after Stage 1, reducing MSE on all eight datasets with an average improvement of 19.82% over frozen Chronos-2. This shows that the target-conditioned latent correction alone can recover substantial predictive information beyond the native multivariate backbone.

Table 1: Main results on Chronos backbones. ΔMAE and ΔMSE denote relative improvements over the frozen Chronos-2 baseline, and higher relative improvements are better.

	Chronos-1		Chronos-2		ReQuA w/o α				ReQuA				
Dataset	MAE ↓	MSE ↓	MAE ↓	MSE ↓	MAE ↓	MSE ↓	ΔMAE ↑	ΔMSE ↑	α	MAE ↓	MSE ↓	ΔMAE ↑	ΔMSE ↑
ETTh1	0.4048	0.5037	0.3849	0.3998	0.3858	0.3703	−0.22%	+7.38%	0.6417	0.3817	0.3733	+0.85%	+6.63%
ETTm1	0.4000	0.4688	0.3156	0.2959	0.3133	0.2665	+0.74%	+9.96%	0.8560	0.3112	0.2667	+1.40%	+9.88%
Weather	0.2251	0.2054	0.1814	0.1565	0.1826	0.1390	−0.63%	+11.15%	0.7495	0.1781	0.1391	+1.85%	+11.13%
AQShunyi	0.4997	0.7292	0.4797	0.6930	0.4632	0.6050	+3.45%	+12.70%	0.8456	0.4601	0.6055	+4.09%	+12.63%
ZafNoo	0.3834	0.4904	0.3838	0.4833	0.3708	0.4103	+3.38%	+15.11%	0.8834	0.3671	0.4106	+4.37%	+15.05%
Electricity	0.2463	0.2004	0.2134	0.1380	0.2120	0.1304	+0.66%	+5.48%	0.9491	0.2118	0.1305	+0.78%	+5.42%
PEMS04	0.7780	1.1272	0.3334	0.2613	0.2759	0.1761	+17.25%	+32.59%	0.8492	0.2769	0.1796	+16.95%	+31.29%
ILI	1.3988	4.8393	1.5456	6.4548	0.9335	2.3132	+39.60%	+64.16%	0.9521	0.9393	2.3237	+39.23%	+64.00%
Average	–	–	–	–	–	–	**+8.03%**	**+19.82%**	–	–	–	**+8.69%**	**+19.50%**

Effect of global calibration. Stage 2 further improves the robustness of the correction. Although calibration slightly reduces the average MSE gain from 19.82% to 19.50%, it eliminates the performance degradation observed on ETTh1 and Weather, turning −0.22% and −0.63% into +0.85% and +1.85%, respectively. Consequently, the final ReQuA improves both MAE and MSE on all eight datasets, while increasing the average MAE gain from 8.03% to 8.69%. This supports the use of α as a conservative calibration mechanism that suppresses over-correction.

Comparison with the univariate TSFM baseline. Chronos-2 outperforms Chronos-1 on most datasets, making it a stronger pretrained reference for ReQuA. ReQuA still improves over Chronos-2 across the benchmark. ILI is the main exception to the Chronos-1/Chronos-2 trend: Chronos-2 performs worse than Chronos-1, while ReQuA improves substantially over both. This case also shows that ReQuA can recover useful cross-variable information when the native multivariate backbone does not use it effectively.



(a) Efficiency **(b) Query-only intervention**

Figure 2: Analysis of ReQuA. **(a)** Inference-time and trainable-parameter overhead on Chronos-2 and TimesFM-3. **(b)** Query-only counterfactual intervention on ETTh1. The frozen Chronos-2 receives identical multivariate inputs in all conditions; only auxiliary variables available to the ReQuA query branch are modified.

Efficiency. Figure 2(a) shows that ReQuA adds only 1.42% and 0.57% trainable parameters to Chronos-2 and TimesFM-3, with inference overhead of about 0.003% in both cases. Log-scale axes are used to make these small overheads visible.

Query-only intervention. Figure 2(b) evaluates the functional role of attended covariates while keeping the Chronos-2 input unchanged. ReQuA lowers MSE from 0.0567 to 0.0415. Removing the two highest-attended variables (MUFL and HUFL) raises it to 0.0501, losing about 57% of the gain, whereas removing the two lowest-attended variables leaves MSE nearly unchanged at 0.0411. High query attention thus aligns with covariates that materially contribute to the correction.

Table 2: Negative adaptation counts over ten datasets relative to frozen Chronos-2.

Method	Auxiliary Retrieval	Target-only Correction	Backbone Unchanged	ΔMAE < 0	ΔMSE < 0
Residual MLP	×	✓	✓	4	2
Tiny Transformer	×	✓	✓	3	2
Output MLP	×	×	✓	8	3
LoRA	×	×	×	2	2
ReQuA	✓	✓	✓	**0**	**0**

Adaptation and module ablation. We evaluate the eight main benchmarks together with ETTh2 and ETTm2 (Zhou et al., 2021). We compare ReQuA with two complementary classes of alternatives: parameter-matched, auxiliary-free residual adapters, which test whether generic correction capacity, future-state interaction, or output-space correction is sufficient, and LoRA, which instead adapts the backbone's shared transformations with a comparable parameter budget(1.7M).

Both classes can improve individual datasets but exhibit negative adaptation on some tasks. The residual baselines show that additional latent capacity, self-attention, or output correction alone does not provide consistent gains, while LoRA shows that modifying shared backbone transformations is likewise not uniformly reliable. In contrast, ReQuA explicitly retrieves complementary auxiliary evidence conditioned on the prediction target, injects the correction only into target forecast states, and preserves the pretrained cross-variable interactions. Together with residual grounding and calibration, this design yields more consistent adaptation across datasets. Detailed configurations and results are provided in Appendix E.3.

5.3 Cross-Backbone Generalization

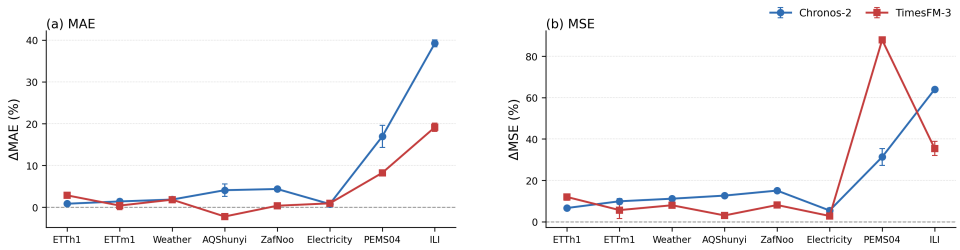


Figure 3: Relative error reduction of ReQuA over different backbones. (a) MAE reduction and (b) MSE reduction. Error bars denote standard deviations across three seeds.

Figure 3 compares ReQuA on Chronos-2 and TimesFM-3. Across the eight datasets, ReQuA improves MSE on all 16 backbone–dataset combinations and MAE on 15 of 16. The average MSE reduction is 19.50% on Chronos-2 and 20.33% on TimesFM-3. Overall, ReQuA remains effective on both native multivariate backbones despite their different architectures.

6 Conclusion

We present ReQuA, a parameter-efficient adaptation framework for frozen multivariate time-series foundation models. ReQuA uses target-conditioned queries to retrieve complementary auxiliary information while explicitly distinguishing prediction targets from supporting evidence. The retrieved information is residual-grounded and injected as a calibrated correction only into the target's forecast-generating latent states. Across eight datasets and multiple pretrained backbones, ReQuA provides broad forecasting improvements with modest parameter and computational overhead. These results suggest that multivariate TSFMs can be effectively adapted by selectively recovering auxiliary information not fully exploited by the backbone, without rewriting their pretrained cross-variable interactions.

Artificial Intelligence Use Statement

Our work used generative artificial intelligence tools to assist in manuscript editing, converting handwritten formulas into LaTeX, compiling references, debugging code, and creating some data graphs. All algorithm designs, experimental results, analyses, and inferences in this paper have been reviewed by the authors. The authors bear full responsibility for the final content of this research.

Ethics Statement

This work uses publicly available benchmark datasets and does not involve human-subject experiments or personally identifiable information. We are not aware of specific ethical risks beyond those associated with general-purpose forecasting models.

Reproducibility Statement

We provide the full training procedure, dataset splits and forecasting protocols, optimization settings, backbone configurations, and complete multi-seed results in the appendix. The main architectural components and training objectives of ReQuA are specified in Section 4.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736, 2022.

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.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229, 2020.

Centers for Disease Control and Prevention. U.S. Outpatient Influenza-like Illness Surveillance Network (ILINet). FluView, 2026. URL `https://www.cdc.gov/fluview/`. Accessed 2026.

Afrin Dange and Sunita Sarawagi. TFMAdapter: Lightweight instance-level adaptation of foundation models for forecasting with covariates. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, 2025. doi: 10.1145/3746252.3761272.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. Long-term forecasting with TiDE: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167, 2024.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16115–16152, 2024.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 922–929, 2019.

Divij Gupta, Anubhav Bhatti, Suraj Parmar, Chen Dan, Yuwei Liu, Bingjie Shen, and San Lee. Low-rank adaptation of time series foundational models for out-of-domain modality forecasting. In *Proceedings of the 26th International Conference on Multimodal Interaction*, 2024. doi: 10.1145/3678957.3685724.

Lu Han, Yu Liu, Lan Li, Qiwen Deng, Jian Jiang, Yinbo Sun, Zhe Yu, Binfeng Wang, Xingyu Lu, Lintao Ma, Han-Jia Ye, and De-Chuan Zhan. UniCA: Unified covariate adaptation for time series foundation model. In *International Conference on Learning Representations*, 2026.

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*, 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*, 2022.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4651–4664, 2021.

Ayush Jain and Rajat Sen. TimesFM-3: A Zero-Shot Foundation Model for Multivariate Forecasting. Google Research Blog, August 2026. URL `https://research.google/blog/timesfm-3-a-zero-shot-foundation-model-for-multivariate-forecasting/`.

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*, 2024.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753, 2019.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742, 2023.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 4582–4597, 2021.

Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764, 2021.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024a.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32369–32399, 2024b.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11525–11538, 2020.

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*, 2023.

Sehyuk Park, Soyeon Caren Han, and Eduard Hovy. UniCast: A unified framework for instance-conditioned multimodal time-series forecasting. *arXiv preprint arXiv:2508.11954*, 2025.

Rafael Poyatos, Víctor Granda, Víctor Flo, et al. Global transpiration data from sap flow measurements: The sapfluxnet database. *Earth System Science Data*, 13:2607–2649, 2021. doi: 10.5194/essd-13-2607-2021.

Guo Qin, Zhi Chen, Yong Liu, Zhiyuan Shi, Haixuan Liu, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. CoRA: Covariate-aware adaptation of time series foundation models. *arXiv preprint arXiv:2510.12681*, 2025.

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.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-MoE: Billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, 2025.

Artur Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. TimeXer: Empowering transformers for time series forecasting with exogenous variables. In *Advances in Neural Information Processing Systems*, 2024.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, 2021.

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.

Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in Beijing. *Proceedings of the Royal Society A*, 473(2205):20170457, 2017. doi: 10.1098/rspa.2017.0457.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

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*, 2023.

A Experimental Details

A.1 Training Procedure

ReQuA is optimized in two stages while all TSFM backbone parameters remain frozen. Throughout all experiments, ReQuA uses only historical (past-only) auxiliary variables; no future covariate values are provided to the adapter. The complete procedure is summarized in Table 3.

Table 3: Two-stage optimization procedure of ReQuA.

Stage 1: Query adaptation and residual grounding

Input: frozen backbone F_θ, training set $\mathcal{D}_{\mathrm{tr}}$, selection set $\mathcal{D}_{\mathrm{sel}}$.
1. Initialize the Query Aggregator, latent correction module, and grounding head G_ψ; set $\alpha = 1$.
2. Sample $(\mathbf{X}, \mathbf{y})$ from $\mathcal{D}_{\mathrm{tr}}$ and obtain the frozen prediction $\hat{\mathbf{y}}_0$ together with the required intermediate backbone states.
3. Construct the aggregated Query representation $\mathbf{Z} = \mathrm{QueryAggregator}(\mathbf{H}_y, \{\mathbf{H}_{x,k}\}, h)$.
4. Compute the latent correction $\Delta\mathbf{H}_Q = \mathrm{Injector}(\mathbf{H}_f, \mathbf{Z})$ and generate the adapted forecast from $\mathbf{H}_f + \Delta\mathbf{H}_Q$.
5. Optimize $\mathcal{L}_{\mathrm{stage1}} = \mathcal{L}_{\mathrm{pred}} + \lambda_G \mathcal{L}_G$. Here, $\mathcal{L}_{\mathrm{pred}} = \mathrm{MSE}(\hat{\mathbf{y}}, \mathbf{y})$ and $\mathcal{L}_G = \mathrm{MSE}(G_\psi(\mathrm{MeanPool}(\mathbf{Z})), \mathbf{y} - \hat{\mathbf{y}}_0)$.
6. Select the checkpoint with the lowest prediction MSE on $\mathcal{D}_{\mathrm{sel}}$ and discard G_ψ.

Stage 2: Global correction calibration

Input: selected Stage-1 checkpoint and calibration set $\mathcal{D}_{\mathrm{cal}}$.
1. Freeze the backbone and all ReQuA parameters.
2. Initialize $a = 0$ and set $\alpha = \sigma(a)$.
3. Optimize only a on $\mathcal{D}_{\mathrm{cal}}$ using prediction MSE, with the corrected latent state $\widetilde{\mathbf{H}}_f = \mathbf{H}_f + \alpha\Delta\mathbf{H}_Q$.
4. Fix the calibrated dataset-level α for all test samples.

A.2 Evaluation Protocol

All backbones use the same temporal split, forecasting windows, evaluation targets, and normalization for each dataset. The validation interval is partitioned 2:1 into non-overlapping subsets for Stage-1 checkpoint selection and Stage-2 calibration.

Table 4: Forecasting protocol used in the main experiments. C and H denote context length and forecasting horizon.

Dataset	Frequency	Split	C	H	Stride	Targets
ETTh1	1h	6:2:2	384	96	24	7
ETTm1	15min	6:2:2	672	96	24	7
Weather	10min	Autoformer	384	96	24	21
AQShunyi	1h	70/10/20	384	96	24	11
ZafNoo	30min	70/10/20	384	96	24	11
Electricity	1h	70/10/20	384	96	24	15/321
PEMS04	5min	60/20/20	384	96	24	15/307
ILI	weekly	70/10/20	48	24	1	7

For ETTh1, ETTm1, Weather, AQShunyi, ZafNoo, and ILI, all variables are evaluated as targets. For Electricity and PEMS04, 15 targets are selected from the beginning, middle, and end of the panel, with five targets from each region. The complete panel remains available as auxiliary input.

All variables are standardized using statistics estimated only from the training partition. MAE and MSE are reported in the train-normalized space unless otherwise specified. Every valid test origin is evaluated and all targets receive equal weight.

A.3 Optimization Settings

Table 5: ReQuA architecture and optimization settings used in the main experiments.

Setting	Value
Query tokens Q	15
Adapter dimension d	128
Query blocks L_q	2
Attention heads	4
Grounding weight λ_G	1.0
Stage-1 optimizer	AdamW
Stage-1 learning rate	1×10^{-4}
Weight decay	1×10^{-4}
Learning-rate schedule	Cosine, no warmup
Minimum learning rate	$0.1\times$ initial learning rate
Gradient clipping	Global norm 1.0
Mixed precision	Enabled
Training-origin sampling	5%
Checkpoint evaluation	Every 200 optimizer steps
Early stopping	4 non-improving evaluations
Stage-1 scaling	$\alpha = 1$
Stage-2 trainable parameter	Global scalar a only
Stage-2 initialization	$a = 0,\ \alpha = 0.5$
Stage-2 learning rate	0.05
Stage-2 steps	1,000
Random seeds	$\{42, 1731, 2024\}$
Reporting	Mean $\pm$ standard deviation

The frozen baseline and ReQuA use identical inputs, targets, forecast origins, and normalization within each run. Relative improvements are computed against the corresponding frozen backbone, with positive values indicating lower forecasting error.

B Datasets and Backbone Models

B.1 Dataset Summary

Table 6: Summary of the datasets used in our experiments.

Dataset	**Description**
ETTh1 (Zhou et al., 2021)	Hourly electricity-transformer measurements with seven variables, including load and oil-temperature related signals.
ETTm1 (Zhou et al., 2021)	Fifteen-minute electricity-transformer measurements from the ETT benchmark, with the same seven-variable structure as ETTh1.
Weather (Wu et al., 2021)	A 10-minute meteorological dataset containing 21 weather variables, including temperature, humidity, pressure, and wind-related measurements.
AQShunyi (Zhang et al., 2017)	Hourly air-quality observations from the Shunyi monitoring station in Beijing, including pollutant concentrations and meteorological variables.
ZafNoo (Poyatos et al., 2021)	A 30-minute ecohydrological dataset from SAPFLUXNET containing sap-flow measurements together with associated environmental variables.
Electricity (Trindade, 2015)	Hourly electricity-consumption measurements from 321 clients in the ElectricityLoadDiagrams20112014 collection.
PEMS04 (Guo et al., 2019)	Five-minute traffic-flow measurements from 307 detectors in the California Performance Measurement System (PeMS).
ILI (Centers for Disease Control and Prevention, 2026)	Weekly U.S. influenza-like-illness surveillance statistics, including ILI rates, age-group visit counts, and provider-related variables.

Table 7: Pretrained time-series foundation models considered in this work.

Model	Description
Chronos-2 (Ansari et al., 2025)	A pretrained TSFM with native support for univariate, multivariate, and covariate-informed forecasting. Chronos-2 uses group attention to exchange information across related series within a shared forecasting backbone.
TimesFM-3 (Jain & Sen, 2026)	A decoder-based pretrained forecasting model with native multivariate support. TimesFM-3 operates on patch-based representations and supports multiple targets together with historical and known-future covariates.
Moirai-1.1 (Woo et al., 2024)	A universal probabilistic forecasting model based on masked encoder pretraining and an any-variate architecture. It supports variable-dimensional multivariate inputs and multiple patch sizes; we use the Moirai-1.1 checkpoint in our experiments.

B.2 Backbone Models

Backbone-specific point readout. ReQuA does not introduce an additional prediction head. For each backbone, the frozen prediction $\hat{\mathbf{y}}_0$ used in residual grounding is obtained from the same native decoding pathway as the adapted prediction, with the latent correction disabled. Because the three backbones expose different native predictive parameterizations, the corresponding point-readout rule is backbone-specific, as summarized in Table 8.

Let $\mathcal{P}_b(\cdot)$ denote the point-readout operator associated with backbone b. The frozen and adapted point forecasts can therefore be written as

$$\hat{\mathbf{y}}_0 = \mathcal{P}_b\Big(D_\theta(\mathbf{H}_0^{(L)})\Big), \qquad \hat{\mathbf{y}} = \mathcal{P}_b\Big(D_\theta(\mathbf{H}_\alpha^{(L)})\Big), \tag{35}$$

where $\mathbf{H}_0^{(L)}$ and $\mathbf{H}_\alpha^{(L)}$ denote the final backbone states obtained without and with ReQuA correction, respectively. Accordingly, the residual target used for grounding is $\mathbf{r}_0 = \mathbf{y} - \hat{\mathbf{y}}_0$. Within each training or evaluation phase, the frozen backbone and ReQuA always use the same point-readout rule.

Table 8: Backbone-specific ReQuA integration and point-readout rules. The frozen baseline and ReQuA use the same point-readout rule within each phase.

Backbone	Checkpoint / width	Injection	Corrected future states	Point readout
Chronos-2	amazon/chronos-2 120M, $D = 768$	After encoder layer 6	Future output-patch states of the target variable	$q = 0.5$ quantile (median), after inverse instance normalization
TimesFM-3	google/timesfm-3.0-pytorch 0.3B, $D = 1280$	After mixing layer 10	Target states from the final context patch onward	$q = 0.5$, i.e., the fifth quantile in $\{0.1, \ldots, 0.9\}$
Moirai-1.1	Salesforce/moirai-1.1-R-base 91.4M, $D = 768$	After encoder layer 6	Prediction-masked horizon patches of the target variable (patch size 32)	Stage 1: differentiable mixture mean. Selection, calibration, and test: median of 100 sampled trajectories

The Query Aggregator reads historical target and auxiliary patch states from the same backbone layer immediately before correction injection. For TimesFM-3, correction starts from num_context_patches - 1; for Moirai-1.1, it starts from target_future_start. Only the target future states listed in Table 8 are modified; the latent states of the remaining variables are left unchanged.

C Full Experimental Results

Table 9 reports the complete three-seed results for Chronos-2, TimesFM-3, and Moirai-1.1. Stage 1 uses the unscaled correction ($\alpha = 1$), while ReQuA denotes the complete method after global calibration. All Δ values are relative error reductions over the corresponding frozen backbone; positive values indicate improvement. Results are reported as mean $\pm$ sample standard deviation over seeds $\{42, 1731, 2024\}$.

Table 9: Complete forecasting results on the eight benchmarks. MAE and MSE are measured in the train-normalized space. Δ denotes relative error reduction (%, higher is better).

	Frozen		ReQuA w/o α		ReQuA	
Dataset	MAE	MSE	ΔMAE	ΔMSE	ΔMAE	ΔMSE
Chronos-2						
ETTh1	0.3849	0.3998	-0.22 ± 0.28	$+7.38\pm1.01$	$\mathbf{+0.85\pm0.18}$	$\mathbf{+6.63\pm0.57}$
ETTm1	0.3156	0.2959	$+0.74\pm0.35$	$+9.96\pm1.19$	$\mathbf{+1.40\pm0.51}$	$\mathbf{+9.88\pm1.32}$
Weather	0.1814	0.1565	-0.63 ± 1.11	$+11.15\pm0.70$	$\mathbf{+1.85\pm0.58}$	$\mathbf{+11.13\pm0.64}$
AQShunyi	0.4797	0.6930	$+3.45\pm2.36$	$+12.70\pm0.54$	$\mathbf{+4.09\pm1.49}$	$\mathbf{+12.63\pm0.61}$
ZafNoo	0.3838	0.4833	$+3.38\pm0.64$	$+15.11\pm0.50$	$\mathbf{+4.37\pm0.47}$	$\mathbf{+15.05\pm0.53}$
Electricity	0.2134	0.1380	$+0.66\pm0.27$	$+5.48\pm1.25$	$\mathbf{+0.78\pm0.34}$	$\mathbf{+5.42\pm1.24}$
PEMS04	0.3334	0.2613	$+17.25\pm1.98$	$+32.59\pm2.23$	$\mathbf{+16.95\pm2.64}$	$\mathbf{+31.29\pm4.11}$
ILI	1.5456	6.4548	$+39.60\pm1.22$	$+64.16\pm1.23$	$\mathbf{+39.23\pm0.84}$	$\mathbf{+64.00\pm0.95}$
TimesFM-3						
ETTh1	0.3922	0.4358	$+2.25\pm0.35$	$+16.68\pm0.77$	$\mathbf{+2.85\pm0.27}$	$\mathbf{+11.96\pm1.55}$
ETTm1	0.3124	0.2929	-1.48 ± 0.71	$+10.11\pm0.46$	$\mathbf{+0.41\pm1.00}$	$\mathbf{+5.62\pm4.01}$
Weather	0.1883	0.1572	$+1.17\pm0.48$	$+8.13\pm0.98$	$\mathbf{+1.82\pm0.29}$	$\mathbf{+7.96\pm1.46}$
AQShunyi	0.4486	0.6152	-2.26 ± 0.46	$+3.03\pm0.15$	$\mathbf{-2.21\pm0.45}$	$\mathbf{+3.04\pm0.14}$
ZafNoo	0.3811	0.4675	$+0.09\pm0.21$	$+7.95\pm0.53$	$\mathbf{+0.37\pm0.23}$	$\mathbf{+8.07\pm0.41}$
Electricity	0.2201	0.1378	$+0.73\pm0.09$	$+3.33\pm0.24$	$\mathbf{+0.94\pm0.08}$	$\mathbf{+2.78\pm0.40}$
PEMS04	0.2712	1.2362	$+8.15\pm0.36$	$+87.90\pm0.06$	$\mathbf{+8.25\pm0.29}$	$\mathbf{+87.85\pm0.07}$
ILI	1.0314	2.7951	$+18.18\pm1.97$	$+33.13\pm6.41$	$\mathbf{+19.14\pm0.97}$	$\mathbf{+35.38\pm3.41}$
Moirai-1.1						
ETTh1	0.4069	0.4493	-0.88 ± 1.20	$+6.21\pm4.48$	$\mathbf{+0.03\pm0.75}$	$\mathbf{+6.33\pm2.29}$
ETTm1	0.5013	0.7424	$+29.31\pm0.94$	$+55.71\pm2.75$	$\mathbf{+29.36\pm0.91}$	$\mathbf{+55.76\pm2.56}$
Weather	0.2290	0.1956	$+17.11\pm0.97$	$+20.79\pm1.52$	$\mathbf{+13.75\pm4.14}$	$\mathbf{+16.35\pm5.50}$
AQShunyi	0.4194	0.5588	-0.50 ± 0.22	-1.59 ± 1.11	$\mathbf{-0.23\pm0.22}$	$\mathbf{-1.02\pm1.20}$
ZafNoo	0.3937	0.4513	$+2.31\pm2.14$	$+3.08\pm1.95$	$\mathbf{+2.57\pm2.06}$	$\mathbf{+3.55\pm1.91}$
Electricity	0.2848	0.2158	$+16.47\pm0.21$	$+30.09\pm0.27$	$\mathbf{+16.29\pm0.29}$	$\mathbf{+29.90\pm0.39}$
PEMS04	0.2575	0.1618	$+13.67\pm0.38$	$+28.11\pm0.71$	$\mathbf{+13.80\pm0.40}$	$\mathbf{+28.25\pm0.74}$
ILI	1.1673	3.0825	$+15.01\pm1.31$	$+16.31\pm1.96$	$\mathbf{+15.34\pm1.13}$	$\mathbf{+16.85\pm1.95}$

D Residual-Grounded Adaptation and Complexity

D.1 Why Residual Grounding Helps

Let $\mathbf{H}^{(\ell)}$ denote the intermediate representation of the frozen TSFM at the injection layer, and let $\mathbf{H}_f^{(\ell)}$ denote its forecast-generating latent states. The frozen prediction and its remaining forecasting residual are

$$\hat{\mathbf{y}}_0 = F_\theta^{(\ell+1:L)}\left(\mathbf{H}^{(\ell)}\right), \qquad \mathbf{r}_0 = \mathbf{y} - \hat{\mathbf{y}}_0. \tag{36}$$

ReQuA aggregates auxiliary information into

$$\mathbf{Z} = \mathcal{A}_\phi(\mathbf{A} \mid \mathbf{z}_y, \mathbf{e}_h) \in \mathbb{R}^{Q\times d}, \tag{37}$$

and produces a latent correction

$$\Delta\mathbf{H}_Q = \mathcal{I}_\omega\left(\mathbf{H}_f^{(\ell)}, \mathbf{Z}\right). \tag{38}$$

During Stage 1, the training-only grounding head directly predicts the residual left by the frozen backbone:

$$\mathcal{L}_G = \frac{1}{h} \left\| G_\psi(\bar{\mathbf{z}}) - \mathbf{r}_0 \right\|_2^2, \qquad \bar{\mathbf{z}} = \mathrm{MeanPool}(\mathbf{Z}). \tag{39}$$

Under squared error, the optimal residual predictor for a fixed representation $\bar{\mathbf{z}}$ is

$$G^*(\bar{\mathbf{z}}) = \mathbb{E}\left[\mathbf{r}_0 \mid \bar{\mathbf{z}}\right]. \tag{40}$$

The corresponding minimum prediction error satisfies

$$\min_G \mathbb{E}\left[\left\| \mathbf{r}_0 - G(\bar{\mathbf{z}}) \right\|_2^2 \right] = \mathbb{E}\left[\left\| \mathbf{r}_0 - \mathbb{E}[\mathbf{r}_0 \mid \bar{\mathbf{z}}] \right\|_2^2 \right]. \tag{41}$$

Thus, residual grounding encourages $\mathbf{Z}$ to retain the component of the frozen forecasting error that remains predictable from the retrieved covariate information. Its supervision is focused on the unexplained residual rather than the full target sequence, which reduces the incentive to relearn patterns already captured by the pretrained multivariate backbone.

The complete Stage-1 objective is

$$\mathcal{L}_{\mathrm{stage1}} = \underbrace{\frac{1}{h} \left\| \hat{\mathbf{y}} - \mathbf{y} \right\|_2^2}_{\mathcal{L}_{\mathrm{pred}}} + \lambda_G \underbrace{\frac{1}{h} \left\| G_\psi(\bar{\mathbf{z}}) - \mathbf{r}_0 \right\|_2^2}_{\mathcal{L}_G}, \tag{42}$$

with the TSFM frozen and $\alpha = 1$.

Calibration as residual shrinkage. Stage 2 calibrates the magnitude of the learned correction using a single global coefficient $\alpha = \sigma(a)$. A first-order expansion of the frozen downstream mapping gives

$$\hat{\mathbf{y}}_\alpha \approx \hat{\mathbf{y}}_0 + \alpha \mathbf{g}, \qquad \mathbf{g} = \mathbf{J}_\theta^{(\ell)} \Delta \mathbf{H}_Q, \tag{43}$$

where $\mathbf{J}_\theta^{(\ell)}$ is the local Jacobian with respect to the forecast-generating latent states. The resulting approximate risk is

$$\mathcal{R}(\alpha) = \mathbb{E}\left[\left\| \mathbf{r}_0 - \alpha \mathbf{g} \right\|_2^2 \right], \tag{44}$$

whose unconstrained optimum is

$$\alpha^* = \frac{\mathbb{E}[\langle \mathbf{r}_0, \mathbf{g} \rangle]}{\mathbb{E}[\|\mathbf{g}\|_2^2]}. \tag{45}$$

The preferred correction strength therefore depends on its alignment with the remaining forecast error and on the magnitude of the correction itself. The Stage-2 scalar provides a simple one-dimensional shrinkage mechanism for controlling over-correction.

D.2 Analysis of ReQuA as a lightweight adapter

Let P_A denote the total number of auxiliary-memory tokens, P_f the number of forecast-generating latent tokens, Q the number of Query tokens, d the ReQuA hidden dimension, D the backbone hidden dimension, and L_q the number of Query blocks. Ignoring constant factors and element-wise operations, the main additional costs introduced by ReQuA are summarized in Table 10.

Table 10: Dominant computational costs introduced by ReQuA.

Component	Dominant complexity	Main interaction
Query self-attention	$\mathcal{O}(BL_qQ^2d)$	Query–Query
Covariate retrieval	$\mathcal{O}(BL_qQP_Ad)$	Query–Auxiliary
Adapter FFNs / projections	$\mathcal{O}(BL_q(P_A+Q)d^2)$	Feature projection
Latent injection	$\mathcal{O}(BP_fQd)$	Forecast–Query
Backbone–adapter projection	$\mathcal{O}(BP_fDd)$	$d \leftrightarrow D$

The key term in auxiliary retrieval is QP_A, rather than P_A^2. ReQuA therefore avoids global self-attention over all auxiliary tokens. With a fixed and small number of Queries ($Q = 15$ in our experiments), the additional attention cost grows linearly with the size of the auxiliary memory. The second cross-attention stage scales with P_fQ, which is also small because both the number of forecast-generating tokens and the Query set are limited.

At inference time, the residual-grounding head is removed. ReQuA therefore adds only the Query Aggregator, latent injector, and one calibrated scalar to the frozen forecasting path. No additional TSFM forward pass is required, and the native prediction head remains unchanged.

Trainable parameter overhead. For the Chronos-2 configuration, the frozen backbone contains $119{,}477{,}664$ parameters. The complete Stage-1 ReQuA training branch contains $1{,}701{,}089$ trainable parameters, including the Query Aggregator, latent injector, and training-only grounding head. The resulting overhead is

$$\frac{1{,}701{,}089}{119{,}477{,}664} \times 100 = 1.424\% \approx \mathbf{1.42}\%. \tag{46}$$

During Stage 2, all of these parameters are frozen and only the scalar calibration logit a is optimized. ReQuA therefore obtains its adaptation capacity with a small trainable parameter budget and attention operations whose dominant cost is linear in the number of auxiliary-memory tokens.

E Robustness and Ablation Analysis

E.1 Hyperparameter Sensitivity

We vary one hyperparameter at a time around the default configuration $Q = 15$, $d = 128$, $\ell = 6$, and $\lambda_G = 1$, using Chronos-2 with seed 1731 and Stage-2 calibration.

Table 11: Hyperparameter sensitivity of ReQuA on ETTh1 and Weather. Δ denotes relative error reduction (%, higher is better). Only one hyperparameter is changed at a time.

Hyperparameter	Setting	ETTh1		Weather	
		ΔMAE	ΔMSE	ΔMAE	ΔMSE
Default	$Q = 15,\ d = 128,\ \ell = 6,\ \lambda_G = 1$	+0.89	+7.23	+1.98	+10.51
Query tokens Q	4	+0.17	+6.99	+0.83	+10.56
	8	+0.28	+7.65	+0.44	+10.32
	32	+0.91	+6.76	+2.70	+10.63
Adapter dimension d	64	+1.08	+6.50	-0.15	+10.61
	256	+1.26	+8.01	+1.85	+10.09
Injection layer ℓ	3	+0.27	+6.09	+1.47	+11.25
	9	+1.46	+6.57	+1.70	+11.23
Grounding weight λ_G	0	+0.17	+7.00	+1.18	+10.09
	0.5	+0.87	+7.65	+1.16	+11.15
	2.0	+0.51	+6.82	-1.84	+11.27

The structural hyperparameters show relatively mild sensitivity. Increasing Q or d improves individual metrics but does not yield a consistent advantage across datasets, suggesting that the Query Aggregator mainly requires sufficient capacity to form a compact retrieval bottleneck. The injection-layer results likewise remain stable across different depths, indicating that ReQuA does not depend on a narrowly chosen latent representation.

Residual grounding exhibits a clearer trade-off. Removing it weakens MAE improvements on both datasets, while an excessively large λ_G degrades Weather MAE despite retaining a strong MSE gain. This behavior is consistent with squared residual supervision placing increasing emphasis on large-error cases. The default $\lambda_G = 1$ provides a balanced interaction between residual grounding and the forecasting objective.

E.2 Forecast Horizon Ablation

We evaluate Chronos-2 across forecasting horizons $H \in \{96, 192, 336, 720\}$, where $H = 96$ corresponds to the main experimental setting.

Table 12: Forecast-horizon robustness of ReQuA with Chronos-2. Δ denotes relative error reduction over the frozen backbone (%, higher is better).

ETTh1				**Weather**			
Dataset	H	ΔMAE	ΔMSE	Dataset	H	ΔMAE	ΔMSE
ETTh1	96	+0.89	+7.23	Weather	96	+1.98	+10.51
ETTh1	192	+1.14	+3.74	Weather	192	+0.79	+12.10
ETTh1	336	+1.65	+4.53	Weather	336	+2.97	+12.73
ETTh1	720	+0.00	−1.04	Weather	720	+2.67	+11.03
ETTm1				**Electricity**			
Dataset	H	ΔMAE	ΔMSE	Dataset	H	ΔMAE	ΔMSE
ETTm1	96	+1.97	+9.35	Electricity	96	+1.13	+6.58
ETTm1	192	+2.35	+12.44	Electricity	192	+1.69	+4.71
ETTm1	336	+3.43	+12.96	Electricity	336	+0.52	+4.83
ETTm1	720	+5.09	+13.29	Electricity	720	+1.42	+3.94

The effect of forecast length is strongly dataset dependent. ETTm1 benefits increasingly from ReQuA as the horizon grows, while Weather maintains a similar MSE gain across all horizons. These patterns indicate that their auxiliary variables continue to carry information about the residual errors of the frozen backbone at longer prediction ranges.

Electricity retains smaller but persistent gains, whereas ETTh1 loses the benefit at $H = 720$. Under a fixed context window, the residual at very long horizons can become less predictable from historical covariates. This agrees with the residual-grounding interpretation in Section D.1: the usefulness of ReQuA depends primarily on how much predictable structure remains in $\mathbf{r}_0$, rather than on the forecast horizon itself.

E.3 Module Ablation

We further study whether the gains of ReQuA can be reproduced by either generic residual adapters or conventional parameter-efficient backbone adaptation. In addition to the eight main benchmarks, we include ETTh2 and ETTm2, resulting in ten datasets in total. All experiments use Chronos-2 with seed 42 and train-set normalization. For each dataset, performance is reported as the relative improvement over the frozen backbone:

$$\Delta(\%) = 100 \frac{\mathcal{E}_{\text{frozen}} - \mathcal{E}_{\text{adapted}}}{\mathcal{E}_{\text{frozen}}}, \tag{47}$$

where $\mathcal{E}$ denotes MAE or MSE, so that a positive value indicates an improvement over the frozen backbone.

Compared methods. We consider two complementary classes of alternatives. The first consists of auxiliary-free residual adapters with approximately the same trainable parameter budget as ReQuA. These variants leave the pretrained backbone transformations unchanged and differ in the form or location of the correction:

- **Residual MLP** applies a two-layer MLP independently to each future latent patch,

$$\Delta\mathbf{H} = W_2\,\mathrm{GELU}(W_1\,\mathrm{LN}(\mathbf{H}_f))\,, \qquad \mathbf{H}_f \leftarrow \mathbf{H}_f + \Delta\mathbf{H}. \tag{48}$$

  It uses neither auxiliary covariates nor interactions across future patches.

- **Tiny Transformer** first projects $\mathbf{H}_f$ from 768 to 256 dimensions, applies a two-layer Transformer encoder with four attention heads and an FFN dimension of 768, and projects

the result back to the backbone dimension. It therefore introduces self-attention among the target future patches while remaining auxiliary-free.

- **Output MLP** leaves the backbone latent states unchanged and instead corrects the decoded forecast:
$$\Delta\hat{\mathbf{y}} = W_2\,\mathrm{GELU}(W_1\,\mathrm{LN}([\mathbf{y}_{\mathrm{hist}};\hat{\mathbf{y}}_0]))\,, \qquad \hat{\mathbf{y}} = \hat{\mathbf{y}}_0 + \Delta\hat{\mathbf{y}}. \tag{49}$$
This baseline tests whether comparable capacity in the output space is sufficient.
- **LoRA** directly adapts the shared backbone transformations while keeping the original Chronos-2 weights frozen. Rank-7 low-rank updates are applied to the Q/K/V/O projections of temporal and group attention, the feed-forward layers, and the output head, yielding $1{,}701{,}168$ trainable parameters. Its effective batch size and checkpoint-selection protocol match those used for ReQuA.
- **ReQuA** retains the complete Query-based correction. The Query Aggregator produces auxiliary-conditioned tokens $\mathbf{Z}$, which are injected into $\mathbf{H}_f$ through cross-attention:
$$\mathbf{H}_f \leftarrow \mathbf{H}_f + \alpha\,\Delta\mathbf{H}_Q. \tag{50}$$
The Query representation is trained with residual grounding, while α is subsequently calibrated with the adapter frozen.

Table 13 summarizes the structural differences. Residual MLP and Output MLP are width-matched to approximately 1.70M parameters, Tiny Transformer contains 1.711M parameters, LoRA contains 1.701M trainable parameters, and ReQuA contains approximately 1.70M trainable parameters during Stage 1.

Table 13: Configurations used in the adaptation and module ablation. Auxiliary retrieval denotes explicit target-conditioned retrieval from auxiliary variables; backbone unchanged indicates that pretrained backbone transformations are not modified.

Method	Auxiliary Retrieval	Target-only Correction	Backbone Unchanged	Operation
Residual MLP	×	✓	✓	Per-patch MLP on $\mathbf{H}_f$
Tiny Transformer	×	✓	✓	Self-attention over target future patches
Output MLP	×	×	✓	MLP on $[\mathbf{y}_{\mathrm{hist}};\hat{\mathbf{y}}_0]$
LoRA	×	×	×	Rank-7 updates to shared backbone transformations
ReQuA	✓	✓	✓	Auxiliary-conditioned Query correction

Full results. Table 14 reports the complete results. The generic residual modules can provide large gains on individual datasets, but their behavior is less consistent across tasks. Residual MLP and Tiny Transformer both exhibit negative adaptation on several datasets, showing that additional latent capacity or future-state self-attention alone is insufficient. Output-space correction is even less stable, indicating that directly perturbing the decoded forecast is less reliable than operating within the backbone latent space.

LoRA provides a stronger comparison: with essentially the same trainable parameter budget as ReQuA, directly adapting the shared backbone transformations yields substantial gains on several datasets. However, it still produces negative MAE and MSE adaptation on the hourly ETT benchmarks. Thus, parameter-efficient backbone adaptation can be effective, but does not eliminate negative adaptation across tasks.

ReQuA is the only method with positive improvements in both MAE and MSE on all ten datasets. These results suggest that generic correction capacity or shared-backbone adaptation alone does not provide the same reliability. ReQuA instead combines target-conditioned auxiliary retrieval with target-local latent correction while preserving pretrained cross-variable interactions; residual grounding and calibration further constrain the correction toward complementary information and suppress harmful updates.

Table 14: Adaptation and module ablation on Chronos-2 over ten datasets. Values are relative improvements (%) over the frozen backbone; higher is better. Negative values indicate negative adaptation.

	ETTh1	ETTm1	Weather	AQShunyi	ZafNoo	Elec.	PEMS04	ILI	ETTh2	ETTm2	Avg.
Δ*MAE (%)*											
Residual MLP	-7.43	+1.45	-0.45	+0.98	+0.81	-0.26	+20.87	+45.08	-12.96	+2.64	+5.07
Tiny Transformer	-2.72	+0.68	+0.42	+0.67	-0.20	+1.61	+15.21	+42.34	-13.93	+2.45	+4.65
Output MLP	-9.60	-4.51	-20.74	-2.46	-8.06	-2.80	+6.03	+15.32	-10.83	-6.21	-4.39
LoRA	-5.42	+1.29	+0.95	+5.53	+1.95	+1.81	+23.53	+40.95	-4.98	+3.38	+6.90
ReQuA	**+0.65**	+1.25	**+1.22**	+2.41	**+4.24**	+0.73	+13.93	+39.40	**+1.45**	+3.10	+6.84
Δ*MSE (%)*											
Residual MLP	-10.32	+7.80	+11.48	+9.95	+12.93	+5.45	+37.41	+70.03	-25.00	+8.41	+12.81
Tiny Transformer	-6.80	+7.57	+11.94	+12.14	+12.22	+7.68	+26.54	+65.98	-30.17	+6.38	+11.35
Output MLP	-2.70	+3.04	-4.38	+7.30	+5.38	+2.74	+13.86	+32.78	-3.83	+2.44	+5.66
LoRA	-4.04	+7.70	+12.86	+14.86	+14.51	+8.96	+39.68	+59.75	-9.07	+8.11	+15.33
ReQuA	**+6.55**	**+8.91**	+11.79	+11.95	**+14.54**	+4.16	+26.74	+64.01	**+3.48**	+7.27	**+15.94**