StatTS-FM: Statistical Structure as an Inductive Bias for Time-Series Foundation Models

Anonymous authors
Paper under double-blind review

Abstract

Pointwise forecasting accuracy does not determine whether a prediction preserves the statistical structure of a time series. We study *statistical fidelity* of the predictive mean through its empirical value distribution, spectral composition, and lag dependence. We introduce StatTS-FM, a framework for supervised foundation-model adaptation that connects statistical representation, learning, and prediction. Trend, seasonal, remainder, and ACF/PACF-based tokens represent history; statistical losses constrain the forecast mean; and historical drift conditions predictive location and scale. We adapt a frozen TimesFM-3 backbone through low-rank adapters and these modules. We show how marginal supervision controls projected location and spread, and derive a statistical-constraint-dependent generalization bound under stationary β-mixing and uniform loss-moment conditions. We evaluate point forecasting across twelve benchmarks and examine the statistical design on ETTh1, Electricity, and Weather at two prediction horizons. Against a supervised raw-patch adaptation of the same backbone, StatTS-FM reduces macro-average MSE by 7.88% while lowering marginal, spectral, and lag-dependence errors. Without the statistical losses, the structured model improves MSE over raw-patch adaptation but worsens all three fidelity metrics; adding these losses improves both with the architecture fixed.

1 Introduction

Foundation models enable time-series forecasting through representations learned across temporal corpora. TimesFM, Timer, and Moirai follow this approach (Das et al., 2024; Liu et al., 2024b; Woo et al., 2024), alongside language-model adaptation through patches and reprogrammed tokens (Zhou et al., 2023; Sun et al., 2024; Jin et al., 2024). This makes the goal of adaptation consequential: *lower point error alone does not establish that a forecast captures the statistical structure of its target.* A trajectory's amplitude, periodicity, and temporal dependence carry information that an aggregate error score cannot resolve (Figure 1a).

We study *statistical fidelity* of the predictive mean through its empirical value distribution, spectral composition, and lag dependence relative to the target trajectory. Prior work examines MSE's limitations for temporal structure and marginal realism (Qiu et al., 2025; Green et al., 2026). Our same-backbone comparison exposes the limits of point-error-only evaluation: without fidelity losses, the statistically structured model achieves lower macro MSE than raw-patch adaptation, yet worse scores on all three statistical diagnostics (Figure 1c). This gap persists despite explicit statistical representations, motivating a design principle for adaptation: *statistical structure should guide both the representation of the history and the supervision of the final forecast.*

Non-stationarity compounds this challenge: changes in level, variance, and frequency composition shift the predictive center and uncertainty. Decomposition-based representations (Wu et al., 2021; Zhou et al., 2022; Wang et al., 2024; Cao et al., 2024), frequency-aware, structural, and distribution-alignment objectives (Wang et al., 2025a; Kudrat et al., 2025; Wang et al., 2026a), and adaptive correction (Wang et al., 2026b) address distinct parts of the problem. Statistical inputs alone do not constrain output statistics. Our design connects these parts by supervising the final, history-adjusted forecast.

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.

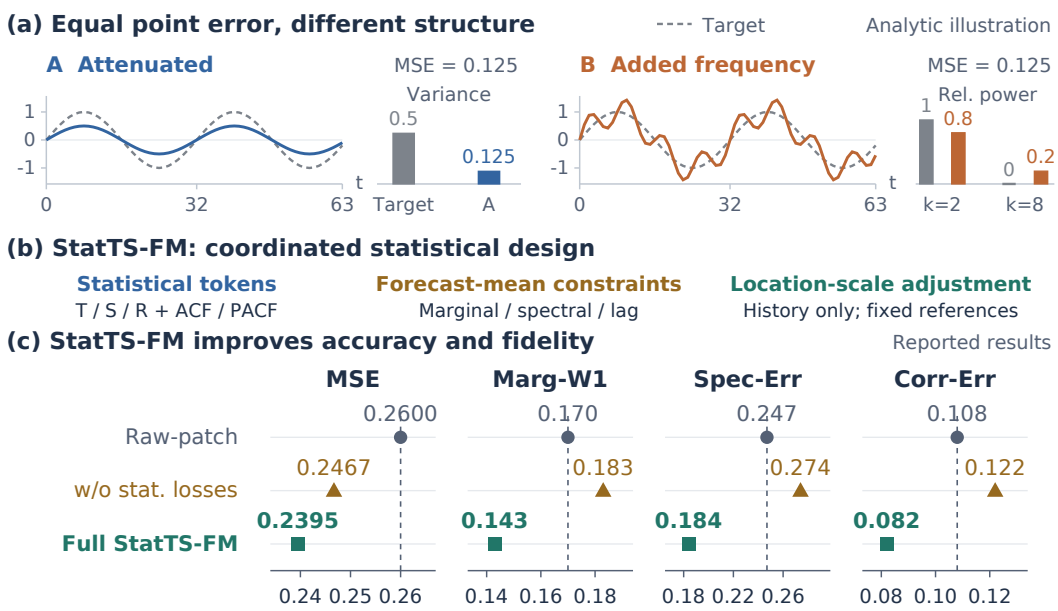


Figure 1: **Statistical structure beyond point error.** (a) Equal-MSE analytical examples, not model outputs, differ in variance and relative spectral energy; k indexes Fourier bins. (b) Statistical representations, forecast-mean constraints, and history-conditioned adjustment. (c) Reported point estimates: equal-weight macro averages over six settings—ETTh1, Electricity, and Weather at $H \in \{96, 336\}$—using training-standardized data. No-stat improves MSE over Raw-patch but worsens all three fidelity errors; Full improves all four metrics. Panels have independent axes, with lower values to the left and dashed Raw-patch references. Table 10 provides complete results.

We introduce **StatTS-FM**, which makes statistical structure an explicit inductive bias throughout supervised foundation-model adaptation (Figure 1b). Trend, seasonal, and remainder patches and ACF/PACF lag summaries represent history; mean, variance, and spectral drift condition predictive location and scale. Losses on the adjusted mean train both representation and adjustment toward forecast-level statistical fidelity. We instantiate the framework with TimesFM-3 and low-rank adaptation of its frozen backbone.

On ETTh1, Electricity, and Weather at two horizons, StatTS-FM reduces macro MSE by 7.88% over supervised raw-patch adaptation of TimesFM-3 and improves all three fidelity metrics (Figure 1c).

Our contributions are:

- **An empirical case for explicit statistical supervision.** We document accuracy–fidelity divergence in foundation-model adaptation and improve both through statistical supervision with the architecture fixed.
- **Statistical structure throughout foundation-model adaptation.** StatTS-FM connects component and dependence representations with drift-conditioned prediction and statistical constraints on the final mean.
- **A theoretical account of statistical regularization.** We characterize the forecast statistics controlled by marginal supervision and derive generalization bounds accounting for statistical constraints, window overlap, and temporal dependence.
- **Joint gains in accuracy and statistical fidelity.** Twelve benchmarks assess point accuracy; six-setting controls evaluate fidelity, predictive uncertainty, input corruption, and fewer adaptation windows.

2 Related Work

Autoformer combines decomposition with autocorrelation, FEDformer uses frequency representations, and TimeMixer mixes trend and seasonal components across temporal scales (Wu et al., 2021; Zhou et al., 2022; Wang et al., 2024). Forecasting objectives address related properties of the output. FreDF supervises forecasts against labels in the frequency domain, Patch-wise Structural Loss matches local mean, variance, and correlation, and DBLoss supervises decomposed trend and seasonal components (Wang et al., 2025a; Kudrat et al., 2025; Qiu et al., 2025). DistDF adds a joint-distribution Wasserstein term to MSE, using the means and covariances of concatenated history–forecast and history–target samples (Wang et al., 2026a). These objectives match different statistics within forecasts or across samples. Green et al. (2026) establish a trade-off between population MSE optimality and marginal realism under conditional uncertainty. StatTS-FM examines statistical supervision during foundation-model adaptation, evaluating value distributions, spectra, and finite-lag dependence within forecasts.

One Fits All adapts GPT-style backbones to patches, TEST uses textual prototypes, and Time-LLM reprograms numerical patches with statistical prompts (Zhou et al., 2023; Sun et al., 2024; Jin et al., 2024). TEMPO uses trend–seasonal–residual decomposition with component-specific prompting; Context-Alignment models structural and logical correspondence between time-series and language representations (Cao et al., 2024; Hu et al., 2025). FM-LLM uses frequency-aware adaptation and a time–frequency objective; CTRL uses LLM controllers for decomposed correction and label-free test-time adaptation (Gu et al., 2026; Kim et al., 2026). Tan et al. (2024) examine the contribution of language pretraining through systematic ablations.

TimesFM, Timer, Chronos, Moirai, and Sundial pretrain on temporal corpora (Das et al., 2024; Liu et al., 2024b; Ansari et al., 2024; Woo et al., 2024; Liu et al., 2025b). CoRA adapts them through time-varying cross-channel relations, their signs, and the channel subsets involved (Cheng et al., 2026). Other adaptation methods use in-context fine-tuning, lightweight online updating, or frequency-domain correction with arrived labels (Faw et al., 2025; Lee et al., 2025; Wang et al., 2026b). RevIN reverses input-window normalization at the output (Kim et al., 2022). StatTS-FM learns corrections to the Gaussian mean and scale from historical level, variance, and spectral deviations against training-time EMA references, with parameters and references fixed at inference.

3 Method

Problem and design. Given a history $\mathbf{X} \in \mathbb{R}^{L\times C}$, we predict $\mathbf{Y} \in \mathbb{R}^{H\times C}$. Pointwise accuracy alone does not specify the distribution of forecast values or their temporal organization. StatTS-FM (Figure 2) exposes statistical structure in the input, constrains the forecast mean, and conditions predictive location and scale on local drift. We use training-set standardization and an observation mask $\mathbf{M}$; $\widetilde{\mathbf{X}}$ denotes the input with missing entries set to zero. We omit the batch index throughout; $\epsilon > 0$ denotes a numerical safeguard.

3.1 Statistical Structure-Aware Representation

Separating temporal components. To expose variations at different scales, we form a learned mixture of centered moving averages and extract a frequency-gated seasonal component:

$$\begin{aligned} \mathbf{T} &= \sum_m \alpha_m \,\mathrm{MA}_{k_m}(\widetilde{\mathbf{X}}, \mathbf{M}), \quad \boldsymbol{\alpha} = \mathrm{softmax}(\boldsymbol{\zeta}), \\ \mathbf{S} &= \mathrm{irFFT}(\mathbf{G} \odot \mathbf{Z}), \qquad\qquad \mathbf{R} = \widetilde{\mathbf{X}} - \mathbf{T} - \mathbf{S}. \end{aligned} \tag{1}$$

Each moving average divides the sum of observed entries by their count, floored at one. Here rFFT is the orthonormal real Fourier transform, irFFT its inverse, and $\mathbf{Z} = \mathrm{rFFT}(\widetilde{\mathbf{X}} - \mathbf{T})$. Writing $A_{fc} = |Z_{fc}|$ and $\phi_{fc} = \arg Z_{fc}$, a shared multilayer perceptron (MLP) with sigmoid output computes G_{fc} for $f > 0$ from $\log(1 + A_{fc})$, $(1 + \cos(\phi_{fc} - \phi_{f-1,c}))/2$, and $A_{fc}/\max(\sum_{f'} A_{f'c}, \epsilon)$; the DC gate is fixed to zero. The remainder retains variation not assigned to trend or seasonality.

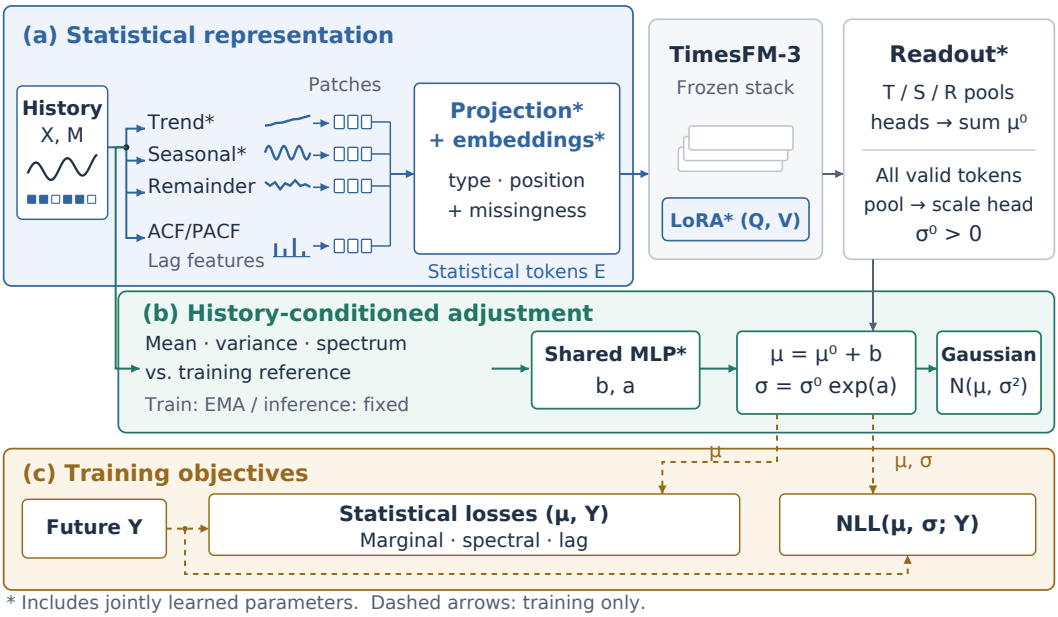


Figure 2: StatTS-FM overview. (a) Component patches and lag summaries are projected into statistical tokens. Asterisks mark modules containing jointly learned parameters. Component heads are summed for the base mean; pooling all valid token types supplies the scale head. (b) Historical drift adjusts location and scale. References use EMA during training; parameters and references are fixed at inference. (c) Separate training links route the final mean to statistical losses and both mean and scale to likelihood. Targets enter losses only. Auxiliary regularizers are defined below; curves are schematic.

Representing lag dependence. Components describe signal structure; lag summaries supply explicit dependence information. For the centered, zero-filled series $z_{ic} = \widetilde{x}_{ic} - L^{-1}\sum_j \widetilde{x}_{jc}$, define the autocorrelation function (ACF)

$$\rho_c(\ell) = \frac{\sum_{i=\ell+1}^{L} z_{ic} z_{i-\ell,c}}{\max(\sum_{i=1}^{L} z_{ic}^2, \epsilon)}. \tag{2}$$

We compute partial autocorrelation function (PACF) coefficients $\pi_c(\ell)$ by stabilized Levinson–Durbin recursion. Up to $D_L = \min(D_{\max}, \lfloor L/2 \rfloor, L-1)$, the lag set Λ combines powers of two with the top-k lags ranked separately by $C^{-1}\sum_c |\rho_c(\ell)|$ and $C^{-1}\sum_c |\pi_c(\ell)|$. Each window has one deduplicated set shared across variables. With pooling width w, $u_c(\ell)$ averages the $n_\ell = \min(w, L-\ell)$ entries ending at index $L-\ell$ of $\widetilde{\mathbf{X}}_{:,c}$, giving $\mathbf{q}_{c\ell} = [\rho_c(\ell), \pi_c(\ell), u_c(\ell)]^\top \in \mathbb{R}^3$.

Statistical tokens. For each variable, we patch each component $U \in \{T, S, R\}$ into length-P vectors $\mathbf{p}_{jc}^U$ with stride s. Separate affine maps $\mathcal{A}_U : \mathbb{R}^P \to \mathbb{R}^d$ and $\mathcal{A}_{\text{dep}} : \mathbb{R}^3 \to \mathbb{R}^d$ produce

$$\begin{aligned} \mathbf{e}_{jc}^U &= \mathcal{A}_U(\mathbf{p}_{jc}^U) + \boldsymbol{\tau}_U + \mathrm{PE}(j) + \mathbf{w}_m m_{jc}, \\ \mathbf{e}_{c\ell}^{\text{dep}} &= \mathcal{A}_{\text{dep}}(\mathbf{q}_{c\ell}) + \boldsymbol{\tau}_{\text{dep}} + \mathrm{PE}(\ell), \end{aligned} \tag{3}$$

where $\boldsymbol{\tau}$ denotes type embeddings, PE sinusoidal positions, and m_{jc} the patch's missing fraction. Concatenating T, S, R, and dependence tokens, followed by root mean square (RMS) normalization, gives $\mathbf{E} \in \mathbb{R}^{C\times(3N+|\Lambda|)\times d}$ for N patches. Fully unobserved patches and padded lag slots are masked.

3.2 Backbone Adaptation and Predictive Readout

The TimesFM-3 Transformer stack processes $\mathbf{E}$ in separate variable groups. We replace its native input and output modules with the statistical token projection and predictive readout defined here.

For variable c, let $\overline{\mathbf{h}}_c^U$ and $\overline{\mathbf{h}}_c$ average valid output tokens within component U and across all token types, respectively. RMS-normalized linear heads into $\mathbb{R}^H$ yield

$$\boldsymbol{\mu}^0_{:,c} = \sum_{U \in \{T,S,R\}} h_U(\overline{\mathbf{h}}_c^U), \qquad \boldsymbol{\sigma}^0_{:,c} = \mathrm{softplus}(h_\sigma(\overline{\mathbf{h}}_c)) + \sigma_{\min}. \tag{4}$$

Here $\sigma_{\min} > 0$ is a floor on the base scale. The component outputs are learned jointly through the total forecast. An explicit remainder head supplies its future contribution without requiring future observations. The original backbone weights are frozen; query and value projections receive low-rank adaptation (LoRA) updates $\Delta W = (\alpha_{\mathrm{L}}/r)BA$ (Hu et al., 2022), with $A \in \mathbb{R}^{r \times d_{\mathrm{in}}}$, $B \in \mathbb{R}^{d_{\mathrm{out}} \times r}$ and dropout on adapter inputs.

3.3 Drift-Conditioned Location and Scale

Changes in level, variance, or frequency composition can alter the predictive center and uncertainty. For each variable, compute observed-entry mean μ_c^x, variance v_c^x, and a spectral summary $\mathbf{p}_c$. The latter averages one-sided Fourier power into Q bins and divides by total pooled power, floored at ϵ. References $(\mu_c^{\mathrm{ref}}, v_c^{\mathrm{ref}}, \mathbf{p}_c^{\mathrm{ref}})$ are exponential moving averages (EMA) of training-window statistics. Define

$$\begin{aligned} \mathbf{d}_c &= \left[\frac{\mu_c^x - \mu_c^{\mathrm{ref}}}{\sqrt{v_c^{\mathrm{ref}}}}, \log \frac{v_c^x + \epsilon}{v_c^{\mathrm{ref}} + \epsilon}, D_{\mathrm{sp}}(\mathbf{p}_c, \mathbf{p}_c^{\mathrm{ref}}) \right]^\top \in \mathbb{R}^3, \\ D_{\mathrm{sp}}(\mathbf{p}, \mathbf{r}) &= \frac{1}{Q} \sum_{q=1}^{Q} \left| \sum_{j=1}^{q} (p_j - r_j) \right|. \end{aligned} \tag{5}$$

Variances are floored at ϵ; spectra use zero-filled inputs. A shared MLP $g_\psi : \mathbb{R}^3 \to \mathbb{R}^{2H}$ outputs location corrections $\mathbf{b}_{:,c}$ and log-scale corrections $\mathbf{a}_{:,c}$, with the latter clipped to $[-5, 5]$. The conditional model is

$$\begin{aligned} \boldsymbol{\mu} &= \boldsymbol{\mu}^0 + \mathbf{b}, \qquad \boldsymbol{\sigma} = \boldsymbol{\sigma}^0 \odot \exp(\mathbf{a}), \\ p(\mathbf{Y} \mid \mathbf{X}, \mathbf{M}) &= \prod_{h=1}^{H} \prod_{c=1}^{C} \mathcal{N}(Y_{hc}; \mu_{hc}, \sigma_{hc}^2). \end{aligned} \tag{6}$$

References are initialized from the first training batch; subsequent batches are scored before the EMA update. At inference, model parameters and references are fixed, and the learned adjustment requires only the observed history.

3.4 Learning Statistical Fidelity

The factorized likelihood fits conditional location and scale. To also constrain the shape and temporal structure of the forecast mean, set $\widehat{\mathbf{Y}} = \boldsymbol{\mu}$ and compare it with the future target. The squared-error-optimal forecast is the conditional mean (Gneiting, 2011). Averaging over possible futures can attenuate trajectory variation even when this point forecast is optimal. With finite second moments, the future's expected sum of squared first differences, conditional on the history, equals that of its conditional mean plus a nonnegative term from conditional uncertainty (Equation 19). Whenever this term is positive, the conditional mean is smoother in this energy sense than the future is on average given the history. Thus, point accuracy alone does not determine trajectory fidelity.

We apply statistical penalties to the final mean to make value distribution, frequency composition, and lag dependence explicit adaptation targets. This is a finite-data inductive bias whose accuracy–structure trade-off is evaluated empirically; it does not lower the population Bayes MSE minimum. The conditional-mean identity motivates separate structural evaluation, without directly establishing properties of the nonlinear spectral and lag losses. Probabilistic forecasts are assessed separately using the predicted location and scale. Appendix F gives the identities and proofs.

Let $s_k(\mathbf{U})$ sort the H projections $\mathbf{v}_k^\top \mathbf{U}_{h,:}$ for fixed random unit vectors $\mathbf{v}_k \in \mathbb{R}^C$, $k = 1, \dots, K$. Let $P_f(\mathbf{u}) = |\operatorname{rFFT}(\mathbf{u})_f|^2 / \max(\sum_{f'} |\operatorname{rFFT}(\mathbf{u})_{f'}|^2, \epsilon)$ and $F_H = \lfloor H/2 \rfloor + 1$. We use

$$\begin{aligned}
\mathcal{L}_{\text{marg}} &= \frac{1}{KH} \sum_k \|s_k(\widehat{\mathbf{Y}}) - s_k(\mathbf{Y})\|_2^2, \\
\mathcal{L}_{\text{freq}} &= \frac{1}{CF_H} \sum_{c,f} \left| \log(P_f(\widehat{\mathbf{Y}}_{:,c}) + \epsilon) - \log(P_f(\mathbf{Y}_{:,c}) + \epsilon) \right|, \\
\mathcal{L}_{\text{corr}} &= \frac{1}{C} \sum_c \sum_{\ell \in \Lambda_H} w_\ell \left(|\widehat{\rho}_c(\ell) - \rho_c^Y(\ell)| + \eta |\widehat{\pi}_c(\ell) - \pi_c^Y(\ell)| \right).
\end{aligned} \tag{7}$$

Hatted coefficients are computed from $\widehat{\mathbf{Y}}$; $\eta \geq 0$ weights the PACF term. The first is a sliced-Wasserstein loss (Bonneel et al., 2015); sorting discards temporal order. The other terms emphasize frequency composition and dependence at selected lags. Spectrum and autocorrelation encode related second-order structure; here, different centering, normalization, lag selection, and weighting impose different optimization biases. Individual-loss ablations assess their empirical contributions. Λ_H uses the same lag-selection rule on the target, with $D_H = \min(D_{\text{corr}}, \lfloor H/2 \rfloor, H-1)$ and $w_\ell = e^{-\ell/D_H} / \sum_{j \in \Lambda_H} e^{-j/D_H}$. Target-based statistics are used only for loss evaluation.

We discourage redundant components with $\mathcal{L}_{\text{sep}}$, the variable-averaged sum of absolute cosine similarities over the three unordered component pairs, using an ϵ-stabilized denominator. Reconstruction is exact by definition of $\mathbf{R}$. To favor small adjustments near the training reference, use $\mathcal{L}_{\text{cal}} = C^{-1} \sum_c e^{-\kappa \|\mathbf{d}_c\|_2} \sum_h (b_{hc}^2 + a_{hc}^2)$, with decay parameter $\kappa \geq 0$. Writing $\bar{v}_{hc} = \max(\sigma_{hc}^2, \epsilon)$, the training objective is

$$\begin{aligned}
\mathcal{L}_{\text{pred}} &= \frac{1}{2HC} \sum_{h,c} \left[\log(2\pi \bar{v}_{hc}) + \frac{(Y_{hc} - \mu_{hc})^2}{\bar{v}_{hc}} \right], \\
\mathcal{L} &= \mathcal{L}_{\text{pred}} + \lambda_{\text{sep}} \mathcal{L}_{\text{sep}} + \lambda_{\text{marg}} \mathcal{L}_{\text{marg}} + \lambda_{\text{freq}} \mathcal{L}_{\text{freq}} \\
&\quad + \lambda_{\text{corr}} \mathcal{L}_{\text{corr}} + \lambda_{\text{cal}} \mathcal{L}_{\text{cal}}.
\end{aligned} \tag{8}$$

Regularization weights are nonnegative. We average $\mathcal{L}$ over training windows and jointly learn the statistical encoder, projections, readout heads, drift network, and adapters. Algorithm 1 summarizes training and inference.

4 Theoretical Analysis

Control of forecast statistics. For a window's mean $m_k(\mathbf{U})$ and standard deviation $\tau_k(\mathbf{U})$ along fixed projection $\mathbf{v}_k$, Proposition 1 gives

$$\frac{1}{K} \sum_{k=1}^K \left[\big(m_k(\boldsymbol{\mu}) - m_k(\mathbf{Y})\big)^2 + \big(\tau_k(\boldsymbol{\mu}) - \tau_k(\mathbf{Y})\big)^2 \right] \leq \mathcal{L}_{\text{marg}}(\boldsymbol{\mu}, \mathbf{Y}). \tag{9}$$

This bound holds for each window without independence or stationarity assumptions. It controls projected forecast-mean location and spread across forecast time points, separately from predictive uncertainty and variability across training runs.

Generalization under dependent windows. Consider stationary β-mixing training windows (Z_t) with law $\mathbb{P}$ and a fixed class $\mathcal{F}$ of complete predictors, including their inference-time reference states. Assume uniformly bounded predictive loss $0 \leq \ell_f \leq M$ and statistical discrepancy $0 \leq d_f \leq B$; d_f combines the three statistical losses with nonnegative weights fixed for analysis. Write $R_{\mathbb{P}}(f)$ and $\widehat{R}_n(f)$ for population and empirical predictive risk, and $D_{\mathbb{P}}(f)$ and $\widehat{D}_n(f)$ for the corresponding statistical discrepancies.

For n windows, fix a block length $a > L + H$ before observing the sample and write $n = 2am + r$, where $m = \lfloor n/(2a) \rfloor \geq 1$ and $0 \leq r < 2a$. Fix a grid $0 \leq \varepsilon_1 < \cdots < \varepsilon_J = B$ before observing the sample. For $0 < \delta < 1$, let $\alpha = \delta - 2(m-1)\beta_Z(a) > 0$ and $t = \sqrt{\log(2(J+1)/\alpha)/(2m)}$. The estimation allowance is $\zeta = 2\mathcal{R}_{m,a}(\mathcal{A}) + Bt + Br/n$, where $\mathcal{A} = \{d_f : f \in \mathcal{F}\}$ and $\mathcal{R}_{m,a}$ is the

expected block Rademacher complexity defined in Appendix G. Define $\mathcal{F}_u = \{f \in \mathcal{F} : D_{\mathbb{P}}(f) \leq u\}$ and $\mathcal{G}_j = \{\ell_f : f \in \mathcal{F}_{\varepsilon_j+\zeta}\}$. Theorem 1 gives, with probability at least $1-\delta$, simultaneously for all j and predictors satisfying $\widehat{D}_n(f) \leq \varepsilon_j$,

$$R_{\mathbb{P}}(f) \leq \widehat{R}_n(f) + 2\mathcal{R}_{m,a}(\mathcal{G}_j) + M\sqrt{\frac{\log(2(J+1)/\alpha)}{2m}} + \frac{Mr}{n}. \tag{10}$$

The three terms beyond empirical risk form $\mathcal{E}_m(\mathcal{F}_{\varepsilon_j+\zeta})$: complexity of the statistically constrained loss class, concentration over m blocks, and the remainder from r windows. Larger blocks reduce dependence between separated blocks but leave fewer blocks for concentration. Window overlap enters through $\beta_Z(a) \leq \beta_x(a-L-H)$ for windows constructed from a stationary raw process (including masks when random), so the number of overlapping windows alone does not determine the bound. The simultaneous guarantee permits selection of the predictor and grid level after training; it does not require a global optimizer.

Tighter statistical constraints nest the predictor classes and cannot increase their complexity term. This identifies a mechanism for controlling estimation error, while retaining the discrepancy-estimation allowance ζ and the possible cost in empirical fit. It does not guarantee a strictly smaller realized generalization gap. Gaussian NLL is not generally bounded; Appendix G gives the extension under uniform loss-moment conditions, full proofs, and deployment bounds with separate assumptions. These are analytical conditions, not assumptions established by the benchmark results.

5 Experiments

5.1 Experimental Setup

Implementation. We adapt frozen TimesFM-3 with query/value LoRA (rank 8, scaling 16, dropout 0.1) and jointly train the statistical modules. AdamW learning rates are 10^{-4} for added modules and 10^{-5} for LoRA. Benchmarks allow 30 epochs with early-stopping patience five; reduced-window and profiling runs use 5,000-update budgets. Appendix E collects representation hyperparameters and training protocols.

Benchmarks. We use ETTh1/2, ETTm1/2, Electricity, Traffic, Weather, Exchange, ILI, Solar-Energy, and PEMS03/08. Splits are chronological: 12/4/4 months for ETT, 6:2:2 for PEMS, and 7:1:2 otherwise. Standardization is fitted on the training partition. Long-horizon comparisons use $H \in \{96, 192, 336, 720\}$, with $L = 96$ for dedicated forecasting architectures and $L = 512$ for LLM-based methods. ILI uses $L = 36$ and $H \in \{24, 36, 48, 60\}$; PEMS uses $L = 96$ and $H \in \{12, 24, 48, 96\}$. The four benchmark tables average over horizons, then give each dataset equal weight in macro averages; ILI is excluded from the LLM macro average. Different comparison settings are not pooled.

Controlled comparisons and metrics. The supplementary experiments use ETTh1, Electricity, and Weather, $L = 96$, and $H \in \{96, 336\}$. MSE and MAE use training-standardized values, averaged over windows, forecast steps, and variables; macro averages weight the six settings equally. Relative changes use unrounded macro errors. We report point estimates without cross-seed standard deviations or significance claims. Statistical fidelity is evaluated on the predictive mean using marginal Wasserstein distance (Marg-W1), spectral error (Spec-Err), and dependence error (Corr-Err). Gaussian CRPS, NLL, and 80% interval coverage and width assess the predictive distribution. Appendix B defines the metrics separately from training losses; Appendix E collects the comparison protocols.

5.2 Point Accuracy and Statistical Fidelity

Same-backbone comparison. Full and Raw-patch freeze the same pretrained backbone revision, use identical query/value LoRA configurations, and receive comparable update and hyperparameter-search budgets. Raw-patch uses ordinary patch inputs and a Gaussian readout without the statistical mechanisms. Full reduces macro MSE from 0.2600 to 0.2395 (7.88%) and MAE from 0.3032 to

0.2858, while improving all three fidelity metrics (Figure 1c; Table 10). Native TimesFM-3's official zero-shot path gives macro MSE/MAE of $0.2842/0.3263$ (Table 8). This comparison also includes the effect of supervised adaptation.

Why supervise trajectory statistics? The statistical losses and fidelity diagnostics target the same families of statistics, with different definitions (Appendix B). Lower diagnostic errors on held-out windows show better matching beyond training. The no-stat variant improves MSE over Raw-patch (0.2467 versus 0.2600), yet worsens all three fidelity metrics (Figure 1c), showing that statistical representations and drift adjustment alone do not ensure output fidelity. Full and no-stat share the same architecture; Full lowers MSE in all six settings (Table 6) and improves all three macro diagnostics. Each separately trained single-loss ablation worsens its targeted macro diagnostic and the other two, so the observed effects extend beyond the statistic directly penalized. Each also increases macro point error; joint removal is worse on all macro metrics (Table 10). Benefits are not uniform: without marginal loss, ETTh1 at $H = 96$ has slightly lower MSE/MAE than Full ($0.349/0.371$ versus $0.350/0.372$; Table 11). These joint gains concern finite-data adaptation, whereas Appendix F characterizes the population Bayes MSE minimum; they do not imply a lower Bayes risk.

Representation and adjustment. We also disable drift-conditioned location and scale adjustment; replace the three component streams and mean heads with one raw-input stream and head; or remove dependence tokens. Unlike Raw-patch, the no-decomposition variant retains the other statistical mechanisms. Removing dependence tokens retains correlation loss; removing drift adjustment retains the Gaussian readout. All removals increase macro MSE and MAE (Table 1), with the largest MSE increases for drift adjustment (5.57%) and decomposition (5.50%). These separately adapted variants support component contributions within the full system, but changes in readout structure or capacity are not isolated. Table 6 gives per-setting results.

Table 1: Component ablations, averaged equally over three datasets and two horizons. ΔMSE is the relative increase over Full.

Variant	MSE ↓	MAE ↓	ΔMSE
Full	**0.2395**	**0.2858**	—
Without statistical losses	0.2467	0.2932	2.99%
Without drift adjustment	0.2528	0.2967	5.57%
Without decomposition	0.2527	0.2968	5.50%
Without dependence tokens	0.2458	0.2913	2.64%

5.3 Benchmark Forecasting Results

Table 2 compares StatTS-FM with TimeMixer++ (Wang et al., 2025b), TimeMixer (Wang et al., 2024), iTransformer (Liu et al., 2024a), PatchTST (Nie et al., 2023), and TimesNet (Wu et al., 2023). StatTS-FM has the lowest reported horizon-averaged MSE and MAE on all nine datasets, with macro MSE/MAE of $0.279/0.294$ versus $0.307/0.324$ for TimeMixer++.

Table 2: Long-horizon forecasting results averaged over prediction horizons 96, 192, 336, and 720 with input length 96. Each entry is MSE / MAE; lower is better. The TimesNet Electricity entry follows Wu et al. (2023, Table 2).

Dataset	StatTS-FM	TimeMixer++	TimeMixer	iTransformer	PatchTST	TimesNet
Electricity	**0.147 / 0.226**	0.165 / 0.253	0.182 / 0.272	0.178 / 0.270	0.216 / 0.318	0.192 / 0.295
Traffic	**0.378 / 0.239**	0.416 / 0.264	0.484 / 0.297	0.428 / 0.282	0.529 / 0.341	0.620 / 0.336
Weather	**0.206 / 0.240**	0.226 / 0.262	0.240 / 0.271	0.258 / 0.278	0.265 / 0.285	0.259 / 0.287
Exchange	**0.320 / 0.354**	0.357 / 0.391	0.391 / 0.453	0.360 / 0.403	0.367 / 0.404	0.416 / 0.443
Solar-Energy	**0.187 / 0.219**	0.203 / 0.238	0.216 / 0.280	0.233 / 0.262	0.287 / 0.333	0.403 / 0.374
ETTh1	**0.380 / 0.389**	0.419 / 0.432	0.447 / 0.440	0.454 / 0.447	0.516 / 0.484	0.458 / 0.450
ETTh2	**0.309 / 0.340**	0.339 / 0.380	0.364 / 0.395	0.383 / 0.407	0.391 / 0.411	0.414 / 0.427
ETTm1	**0.333 / 0.342**	0.369 / 0.378	0.381 / 0.395	0.407 / 0.410	0.406 / 0.407	0.400 / 0.406
ETTm2	**0.247 / 0.294**	0.269 / 0.320	0.275 / 0.323	0.288 / 0.332	0.290 / 0.334	0.291 / 0.333
Macro avg.	**0.279 / 0.294**	0.307 / 0.324	0.331 / 0.347	0.332 / 0.343	0.363 / 0.369	0.384 / 0.372

Table 3 compares supervised StatTS-FM with zero-shot Sundial-L, Time-MoE-U (Shi et al., 2025), Timer-XL (Liu et al., 2025a), Moirai-B (Woo et al., 2024), and Chronos-L (Ansari et al., 2024), using baseline results reported by Liu et al. (2025b, Table 1). StatTS-FM has the lowest reported errors on the six datasets, with macro MSE approximately 8.7% below zero-shot Sundial-L; this comparison also includes the effect of supervised adaptation. The longer-context LLM and short-horizon PEMS results appear in Appendix C. The same-backbone controls above assess the statistical design more directly.

Table 3: Supervised StatTS-FM versus published zero-shot foundation models; MSE / MAE averaged over horizons 96, 192, 336, and 720 (lower is better). Time-MoE-U's Electricity entry is omitted because Electricity was included in its pretraining; its macro average is therefore omitted.

Dataset	StatTS-FM	Sundial-L	Time-MoE-U	Timer-XL	Moirai-B	Chronos-L
ETTh1	**0.361 / 0.381**	0.395 / 0.420	0.412 / 0.426	0.404 / 0.417	0.417 / 0.419	0.588 / 0.466
ETTh2	**0.305 / 0.349**	0.334 / 0.387	0.371 / 0.399	0.347 / 0.388	0.362 / 0.382	0.455 / 0.427
ETTm1	**0.302 / 0.337**	0.331 / 0.369	0.356 / 0.391	0.373 / 0.392	0.406 / 0.385	0.555 / 0.465
ETTm2	**0.232 / 0.288**	0.254 / 0.315	0.288 / 0.344	0.273 / 0.336	0.311 / 0.337	0.295 / 0.338
Electricity	**0.152 / 0.239**	0.166 / 0.262	–	0.174 / 0.278	0.187 / 0.274	0.204 / 0.273
Weather	**0.217 / 0.251**	0.238 / 0.275	0.256 / 0.288	0.256 / 0.294	0.287 / 0.281	0.279 / 0.306
Macro avg.	**0.261 / 0.307**	0.286 / 0.338	–	0.305 / 0.351	0.328 / 0.346	0.396 / 0.379

5.4 Additional Evaluations and Cost

Missing inputs and fewer training windows. Under shared 20% MCAR missingness, Full's macro MSE increases by 12.15%, versus 18.00% for Raw-patch (Table 7). In the reduced-window comparison, all runs start from the pretrained backbone with a 5,000-update budget; Full and Raw-patch use the same approximately 10% subset of training windows. MSE increases by 15.74% and 22.42%, respectively (Table 12). Full has lower absolute error and relative degradation in both experiments. Standardization uses the full training segment; only the number of adaptation windows is reduced.

Predictive distributions and cost. Full attains lower Gaussian CRPS and NLL than Raw-patch and the no-drift variant. Its 80% interval coverage is 80.62%, versus 74.21% without drift adjustment, with a wider mean interval (0.9301 versus 0.8363; Table 9). This coverage–width trade-off concerns predictive uncertainty, separately from the mean-trajectory diagnostics. Under ETTh1 profiling (Table 13), Full takes 132 versus 81 minutes for 5,000 updates and processes 410 versus 870 windows/s at inference, with greater memory use than Raw-patch. These costs are configuration-specific.

6 Conclusion

StatTS-FM connects statistical representations, forecast-mean constraints, and drift-conditioned prediction. Same-backbone controls reveal accuracy–fidelity divergence and joint gains from statistical supervision with the architecture fixed. Our analysis relates marginal loss to projected trajectory location and spread, and statistical constraints to generalization under stationary dependence. Twelve benchmarks assess point forecasting, while six-setting controls support the statistical design through component ablations and fidelity diagnostics. These TimesFM-3 results motivate explicit statistical supervision and evaluation; transfer to other backbones and cross-seed stability remain open.

REPRODUCIBILITY STATEMENT

To support reproducibility, we document the model and learning procedure, experimental protocols, and the assumptions underlying our theoretical results. Section 3 and Algorithm 1 specify the model and learning procedure. Appendices B and E define metrics, data splits, comparisons, ablations, and aggregation. Appendices F–G provide the statistical interpretation, theoretical assumptions, and proofs. An anonymized implementation of StatTS-FM is provided in the supplementary material.

AI USE STATEMENT

We used generative AI tools for aiding and polishing manuscript, figures and algorithm pseudocode. The authors take responsibility for the final methods, derivations, experimental results, and conclusions.

REFERENCES

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. ISSN 2835-8856.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015. doi: 10.1007/s10851-014-0506-3.

Defu Cao, Furong Jia, Sercan O. Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. TEMPO: Prompt-based generative pre-trained transformer for time series forecasting. In *International Conference on Learning Representations*, 2024.

Hanyin Cheng, Xingjian Wu, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. CoRA: Boosting time series foundation models for multivariate forecasting through correlation-aware adapter. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=JRlNrcTllN.

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. PMLR, 2024.

Matthew Faw, Rajat Sen, Yichen Zhou, and Abhimanyu Das. In-context fine-tuning for time-series foundation models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 16355–16374. PMLR, 2025.

Matthias Gelbrich. On a formula for the L^2 Wasserstein metric between measures on Euclidean and Hilbert spaces. *Mathematische Nachrichten*, 147(1):185–203, 1990. doi: 10.1002/mana.19901470121.

Tilmann Gneiting. Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106(494):746–762, 2011. doi: 10.1198/jasa.2011.r10138.

Riku Green, Zahraa S. Abdallah, and Telmo M. Silva Filho. Expectations vs. realities: The cost of MSE-optimal forecasting under conditional uncertainty. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. Association for Computing Machinery, 2026. doi: 10.1145/3770855.3818087. URL https://arxiv.org/abs/2606.04342.

Rentao Gu, Yihang Ding, Junjie Li, Yi Ding, Weijing Sang, Xiaoli Huo, Xin Qin, and Yuefeng Ji. FM-LLM: A frequency-enhanced mixture-of-experts framework for adapting LLMs to time series forecasting. *Knowledge-Based Systems*, 341:115776, 2026. doi: 10.1016/j.knosys.2026.115776.

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.

Yuxiao Hu, Qian Li, Dongxiao Zhang, Jinyue Yan, and Yuntian Chen. Context-Alignment: Activating and enhancing LLM capabilities in time series. In *International Conference on Learning Representations*, 2025.

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.

Minkyoung Kim, Daeun Ji, Yohan Lee, Beomsoo Kim, and Beakcheol Jang. CTRL: Control-based time series forecasting with LLM-guided residual learning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 21952–21968, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1104.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Dilfira Kudrat, Zongxia Xie, Yanru Sun, Tianyu Jia, and Qinghua Hu. Patch-wise structural loss for time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31841–31859. PMLR, 2025.

Thomas L. Lee, William Toner, Rajkarn Singh, Artjom Joosen, and Martin Asenov. Lightweight online adaption for time series foundation model forecasts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 33736–33764. PMLR, 2025.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023. URL https://arxiv.org/abs/2305.10721v1.

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. PMLR, 2024b.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-XL: Long-context transformers for unified time series forecasting. In *International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=KMCJXjlDDr.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 39295–39317. PMLR, 2025b.

Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1–2):115–166, 2018. doi: 10.1007/s10107-017-1172-1.

Mehryar Mohri and Afshin Rostamizadeh. Rademacher complexity bounds for non-i.i.d. processes. In *Advances in Neural Information Processing Systems 21*, pp. 1097–1104, 2008.

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.

Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. S^2IP-LLM: Semantic space informed prompt learning with LLM for time series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39135–39153. PMLR, 2024. URL https://proceedings.mlr.press/v235/pan24c.html.

Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. DBLoss: Decomposition-based loss function for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0936.

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. URL https://openreview.net/forum?id=e1wDDFmlVu.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. TEST: Text prototype aligned embedding to activate LLM's ability for time series. In *International Conference on Learning Representations*, 2024.

Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1922.

Hao Wang, Licheng Pan, Yuan Shen, Zhichao Chen, Degui Yang, Yifei Yang, Sen Zhang, Xinggao Liu, Haoxuan Li, and Dacheng Tao. FreDF: Learning to forecast in the frequency domain. In *International Conference on Learning Representations*, 2025a.

Hao Wang, Licheng Pan, Yuan Lu, Zhixuan Chu, Xiaoxi Li, Shuting He, Zhichao Chen, Qingsong Wen, Haoxuan Li, and Zhouchen Lin. DistDF: Time-series forecasting needs joint-distribution Wasserstein alignment. In *International Conference on Learning Representations*, 2026a.

Haochun Wang, Ruichen Xu, Georgios Kementzidis, Karen Cho, Sebastian Ramirez Villarreal, and Yuefan Deng. Towards principled test-time adaptation for time series forecasting. *arXiv preprint arXiv:2605.17250*, 2026b.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. TimeMixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, 2025b.

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. PMLR, 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*, volume 34, pp. 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02186.

Bin Yu. Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability*, 22(1):94–116, 1994. doi: 10.1214/aop/1176988849.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

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*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 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*, volume 36, 2023.

A TRAINING AND INFERENCE PROCEDURE

Trainable parameters Θ comprise the statistical encoder, projections and embeddings, custom normalization and readout layers, drift network, and LoRA factors; the original backbone weights θ_0 are frozen. FORWARD shares Θ and reference state $\mathcal{R} = (\boldsymbol{\mu}^{\mathrm{ref}}, \mathbf{v}^{\mathrm{ref}}, \mathbf{p}^{\mathrm{ref}})$; u enables reference updates. Let $\mathcal{B}$ index the input windows and s_j denote window j's masked mean, variance, and normalized spectral summary (Section 3.3).

Algorithm 1 Training and inference of StatTS-FM

Require: Training and validation windows $(\mathbf{X}, \mathbf{M}, \mathbf{Y})$, using training-set standardization; pretrained weights θ_0; EMA rate $\beta \in (0, 1]$.
Ensure: Selected $(\Theta^*, \mathcal{R}^*)$ and predictive mean/scale for a new history.

Shared forward computation

function FORWARD($\mathbf{X}, \mathbf{M}, u$)
  Form $\mathbf{T}, \mathbf{S}, \mathbf{R}$ and historical ACF/PACF lag summaries (Section 3.1).
  Construct statistical tokens $\mathbf{E}$ and their validity mask.
  Apply TimesFM-3 with LoRA; read out $\boldsymbol{\mu}^0, \boldsymbol{\sigma}^0$ (Eq. 4).
  Compute s_j and $\bar{s} \leftarrow |\mathcal{B}|^{-1} \sum_{j \in \mathcal{B}} s_j$; $f \leftarrow u$ **and** ($\mathcal{R}$ is uninitialized).
  if f **then** $\mathcal{R} \leftarrow \mathrm{stopgrad}(\bar{s})$.
  Compute drift $\mathbf{d}$ against $\mathcal{R}$ (Eq. 5).
  if u **and not** f **then** $\mathcal{R} \leftarrow (1 - \beta)\mathcal{R} + \beta\, \mathrm{stopgrad}(\bar{s})$.
  $(\mathbf{b}, \mathbf{a}) \leftarrow g_\psi(\mathbf{d})$; $\mathbf{a} \leftarrow \mathrm{clip}(\mathbf{a}, -5, 5)$.
  $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}^0 + \mathbf{b}$; $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma}^0 \odot \exp(\mathbf{a})$.
  return $\boldsymbol{\mu}, \boldsymbol{\sigma}$ and intermediates for the loss.
end function

Training

Initialize Θ, AdamW, and fixed sliced-Wasserstein projections; $\mathcal{R} \leftarrow$ uninitialized.
for each epoch in training mode **do**
  for each training minibatch $\mathcal{B}$ **do**
    FORWARD($\mathbf{X}, \mathbf{M}$, true).
    Compute batch-mean $\mathcal{L}$ against $\mathbf{Y}$ (Eq. 8); update Θ with AdamW.
  end for
  Validate with FORWARD($\mathbf{X}, \mathbf{M}$, false) using the total loss.
  Save $(\Theta, \mathcal{R})$ on improvement; stop when patience is exhausted.
end for
Restore $(\Theta^*, \mathcal{R}^*)$ with the lowest validation total loss.

Inference

Fix $(\Theta^*, \mathcal{R}^*)$; use evaluation mode without gradients.
For a new history $(\mathbf{X}, \mathbf{M})$, return $\boldsymbol{\mu}, \boldsymbol{\sigma}$ from FORWARD($\mathbf{X}, \mathbf{M}$, false).

References average window-level statistics, including variances, and persist across epochs. Training clears and clips gradients and follows the schedule in Appendix E; validation uses evaluation mode without gradients. Targets enter only the loss: NLL uses the calibrated mean and scale; statistical fidelity losses use the calibrated mean.

B Evaluation Metrics

The six-setting experiments evaluate standardized predictions and targets. For setting s, let N_s be the number of test windows and write $\widehat{\mathbf{Y}}_i = \boldsymbol{\mu}_i \in \mathbb{R}^{H\times C}$ for the predictive mean. Dataset statistics are fitted on the training partition. We first aggregate within each setting and then average its metric equally with those of the other five settings. The fidelity diagnostics target the same statistical families as the regularizers in Equation 7; the definitions below specify their differences.

Point error and marginal fidelity. MSE and MAE average squared and absolute errors over windows, horizons, and variables. Marg-W1 instead compares the empirical distributions of values within each window and variable. If $\widehat{y}_{i,(h),c}$ and $y_{i,(h),c}$ are the independently sorted forecast and target values, respectively, then

$$\text{Marg-W1}_s = \frac{1}{N_s C H} \sum_{i=1}^{N_s} \sum_{c=1}^{C} \sum_{h=1}^{H} |\widehat{y}_{i,(h),c} - y_{i,(h),c}|. \tag{11}$$

The forecast and target windows are not standardized separately. This is a per-variable marginal diagnostic, rather than the multivariate sliced loss used in training.

Spectral fidelity. We use the normalized powers P_f defined in Section 3.4, with $\epsilon = 10^{-6}$ and $f = 0, \ldots, \lfloor H/2 \rfloor$. The orthonormal one-sided transform includes DC, without frequency-bin pooling or doubling of interior bins. Zero-energy sequences have zero normalized power. We compute

$$\text{Spec-Err}_s = \frac{1}{N_s C} \sum_{i,c} \sum_{f} |P_f(\widehat{\mathbf{Y}}_{i,:,c}) - P_f(\mathbf{Y}_{i,:,c})|. \tag{12}$$

Unlike $\mathcal{L}_{\text{freq}}$, this diagnostic measures absolute normalized power differences without a logarithm.

Lag-dependence fidelity. ACF and stabilized Levinson–Durbin PACF use the estimators in Section 3.1. The maximum lag is $D_H = \min(24, \lfloor H/2 \rfloor, H-1)$. Each target window supplies a deduplicated set Λ_i: powers of two up to D_H, together with the top eight ACF and top eight PACF lags, ranked by channel-averaged absolute value. The same selected lags are used for forecast and target. Define

$$e_{ic\ell} = |\widehat{\rho}_{ic}(\ell) - \rho^Y_{ic}(\ell)| + |\widehat{\pi}_{ic}(\ell) - \pi^Y_{ic}(\ell)|. \tag{13}$$

For an evaluation batch $\mathcal{B}$, the metric implementation averages all valid lag entries and variables; batches are then weighted by window count:

$$\text{Corr-Err}_s = \frac{1}{N_s} \sum_{\mathcal{B}} |\mathcal{B}| \frac{\sum_{i\in\mathcal{B}} \sum_c \sum_{\ell\in\Lambda_i} e_{ic\ell}}{C \sum_{i\in\mathcal{B}} |\Lambda_i|}. \tag{14}$$

Padded lag slots are excluded. ACF and PACF receive equal weight, with no lag-decay weights; thus this diagnostic is distinct from $\mathcal{L}_{\text{corr}}$. Singleton batches reduce to a mean of per-window errors. For larger batches, windows with more selected lags contribute more within that batch, so the evaluation batching must be held fixed across methods.

Gaussian predictive scores. Let $z = (y-\mu)/\sigma$, with σ denoting standard deviation, and let ϕ and Φ be the standard normal density and CDF. We use analytic Gaussian CRPS and Gaussian NLL:

$$\begin{aligned} \text{CRPS}(\mu, \sigma; y) &= \sigma \left[z(2\Phi(z) - 1) + 2\phi(z) - \frac{1}{\sqrt{\pi}} \right], \\ \text{NLL}(\mu, \sigma; y) &= \frac{1}{2} \left[\log(2\pi\bar{v}) + (y-\mu)^2/\bar{v} \right], \quad \bar{v} = \max(\sigma^2, 10^{-6}). \end{aligned} \tag{15}$$

CRPS uses σ floored at 10^{-6}. For the 80% central interval, $[l, u] = [\mu - q\sigma, \mu + q\sigma]$ with $q = \Phi^{-1}(0.9)$, PICP80 averages $\mathbf{1}\{l \le y \le u\}$ and MPIW80 averages $u - l$. All four metrics average over forecast elements within each setting, then over settings. The probabilistic table compares Gaussian outputs only; the native zero-shot model is included in the point-forecast comparison.

C Additional Benchmark Comparisons

LLM-based forecasters. Table 4 compares FSCA (Hu et al., 2025), S^{2}IP-LLM (Pan et al., 2024), Time-LLM (Jin et al., 2024), and GPT4TS (Zhou et al., 2023). StatTS-FM has the lowest reported horizon-averaged MSE and MAE in the listed settings. Across the seven non-ILI datasets, macro MSE decreases from 0.296 for FSCA to 0.257, approximately 13.1% lower based on the tabulated values. ILI is reported separately.

Table 4: Comparison with LLM-based forecasting methods. Except for ILI, input length is 512 and results are averaged over horizons 96, 192, 336, and 720; ILI uses horizons 24, 36, 48, and 60. Each entry is MSE / MAE.

Dataset	StatTS-FM	FSCA	S^{2}IP-LLM	Time-LLM	GPT4TS
ILI	**1.208 / 0.692**	1.380 / 0.783	1.552 / 0.826	1.713 / 0.858	1.925 / 0.903
Weather	**0.195 / 0.229**	0.224 / 0.262	0.228 / 0.265	0.237 / 0.269	0.237 / 0.270
Electricity	**0.139 / 0.216**	0.159 / 0.252	0.166 / 0.262	0.167 / 0.264	0.167 / 0.263
Traffic	**0.337 / 0.229**	0.386 / 0.263	0.405 / 0.286	0.407 / 0.289	0.414 / 0.294
ETTh1	**0.338 / 0.365**	0.394 / 0.424	0.418 / 0.436	0.426 / 0.435	0.427 / 0.426
ETTh2	**0.269 / 0.317**	0.316 / 0.375	0.355 / 0.399	0.361 / 0.398	0.354 / 0.394
ETTm1	**0.302 / 0.331**	0.342 / 0.378	0.346 / 0.382	0.354 / 0.384	0.352 / 0.383
ETTm2	**0.219 / 0.278**	0.250 / 0.314	0.262 / 0.326	0.275 / 0.334	0.266 / 0.326
Macro avg. (excl. ILI)	**0.257 / 0.281**	0.296 / 0.324	0.311 / 0.337	0.318 / 0.339	0.317 / 0.337

Short-horizon traffic forecasting. Table 5 compares PEMS03 and PEMS08, additionally including RLinear (Li et al., 2023) and Crossformer (Zhang & Yan, 2023). StatTS-FM has the lowest reported MSE and MAE on both datasets, with macro MSE/MAE of $0.120/0.204$, versus $0.132/0.224$ for iTransformer.

Table 5: PEMS03 and PEMS08 forecasting results averaged over horizons 12, 24, 48, and 96 with input length 96. Each entry is MSE / MAE.

Dataset	StatTS-FM	iTransformer	RLinear	PatchTST	Crossformer	TimesNet
PEMS03	**0.103 / 0.202**	0.113 / 0.221	0.495 / 0.472	0.180 / 0.291	0.169 / 0.281	0.147 / 0.248
PEMS08	**0.137 / 0.206**	0.150 / 0.226	0.529 / 0.487	0.280 / 0.321	0.268 / 0.307	0.193 / 0.271
Macro avg.	**0.120 / 0.204**	0.132 / 0.224	0.512 / 0.480	0.230 / 0.306	0.219 / 0.294	0.170 / 0.260

D Supplementary Experimental Results

The following experiments use ETTh1, Electricity, and Weather with input length $L = 96$ and horizons $H \in \{96, 336\}$. Errors are measured on the training-set standardized scale. Paired entries denote MSE / MAE, and macro averages weight the six dataset–horizon settings equally. Relative changes are computed from unrounded macro averages. Results are point estimates; cross-seed uncertainty and statistical significance are not reported.

D.1 Component Ablations

Table 6 gives the individual settings underlying the component comparison. Removing dependence tokens retains the correlation loss. Removing drift adjustment retains the Gaussian readout but removes the adjustment regularizer. Removing decomposition also removes its auxiliary regularizer and changes the component readout. These comparisons measure the contribution of each component within the full system; changes in capacity and input structure are not separately controlled.

Table 6: Component ablations. All entries are MSE / MAE; lower is better. "Stat." removes all three statistical losses, "drift" removes location and scale adjustment, "decomp." removes decomposition, and "dep." removes dependence tokens.

Dataset	H	Full	w/o stat.	w/o drift	w/o decomp.	w/o dep.
ETTh1	96	0.350 / 0.372	0.361 / 0.382	0.362 / 0.379	0.368 / 0.384	0.357 / 0.378
ETTh1	336	0.408 / 0.415	0.418 / 0.425	0.436 / 0.432	0.426 / 0.427	0.420 / 0.422
Electricity	96	0.132 / 0.219	0.136 / 0.224	0.137 / 0.226	0.140 / 0.227	0.137 / 0.224
Electricity	336	0.162 / 0.242	0.166 / 0.247	0.174 / 0.253	0.174 / 0.253	0.167 / 0.246
Weather	96	0.158 / 0.200	0.164 / 0.207	0.168 / 0.210	0.168 / 0.210	0.161 / 0.205
Weather	336	0.227 / 0.267	0.235 / 0.274	0.240 / 0.280	0.240 / 0.280	0.233 / 0.273

D.2 Missing Historical Inputs

Models trained on clean data are evaluated after masking originally observed historical entries independently with probability 0.2 (MCAR). Full and Raw-patch share the masks; masked values are set to zero after standardization, with no additional mask input, and future targets are unchanged. It does not separately assess the contribution of the mask-aware components. Table 7 includes clean-input controls. Full's macro MSE increases by 12.15%, compared with 18.00% for Raw-patch; the corresponding MAE increases are 7.90% and 10.67%. Full retains both lower absolute error and a smaller relative degradation under this missingness protocol.

Table 7: Clean and 20% MCAR-missing inputs. Each entry is MSE / MAE.

Dataset	H	Full, clean	Full, missing	Raw, clean	Raw, missing
ETTh1	96	0.3500 / 0.3720	0.3960 / 0.4020	0.3770 / 0.3890	0.4510 / 0.4330
ETTh1	336	0.4080 / 0.4150	0.4580 / 0.4490	0.4390 / 0.4360	0.5210 / 0.4850
Electricity	96	0.1320 / 0.2190	0.1470 / 0.2350	0.1450 / 0.2330	0.1650 / 0.2550
Electricity	336	0.1620 / 0.2420	0.1810 / 0.2630	0.1780 / 0.2590	0.2040 / 0.2840
Weather	96	0.1580 / 0.2000	0.1770 / 0.2210	0.1720 / 0.2160	0.2000 / 0.2420
Weather	336	0.2270 / 0.2670	0.2526 / 0.2804	0.2490 / 0.2860	0.2998 / 0.3140
Macro average	–	0.2395 / 0.2858	0.2686 / 0.3084	0.2600 / 0.3032	0.3068 / 0.3355

D.3 Same-Backbone Comparisons

Table 8 separates supervised adaptation from native zero-shot prediction. Full and Raw-patch freeze the same backbone revision with identical query/value LoRA configurations and comparable update and hyperparameter-search budgets; native TimesFM-3 retains its official input/output path. The Full–Raw-patch comparison measures the complete statistical design relative to ordinary patch-based adaptation. The native TimesFM-3 comparison also includes the effect of supervised adaptation. All three use a pretrained backbone, so this experiment does not isolate the contribution of pretraining itself.

Table 8: Same-backbone comparison (MSE / MAE). Full and Raw-patch use supervised adaptation; native TimesFM-3 uses zero-shot prediction.

Dataset	H	Full	Raw-patch	Native TimesFM-3
ETTh1	96	0.350 / 0.372	0.377 / 0.389	0.402 / 0.415
ETTh1	336	0.408 / 0.415	0.439 / 0.436	0.464 / 0.454
Electricity	96	0.132 / 0.219	0.145 / 0.233	0.169 / 0.257
Electricity	336	0.162 / 0.242	0.178 / 0.259	0.205 / 0.281
Weather	96	0.158 / 0.200	0.172 / 0.216	0.191 / 0.241
Weather	336	0.227 / 0.267	0.249 / 0.286	0.274 / 0.310
Macro average	–	0.2395 / 0.2858	0.2600 / 0.3032	0.2842 / 0.3263

D.4 PROBABILISTIC FORECASTS

Table 9 evaluates Gaussian forecasts using analytic CRPS, Gaussian NLL, and central 80% prediction intervals. NLL and CRPS are averaged over forecast elements within each setting, followed by an equal average over the six settings. Full has lower CRPS and NLL, with coverage closer to the nominal level, but also wider intervals. This result supports better distributional scores and coverage at the evaluated level, without establishing calibration at every level or within every subset.

Table 9: Gaussian probabilistic forecasts, averaged over six settings. Lower CRPS and NLL are better. PICP_{80} is coverage of central 80% intervals; MPIW_{80} is their mean width.

Method	CRPS ↓	NLL ↓	PICP_{80}	MPIW_{80}
Full	0.2021	0.1096	80.62%	0.9301
w/o drift	0.2112	0.1904	74.21%	0.8363
Raw-patch	0.2151	0.2041	75.63%	0.8814

D.5 INDIVIDUAL STATISTICAL LOSSES

Table 10 provides the complete macro averages behind Figure 1, including MAE and all three single-loss ablations.

Table 10: Same-backbone control and statistical-loss ablations: six-setting macro averages. Lower is better for all metrics.

Variant	MSE	MAE	Marg-W1	Spec-Err	Corr-Err
Full	**0.2395**	**0.2858**	**0.143**	**0.184**	**0.082**
Raw-patch	0.2600	0.3032	0.170	0.247	0.108
Without all three losses	0.2467	0.2932	0.183	0.274	0.122
Without marginal loss	0.2438	0.2905	0.170	0.193	0.085
Without frequency loss	0.2446	0.2910	0.148	0.246	0.088
Without correlation loss	0.2442	0.2908	0.150	0.199	0.114

Table 11 compares separately trained variants that each remove one statistical loss. The full objective need not give the best point error in every setting: without the marginal loss, ETTh1 at $H = 96$ has slightly lower MSE and MAE. The macro-average advantage therefore should not be interpreted as a uniform gain in every dataset–horizon pair. Jointly removing all three losses tests a different intervention from these individual ablations.

Table 11: Individual statistical-loss ablations (MSE / MAE). Each variant removes only the named loss.

Dataset	H	Full	w/o marginal	w/o frequency	w/o correlation
ETTh1	96	0.3500 / 0.3720	0.3490 / 0.3710	0.3560 / 0.3760	0.3540 / 0.3750
ETTh1	336	0.4080 / 0.4150	0.4170 / 0.4230	0.4190 / 0.4240	0.4160 / 0.4210
Electricity	96	0.1320 / 0.2190	0.1350 / 0.2230	0.1350 / 0.2230	0.1360 / 0.2240
Electricity	336	0.1620 / 0.2420	0.1650 / 0.2470	0.1660 / 0.2470	0.1670 / 0.2460
Weather	96	0.1580 / 0.2000	0.1610 / 0.2060	0.1630 / 0.2060	0.1620 / 0.2050
Weather	336	0.2270 / 0.2670	0.2358 / 0.2730	0.2286 / 0.2700	0.2302 / 0.2738

D.6 REDUCED TRAINING WINDOWS

For a training segment of length T_{train}, stride-one extraction gives $T_{\text{train}} - 96 - H + 1$ windows; the reduced set contains the ceiling of 10% of that count. Full and Raw-patch share the reduced-window indices. All groups in Table 12 start adaptation from the pretrained backbone with a 5,000-update budget; the reduced-window models are not continued from full-window adaptation. Standardization uses the full training segment, so this experiment reduces adaptation windows rather than access to raw data. Full's macro MSE and MAE increase by 15.74% and 8.38%, compared with 22.42% and 10.60% for Raw-patch. Full also has lower absolute error with the reduced window set.

Table 12: Full and approximately 10% training-window budgets (MSE / MAE). The window-count column gives 100% / 10% counts.

Dataset	H	Windows	Full, 100%	Full, 10%	Raw, 100%	Raw, 10%
ETTh1	96	8,449 / 845	0.3500 / 0.3720	0.4040 / 0.4100	0.3770 / 0.3890	0.4660 / 0.4470
ETTh1	336	8,209 / 821	0.4080 / 0.4150	0.4740 / 0.4480	0.4390 / 0.4360	0.5410 / 0.4890
Electricity	96	18,221 / 1,823	0.1320 / 0.2190	0.1540 / 0.2340	0.1450 / 0.2330	0.1770 / 0.2550
Electricity	336	17,981 / 1,799	0.1620 / 0.2420	0.1860 / 0.2610	0.1780 / 0.2590	0.2160 / 0.2850
Weather	96	36,696 / 3,670	0.1580 / 0.2000	0.1820 / 0.2160	0.1720 / 0.2160	0.2110 / 0.2410
Weather	336	36,456 / 3,646	0.2270 / 0.2670	0.2632 / 0.2898	0.2490 / 0.2860	0.2988 / 0.2948
Macro average	–	–	0.2395 / 0.2858	0.2772 / 0.3098	0.2600 / 0.3032	0.3183 / 0.3353

D.7 Computational Cost

Table 13 reports profiling measurements on ETTh1 with $L = H = 96$, one A100 80GB GPU, batch size 32, BF16, and 5,000 training updates. This configuration uses 12 patch tokens per stream and 16 additional tokens when dependence encoding is enabled. Its token and parameter counts are specific to this configuration. Full and the variant without statistical losses have the same network, token count, and reported inference throughput; the loss terms add training cost. Relative to Raw-patch, Full uses more trainable parameters, memory, and training time, with lower inference throughput. The predictive gains thus come with a computational tradeoff, and these measurements should not be extrapolated to other hardware or sequence lengths.

Table 13: Computational cost under the reported profiling configuration described above. Parameters are trainable parameters; memory is peak GPU memory. Training time covers 5,000 updates, and throughput is in windows/s.

Method	Parameters	Tokens	Memory	Time	Throughput
Full	6.84 M	52	13.2 GB	132 min	410
Raw-patch	4.12 M	12	8.1 GB	81 min	870
w/o stat.	6.84 M	52	12.7 GB	118 min	410
w/o drift	6.22 M	52	12.9 GB	125 min	435
w/o decomp.	4.80 M	28	9.4 GB	94 min	655
w/o dep.	6.78 M	36	11.1 GB	109 min	540

E EXPERIMENTAL PROTOCOL REFERENCE

Table 14 indexes settings and results. Benchmark comparisons average over horizons within datasets, then across datasets; controlled experiments weight six dataset–horizon settings equally.

E.1 COMPARISON SETTINGS

Table 14: Comparison families and their reported evaluation ranges. Lengths are in time steps. The profiling row describes its own configuration.

Comparison	Context and horizons	Coverage and results
Dedicated forecasters	$L = 96$; $H \in \{96, 192, 336, 720\}$	Nine datasets in Table 2.
Foundation models	$H \in \{96, 192, 336, 720\}$; context not specified per model	ETTh1/2, ETTm1/2, Electricity, Weather; Table 3.
LLM-based forecasters	$L = 512$; $H \in \{96, 192, 336, 720\}$; ILI: $L = 36$, $H \in \{24, 36, 48, 60\}$	Eight datasets in Table 4; ILI is excluded from the macro average.
Short-horizon traffic	$L = 96$; $H \in \{12, 24, 48, 96\}$	PEMS03 and PEMS08; Table 5.
Controlled experiments	$L = 96$; $H \in \{96, 336\}$	ETTh1, Electricity, Weather; Appendix D.
Profiling	$L = H = 96$	ETTh1; Table 13.

Data partitions. Chronological splits are 12/4/4 months for ETT, 6:2:2 for PEMS, and 7:1:2 otherwise, with standardization fitted on the training partition (Section 5.1). For controlled experiments, training/test segment lengths T_{train} and T_{test} are 8,640/2,880 for ETTh1, 18,412/5,260 for Electricity, and 36,887/10,539 for Weather. With stride one, the training and test window counts are

$$N_{\text{train}} = T_{\text{train}} - L - H + 1, \qquad N_{\text{test}} = T_{\text{test}} - H + 1. \tag{16}$$

Test histories may precede the test segment, but each forecast target lies entirely within it. Per-setting training-window counts are reported in Table 12.

Comparison identities. Full and Raw-patch adapt the same frozen TimesFM-3 revision, with identical query/value LoRA configurations and comparable update and search budgets. Raw-patch uses ordinary patches and a Gaussian readout. Native TimesFM-3 (Table 8) uses the official input/output path zero-shot. For the separate foundation-model comparison (Table 3), StatTS-FM is supervised, while the baseline entries reproduce the zero-shot results in Liu et al. (2025b, Table 1). This comparison does not hold adaptation regimes fixed. StatTS-FM's context length in this comparison is unspecified. Time-MoE-U's Electricity entry is omitted owing to pretraining overlap, and no macro average is reported for that model.

E.2 TRAINING AND CONTROLLED VARIANTS

Representation and optimization. Component patches have length 16 and stride 8 (8 and 4 for ILI). Moving-average kernels are $\{3, 7, 15, 31\}$ ($\{3, 5, 7\}$ for ILI). The candidate-lag limit is $\min(24, \lfloor L/2 \rfloor, L - 1)$, with geometric lags and the top eight ACF and PACF lags. StatTS-FM's benchmark settings are LoRA rank 8, scaling 16, dropout 0.1; AdamW with learning rates 10^{-4} for added modules and 10^{-5} for LoRA, weight decay 10^{-4}, cosine decay with 5% warmup, and gradient clipping at 1.0. This benchmark schedule allows up to 30 epochs, with patience five. Algorithm 1 specifies selection by validation total loss, restoring parameters and EMA references together. Experiment-specific update budgets are given below.

Full and no-stat share the same network; decomposition and drift ablations change architecture or capacity. No-decomposition retains the other statistical mechanisms, unlike Raw-patch (Table 15).

Reduced-window adaptation. Both full-window and approximately 10% runs start from the pretrained backbone with a 5,000-update budget (Table 12). Full and Raw-patch share $\lceil 0.1 N_{\text{train}} \rceil$ training-window indices. Standardization uses the complete training segment; the reduction concerns adaptation windows, not raw-data access.

Table 15: Boundaries of the controlled ablations relative to Full. Each variant is adapted separately. All retain a Gaussian predictive readout.

Variant	Removed or replaced	Retained or adjusted
Without statistical losses	All three statistical losses	Full network, Gaussian NLL, separation and adjustment regularizers.
Without drift adjustment	History-conditioned location and scale adjustment	Gaussian readout and statistical losses; $\mathcal{L}_{\mathrm{cal}}$ is omitted.
Without decomposition	Three component streams and mean heads, replaced by one raw-input stream and head	Dependence tokens, drift adjustment and statistical losses; $\mathcal{L}_{\mathrm{sep}}$ is omitted.
Without dependence tokens	Input lag-summary tokens	The correlation loss and other applicable components.
Without marginal loss	$\mathcal{L}_{\mathrm{marg}}$	Other losses and the Full architecture.
Without frequency loss	$\mathcal{L}_{\mathrm{freq}}$	Other losses and the Full architecture.
Without correlation loss	$\mathcal{L}_{\mathrm{corr}}$	Other losses, dependence tokens and the Full architecture.

Profiling. Table 13 uses one A100 80GB GPU, batch size 32, BF16, and 5,000 updates on ETTh1 at $L = H = 96$, with 12 patch tokens per component and 16 dependence tokens when enabled. This batch size describes profiling, not other tables' evaluation batching.

E.3 Evaluation and Aggregation

Point and distributional outputs. Controlled point and trajectory-statistic evaluations use the final mean on the training-standardized scale, without separate window standardization. Gaussian NLL and analytic CRPS use mean and standard deviation; intervals are central 80% (Appendix B). The no-drift probabilistic baseline is independently trained, not disabled at inference.

Spectral and dependence diagnostics. Spec-Err sums absolute differences of normalized power, including DC, without the training loss's logarithm. Corr-Err weights valid ACF/PACF errors equally, without lag-decay weights. Appendix B defines selected lags and within-batch aggregation; evaluation batching must be fixed across methods.

Missing inputs. Missing-input evaluation uses clean-trained Full and Raw-patch with shared 20% MCAR corruption of observed history entries, zero-filled after standardization. Targets are unchanged; no additional corruption mask is supplied to the model.

Aggregation and relative changes. Let e_s be the error for one of the six controlled dataset–horizon settings, with the within-setting aggregation defined in Appendix B. We use

$$\bar{e} = \frac{1}{6}\sum_{s=1}^{6} e_s, \qquad \mathrm{Reduction}(A; B) = 100\frac{\bar{e}_B - \bar{e}_A}{\bar{e}_B}\,\%. \tag{17}$$

Degradation divides the error increase by the same method's clean-input or full-window reference error. Percentages use unrounded macro errors, not averages of per-setting percentages. Results are point estimates, without cross-seed standard deviations or significance claims.

F MATHEMATICAL INTERPRETATION OF STATISTICAL REGULARIZATION

We first clarify the squared-error target and the empirical distribution in Section 3.4, then derive the marginal loss's control of projected location and spread.

Conditional averaging and trajectory variation. Let $\mathcal{I} = \sigma(\mathbf{X}, \mathbf{M})$ denote the available historical information, and let $\mathbf{y} = \mathrm{vec}(\mathbf{Y}) \in \mathbb{R}^{HC}$ satisfy $\mathbb{E}\|\mathbf{y}\|_2^2 < \infty$. Write $\mathbf{m} = \mathbb{E}[\mathbf{y} \mid \mathcal{I}]$ and $\mathbf{\Sigma} = \mathrm{Cov}(\mathbf{y} \mid \mathcal{I})$. For any square-integrable, $\mathcal{I}$-measurable point forecast $\mathbf{f}$,

$$\mathbb{E}[\|\mathbf{y} - \mathbf{f}\|_2^2 \mid \mathcal{I}] = \mathbb{E}[\|\mathbf{y} - \mathbf{m}\|_2^2 \mid \mathcal{I}] + \|\mathbf{m} - \mathbf{f}\|_2^2. \tag{18}$$

Thus, $\mathbf{m}$ minimizes conditional squared error (Gneiting, 2011). For a fixed linear operator $D \in \mathbb{R}^{q \times HC}$, we also have

$$\mathbb{E}[\|D\mathbf{y}\|_2^2 \mid \mathcal{I}] = \|D\mathbf{m}\|_2^2 + \mathrm{tr}(D\mathbf{\Sigma}D^\top). \tag{19}$$

To derive both identities, write $\mathbf{y} = \mathbf{m} + \boldsymbol{\varepsilon}$ with $\mathbb{E}[\boldsymbol{\varepsilon} \mid \mathcal{I}] = 0$ and expand the squared norms. The cross terms vanish, while $\mathbb{E}[\boldsymbol{\varepsilon}^\top D^\top D\boldsymbol{\varepsilon} \mid \mathcal{I}] = \mathrm{tr}(D\mathbf{\Sigma}D^\top)$.

Choosing D to take first differences along each variable's forecast trajectory makes Equation 19 a statement about roughness: the conditional mean has no greater difference energy than the conditional expected energy of realized futures. The gap is positive when $\mathrm{tr}(D\mathbf{\Sigma}D^\top) > 0$. This comparison holds in conditional expectation, not for every realization, and requires neither stationarity nor independent future observations. Equation 18 characterizes the Bayes minimum of population MSE, which statistical penalties cannot lower. Our regularized Gaussian objective introduces additional inductive biases whose accuracy–fidelity trade-off is assessed empirically. The fixed-linear-operator identity does not directly apply to normalized log spectra or target-selected lag statistics.

The distribution matched by the marginal loss. For a window $\mathbf{U} \in \mathbb{R}^{H \times C}$ and the fixed unit directions $\mathbf{v}_k$ used in training, define

$$\nu_{\mathbf{U}} = \frac{1}{H}\sum_{h=1}^{H} \delta_{\mathbf{U}_{h,:}}, \qquad \nu_{\mathbf{U}}^k = \frac{1}{H}\sum_{h=1}^{H} \delta_{\mathbf{v}_k^\top \mathbf{U}_{h,:}}, \tag{20}$$

where δ is a unit point mass. These are empirical distributions of values within a window, without an independence assumption on its time points. For equally weighted one-dimensional empirical measures, squared Wasserstein distance pairs the sorted values (Bonneel et al., 2015). Write ν_{pred}^k and ν_{target}^k for the projections of $\widehat{\mathbf{Y}} = \boldsymbol{\mu}$ and $\mathbf{Y}$, respectively. Then

$$\mathcal{L}_{\mathrm{marg}} = \frac{1}{K}\sum_{k=1}^{K} W_2^2(\nu_{\mathrm{pred}}^k, \nu_{\mathrm{target}}^k) = \frac{1}{KH}\sum_{k=1}^{K} \|s_k(\widehat{\mathbf{Y}}) - s_k(\mathbf{Y})\|_2^2. \tag{21}$$

This is an average over the chosen finite set of projections. It compares C-dimensional value distributions, rather than the conditional law of an ordered future trajectory in $\mathbb{R}^{HC}$. Any permutation of the time rows leaves $\nu_{\mathbf{U}}$ unchanged, so the marginal loss can vanish despite temporal misalignment. Agreement on finitely many projections also need not identify an arbitrary multivariate distribution. Point alignment, temporal statistics, and probabilistic quality therefore remain distinct evaluation questions. Appendix B also distinguishes the per-variable Marg-W1 diagnostic from this projected, squared training loss.

Projected location and spread. The marginal loss directly controls two interpretable properties of the forecast values. For a window $\mathbf{U} \in \mathbb{R}^{H \times C}$, define the empirical mean and standard deviation along direction $\mathbf{v}_k$ by

$$\begin{aligned} m_k(\mathbf{U}) &= \frac{1}{H}\sum_{h=1}^{H} \mathbf{v}_k^\top \mathbf{U}_{h,:}, \\ \tau_k(\mathbf{U}) &= \left[\frac{1}{H}\sum_{h=1}^{H} \big(\mathbf{v}_k^\top \mathbf{U}_{h,:} - m_k(\mathbf{U})\big)^2\right]^{1/2}. \end{aligned} \tag{22}$$

Here τ_k measures dispersion across the forecast horizon; it is distinct from the predictive scale $\boldsymbol{\sigma}$ and from variability across training runs. The next proposition applies the standard Wasserstein moment bound (Gelbrich, 1990) to each fixed projection.

Proposition 1 (Projected moment control) *For any two windows $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{H\times C}$, the finite-projection marginal loss in Equation 21 satisfies*

$$\frac{1}{K}\sum_{k=1}^{K}\Big[\big(m_k(\mathbf{U}) - m_k(\mathbf{V})\big)^2 + \big(\tau_k(\mathbf{U}) - \tau_k(\mathbf{V})\big)^2\Big] \le \mathcal{L}_{\mathrm{marg}}(\mathbf{U}, \mathbf{V}). \tag{23}$$

Proof. Fix k and write $\mathbf{a} = s_k(\mathbf{U})$ and $\mathbf{b} = s_k(\mathbf{V})$ for the sorted projections. Sorting preserves means and standard deviations. Let $\mathbf{a}_c = \mathbf{a} - m_k(\mathbf{U})\mathbf{1}$ and $\mathbf{b}_c = \mathbf{b} - m_k(\mathbf{V})\mathbf{1}$. Both centered vectors are orthogonal to $\mathbf{1}$, so

$$\begin{aligned}\frac{1}{H}\|\mathbf{a}-\mathbf{b}\|_2^2 &= \big(m_k(\mathbf{U}) - m_k(\mathbf{V})\big)^2 + \frac{1}{H}\|\mathbf{a}_c - \mathbf{b}_c\|_2^2 \\ &\ge \big(m_k(\mathbf{U}) - m_k(\mathbf{V})\big)^2 + \frac{1}{H}\big(\|\mathbf{a}_c\|_2 - \|\mathbf{b}_c\|_2\big)^2 \\ &= \big(m_k(\mathbf{U}) - m_k(\mathbf{V})\big)^2 + \big(\tau_k(\mathbf{U}) - \tau_k(\mathbf{V})\big)^2.\end{aligned} \tag{24}$$

The inequality is the reverse triangle inequality. Averaging over the K fixed directions proves the claim. ∎

With $\mathbf{U} = \widehat{\mathbf{Y}} = \boldsymbol{\mu}$ and $\mathbf{V} = \mathbf{Y}$, Equation 23 explains how a small attained marginal loss limits errors in projected location and within-window spread. The result holds for each window without independence or stationarity assumptions. It does not imply that matching realized trajectory statistics lowers population MSE; the conditional-risk identity in Equation 18 still applies.

G Statistical Constraints, Dependent Windows, and Deployment Risk

We combine independent-block generalization bounds (Yu, 1994; Mohri & Rostamizadeh, 2008) with population classes defined by statistical discrepancy. The result applies to a predictor selected by training, including its learned reference states, without requiring a global optimizer. We then extend the bound to a neighborhood of deployment laws.

Examples, predictors, and assumptions. Let (Z_t), indexed by $t \in \mathbb{Z}$, be a stationary process of forecasting windows on a standard Borel space $\mathcal{Z}$, with marginal law $\mathbb{P}$. Define its absolute-regularity coefficient using total variation in $[0, 1]$:

$$\beta_Z(k) = \sup_t \|\mathcal{L}(Z_{s\le t}, Z_{s\ge t+k}) - \mathcal{L}(Z_{s\le t}) \otimes \mathcal{L}(Z_{s\ge t+k})\|_{\mathrm{TV}}. \tag{25}$$

Assume $\beta_Z(k) \to 0$. If Z_t is a measurable function of $(x_{t-L+1}, \dots, x_{t+H})$ from a stationary raw process, then, for $k > q = L + H$, data processing and monotonicity give $\beta_Z(k) \le \beta_x(k-q+1) \le \beta_x(k-q)$. Masks, when random, must be included in the underlying process for this statement; stationarity or mixing is not inferred from chronological splitting.

Let $\mathcal{F}$ be a fixed, data-independent class of complete predictors. An index f includes trainable parameters and every state fixed at inference, including standardization parameters and drift references. A learning procedure may select this index using the entire training sample, provided it remains in $\mathcal{F}$. Thus, data-dependent EMA references are not incorrectly conditioned on as if they had been fixed independently of training. Any bounds defining this class are part of the assumptions.

For each f, let $\ell_f : \mathcal{Z} \to [0, M]$ be a predictive loss and let the statistical discrepancy be

$$d_f = \lambda_{\mathrm{marg}}\mathcal{L}_{\mathrm{marg}}(f;\cdot) + \lambda_{\mathrm{freq}}\mathcal{L}_{\mathrm{freq}}(f;\cdot) + \lambda_{\mathrm{corr}}\mathcal{L}_{\mathrm{corr}}(f;\cdot), \qquad 0 \le d_f \le B, \tag{26}$$

with fixed nonnegative weights and uniform deterministic bounds $M, B > 0$. These weights are fixed for analysis before observing the sample; training and validation may select any predictor in $\mathcal{F}$, including one trained with different weights. A bounded signed loss may be shifted by a fixed constant to this range without changing the risk gap. Assume the classes are pointwise measurable, or otherwise that all suprema below are measurable. Define

$$\begin{aligned} R_{\mathbb{P}}(f) &= \mathbb{E}_{\mathbb{P}}\ell_f(Z), & D_{\mathbb{P}}(f) &= \mathbb{E}_{\mathbb{P}}d_f(Z), \\ \widehat{R}_n(f) &= \frac{1}{n}\sum_{t=1}^{n}\ell_f(Z_t), & \widehat{D}_n(f) &= \frac{1}{n}\sum_{t=1}^{n} d_f(Z_t). \end{aligned} \tag{27}$$

These empirical quantities evaluate one fixed final predictor and state on the training windows; they are not averages of online losses from changing checkpoints or changing EMA states. The discrepancy is evaluated per window.

Block complexity and remainder. Choose integers $a \ge 1$ and $m \ge 1$ with $m = \lfloor n/(2a) \rfloor$, and write $n = 2am + r$, $0 \le r < 2a$. Partition the first $2am$ examples into consecutive blocks of length a. Each parity contains m blocks. Let $\mathbb{P}_a$ be the law of a consecutive length-a block, and for a function h write $\bar{h}(z_1, \dots, z_a) = a^{-1}\sum_{t=1}^{a} h(z_t)$. For a fixed function class $\mathcal{H}$, define the expected block Rademacher complexity with the $1/m$ convention:

$$\mathcal{R}_{m,a}(\mathcal{H}) = \mathbb{E}_{\widetilde{B}_1,\dots,\widetilde{B}_m,\xi}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\xi_i\bar{h}(\widetilde{B}_i)\right], \qquad \widetilde{B}_i \overset{\mathrm{iid}}{\sim} \mathbb{P}_a. \tag{28}$$

The independent signs ξ_i take values ± 1 with equal probability. The definition does not assert independence within a block.

Building on the independent-block arguments of Yu (1994); Mohri & Rostamizadeh (2008), the following lemma handles a finite family of classes and the remainder from incomplete blocks.

Lemma 1 (A finite family of uniform block bounds) *Let $\mathcal{H}_1, \dots, \mathcal{H}_T$ be fixed nonempty function classes, with $h \in [0, C_j]$ for $h \in \mathcal{H}_j$. Set*

$$c_m = 2(m-1)\beta_Z(a), \qquad \alpha = \delta - c_m > 0, \qquad 0 < \delta < 1. \tag{29}$$

With probability at least $1-\delta$, simultaneously for all j and $h \in \mathcal{H}_j$,

$$\mathbb{P}h \le \frac{1}{n}\sum_{t=1}^{n} h(Z_t) + 2\mathcal{R}_{m,a}(\mathcal{H}_j) + C_j\sqrt{\frac{\log(2T/\alpha)}{2m}} + \frac{C_j r}{n}. \tag{30}$$

Proof. First consider m independent blocks from $\mathbb{P}_a$. For one class define $V = \sup_h\{\mathbb{P}h - m^{-1}\sum_i \bar{h}(\widetilde{B}_i)\}$. An independent ghost sample, Jensen's inequality, and sign symmetrization give

$$\begin{aligned}\mathbb{E}V &\le \mathbb{E}\sup_h \frac{1}{m}\sum_i \{\bar{h}(\widetilde{B}'_i) - \bar{h}(\widetilde{B}_i)\} \\ &= \mathbb{E}\sup_h \frac{1}{m}\sum_i \xi_i\{\bar{h}(\widetilde{B}'_i) - \bar{h}(\widetilde{B}_i)\} \\ &\le 2\mathcal{R}_{m,a}(\mathcal{H}_j).\end{aligned} \tag{31}$$

Replacing one block changes V by at most C_j/m. The bounded-differences inequality therefore gives $\Pr\{V > 2\mathcal{R}_{m,a}(\mathcal{H}_j) + u\} \le \exp(-2mu^2/C_j^2)$. Taking $u = C_j\sqrt{\log(2T/\alpha)/(2m)}$ and a union bound over the T classes bounds failure for one independent parity sample by $\alpha/2$.

For either actual parity, consecutive selected blocks are separated by an omitted block of length a. Its joint block law differs in total variation from $\mathbb{P}_a^{\otimes m}$ by at most $(m-1)\beta_Z(a)$. To see this directly, factor the joint law into successive conditional block laws and replace them one at a time by their marginal law. Each replacement contributes at most the expected conditional total-variation distance of the next block from its marginal, bounded by $\beta_Z(a)$ by Equation 25 and data processing. The triangle inequality gives the stated sum. Thus the probability of any event for this parity changes by at most $(m-1)\beta_Z(a)$.

A union bound over the two parities gives total failure at most $\alpha + 2(m-1)\beta_Z(a) = \delta$. No independence between the two parities is required. Averaging their two successful bounds gives the same deviation above the empirical average over the first $n' = 2am$ observations. If $r > 0$,

$$\widehat{\mathbb{P}}_{n'}h - \widehat{\mathbb{P}}_n h = \frac{r}{n}(\widehat{\mathbb{P}}_{n'}h - \widehat{\mathbb{P}}_{\text{tail}}h) \le \frac{C_j r}{n}. \tag{32}$$

For $r = 0$ the difference is zero. This proves Equation 30. ∎

Theorem 1 (Statistical localization with an adaptive grid level) *Before observing the training sample, fix a finite grid $0 \le \varepsilon_1 < \cdots < \varepsilon_J = B$ and the block length a. Let $\mathcal{A} = \{d_f : f \in \mathcal{F}\}$, and use α from Equation 29. Define the deterministic quantities*

$$\begin{aligned} t &= \sqrt{\frac{\log(2(J+1)/\alpha)}{2m}}, \qquad \zeta = 2\mathcal{R}_{m,a}(\mathcal{A}) + Bt + \frac{Br}{n}, \\ \mathcal{F}_{\varepsilon_j+\zeta} &= \{f \in \mathcal{F} : D_{\mathbb{P}}(f) \le \varepsilon_j + \zeta\}, \quad \mathcal{G}_j = \{\ell_f : f \in \mathcal{F}_{\varepsilon_j+\zeta}\}. \end{aligned} \tag{33}$$

Then, with probability at least $1-\delta$, for every j and every $f \in \mathcal{F}$ satisfying $\widehat{D}_n(f) \le \varepsilon_j$,

$$f \in \mathcal{F}_{\varepsilon_j+\zeta}, \qquad R_{\mathbb{P}}(f) \le \widehat{R}_n(f) + 2\mathcal{R}_{m,a}(\mathcal{G}_j) + Mt + \frac{Mr}{n}. \tag{34}$$

In particular, after training one may select any $\widehat{f} \in \mathcal{F}$ and the smallest grid point at least $\widehat{D}_n(\widehat{f})$ and apply this bound at that selected grid point. The endpoint B guarantees that such a point exists.

Proof. The class $\mathcal{A}$, its expected complexity, ζ, and the population classes $\mathcal{G}_j$ are deterministic; they do not use the realized sample. Apply Lemma 1 simultaneously to $\mathcal{A}$ and the at most J nonempty classes $\mathcal{G}_j$, allocating the failure budget as for $T = J+1$. For $\mathcal{A}$, the resulting event gives $D_{\mathbb{P}}(f) \le \widehat{D}_n(f) + \zeta$ for every f. Hence any predictor feasible at level ε_j belongs to $\mathcal{F}_{\varepsilon_j+\zeta}$. On the same event, the predictive bound for $\mathcal{G}_j$ applies to that predictor. Empty local classes have no feasible predictor on this event and require no additional bound. Since the event holds simultaneously for the entire grid and all predictors, selection of j and f using training data is valid. ∎

Writing $\mathcal{E}_m(\mathcal{F}_{\varepsilon_j+\zeta}) = 2\mathcal{R}_{m,a}(\mathcal{G}_j) + Mt + Mr/n$ gives the estimation term in Section 4. Class nesting implies $\mathcal{R}_{m,a}(\mathcal{G}_j) \le \mathcal{R}_{m,a}(\{\ell_f : f \in \mathcal{F}\})$. It does not imply a strict reduction, nor that a

learner achieves lower predictive risk. A restrictive statistical constraint may sacrifice predictive fit. The grid contributes a confidence penalty and ζ accounts for estimating the population constraint; neither cost should be omitted. Adding separation or drift-adjustment regularizers to the training objective does not invalidate this uniform statement, because it asserts no equivalence between that objective and an exact constrained empirical-risk minimum.

Deployment shift without a continuity assumption. For losses in $[0, M]$ on all permitted deployment examples, the same event gives, for any f with $\widehat{D}_n(f) \le \varepsilon_j$ and $0 \le \varrho_{\mathrm{TV}} \le 1$,

$$\sup_{\mathbb{Q}:\|\mathbb{P}-\mathbb{Q}\|_{\mathrm{TV}}\le\varrho_{\mathrm{TV}}} R_{\mathbb{Q}}(f) \le \widehat{R}_n(f) + \mathcal{E}_m(\mathcal{F}_{\varepsilon_j+\zeta}) + M\varrho_{\mathrm{TV}}. \tag{35}$$

Indeed, writing $\mathbb{E}_{\mathbb{P}}\ell_f = \int_0^M \mathbb{P}(\ell_f > u)\, du$ gives $|R_{\mathbb{Q}}(f) - R_{\mathbb{P}}(f)| \le M\|\mathbb{P} - \mathbb{Q}\|_{\mathrm{TV}}$ directly. This bound accommodates discontinuous predictors; the following refinement in terms of Wasserstein distance requires additional regularity.

Corollary 1 (Conditional Wasserstein worst-case certificate) *Suppose additionally that $\mathcal{Z}$ is a Polish metric space with metric d, that $\mathbb{P}$ has finite first moment, and that every ℓ_f in each nonempty $\mathcal{G}_j$ is globally K_j-Lipschitz on the space containing all permitted deployment examples, for a finite deterministic K_j. This uses a Wasserstein neighborhood as in distributionally robust analysis (Mohajerin Esfahani & Kuhn, 2018). For $\varrho \ge 0$, let $\mathcal{Q}_\varrho = \{\mathbb{Q} : \mathbb{Q}$ has finite first moment, $W_1(\mathbb{P}, \mathbb{Q}) \le \varrho\}$. On the event of Theorem 1, any f feasible at level ε_j satisfies*

$$\sup_{\mathbb{Q}\in\mathcal{Q}_\varrho} R_{\mathbb{Q}}(f) \le \widehat{R}_n(f) + 2\mathcal{R}_{m,a}(\mathcal{G}_j) + Mt + \frac{Mr}{n} + K_j\varrho. \tag{36}$$

Proof. For every coupling (Z, Z') with marginals $\mathbb{P}, \mathbb{Q}$, Lipschitz continuity gives $R_{\mathbb{Q}}(f) - R_{\mathbb{P}}(f) \le K_j\mathbb{E}d(Z, Z')$. Taking the infimum over couplings yields $R_{\mathbb{Q}}(f) \le R_{\mathbb{P}}(f) + K_j W_1(\mathbb{P}, \mathbb{Q})$. Apply Theorem 1 and take the supremum over $\mathbb{Q} \in \mathcal{Q}_\varrho$. This deterministic argument adds no failure probability. ∎

This corollary requires additional regularity. Hard top-k selection and phase-dependent gates can be discontinuous at lag ties and vanishing Fourier coefficients. For a fixed history, Gaussian NLL grows quadratically in the target, so it is neither bounded nor globally Lipschitz on an unbounded target domain with the usual Euclidean metric. The implementation therefore does not by itself establish the conditions of this corollary or the bounded-loss TV result. Neither result proves that drift adjustment reduces the deployment distance or improves worst-case risk over an unadjusted model.

Geometric mixing and window length. If the raw process obeys $\beta_x(k) \le b_0 e^{-b_1 k}$ for positive constants b_0, b_1, choose, rather than merely lower-bound, the block length as

$$a = q + \max\left\{1, \left\lceil b_1^{-1}\log(2b_0 n/\delta)\right\rceil\right\}. \tag{37}$$

Then $\beta_Z(a) \le \delta/(2n)$, so $c_m \le n\beta_Z(a)/a \le \delta/2$ and $\alpha \ge \delta/2$. When $n \ge 4a$, $n/(4a) \le m \le n/(2a)$, giving

$$m = \Theta\left(\frac{n}{q + 1 + b_1^{-1}\log_+(2b_0 n/\delta)}\right), \qquad \log_+(u) = \max\{0, \log u\}. \tag{38}$$

The concentration contribution is therefore $O(\sqrt{a\log((J+1)/\delta)/n})$, and the remainder contribution is $O(a/n)$. The Rademacher terms require their own class-complexity control; they do not automatically have this rate for an unrestricted neural class. This identifies the dependence and window-length cost without claiming an exact independent-sample rate or verified stationarity of the benchmarks.

Finite-moment losses without changing training. Uniform loss bounds can be replaced by uniform moments. Retain the stationary mixing process, fixed predictor class including learned state, measurability, and block notation of Lemma 1. In this paragraph, replace boundedness by nonnegative finite-valued ℓ_f, d_f satisfying

$$\sup_{f\in\mathcal{F}} \mathbb{E}_{\mathbb{P}}\ell_f^{p_\ell} \le V_\ell, \qquad \sup_{f\in\mathcal{F}} \mathbb{E}_{\mathbb{P}} d_f^{p_d} \le V_d, \qquad p_\ell, p_d > 1. \tag{39}$$

Fix $M, B > 0$ before observing the sample and define $\ell_f^M = \min\{\ell_f, M\}$, $d_f^B = \min\{d_f, B\}$ and $\mathcal{A}^B = \{d_f^B : f \in \mathcal{F}\}$. Truncation is used only in the analysis; $R_{\mathbb{P}}, \widehat{R}_n, D_{\mathbb{P}}, \widehat{D}_n$ below use the original losses.

Proposition 2 (Localization under finite moments) *Fix a finite grid $0 \le \varepsilon_1 < \cdots < \varepsilon_J = \infty$ before training, with $\mathcal{F}_\infty = \mathcal{F}$. For $n = 2am + r$, $m \ge 1$, and $\alpha = \delta - 2(m-1)\beta_Z(a) > 0$, $0 < \delta < 1$, define*

$$\begin{aligned} t &= \sqrt{\frac{\log(2(J+1)/\alpha)}{2m}}, \\ \zeta_B &= 2\mathcal{R}_{m,a}(\mathcal{A}^B) + Bt + \frac{Br}{n} + \frac{V_d}{B^{p_d-1}}, \\ \mathcal{F}_u &= \{f \in \mathcal{F} : D_{\mathbb{P}}(f) \le u\}, \qquad \mathcal{G}_j^M = \{\ell_f^M : f \in \mathcal{F}_{\varepsilon_j+\zeta_B}\}. \end{aligned} \tag{40}$$

With probability at least $1-\delta$, simultaneously for every j and every f satisfying $\widehat{D}_n(f) \le \varepsilon_j$, $f \in \mathcal{F}_{\varepsilon_j+\zeta_B}$ and

$$R_{\mathbb{P}}(f) \le \widehat{R}_n(f) + 2\mathcal{R}_{m,a}(\mathcal{G}_j^M) + Mt + \frac{Mr}{n} + \frac{V_\ell}{M^{p_\ell-1}}. \tag{41}$$

The infinity endpoint provides a full-class fallback when no finite grid level contains the attained empirical discrepancy.

Proof. For a nonnegative random variable U, $s > 1$, and $b > 0$,

$$\mathbb{E}[U - \min\{U, b\}] = \mathbb{E}(U-b)_+ \le \mathbb{E}[U\mathbf{1}_{U>b}] \le b^{1-s}\mathbb{E}U^s. \tag{42}$$

All classes in Equation 40 are deterministic. Apply Lemma 1 jointly to $\mathcal{A}^B$ and the at most J nonempty classes $\mathcal{G}_j^M$, with the allocation for $T = J+1$. On its common event, the tail bound yields, for every f,

$$D_{\mathbb{P}}(f) \le \mathbb{P}d_f^B + V_d/B^{p_d-1} \le \widehat{D}_n(f) + \zeta_B,$$

because the original empirical discrepancy dominates its truncated version. This proves feasibility in the stated population class. The predictive block bound and the same tail inequality then give

$$R_{\mathbb{P}}(f) \le \mathbb{P}\ell_f^M + V_\ell/M^{p_\ell-1} \le \widehat{R}_n(f) + 2\mathcal{R}_{m,a}(\mathcal{G}_j^M) + Mt + Mr/n + V_\ell/M^{p_\ell-1}.$$

Empty local classes have no feasible predictor on this event. Its simultaneous coverage permits sample-dependent selection of the predictor and grid level; M, B themselves are fixed in advance. ■

For the Gaussian objective, $\bar{v} = \max(\sigma^2, \epsilon)$ implies $c_{\mathrm{NLL}} = \frac{1}{2}\log(2\pi\epsilon)$ is a uniform NLL lower bound, so the analysis uses $\ell_f = \mathcal{L}_{\mathrm{pred}} - c_{\mathrm{NLL}} \ge 0$; this fixed shift cancels between empirical and population risk without changing training. Since $\ell_f = (2HC)^{-1}\sum_{h,c}[\log(\bar{v}_{hc}/\epsilon) + (Y_{hc} - \mu_{hc})^2/\bar{v}_{hc}]$, uniform $2p_\ell$ moments of targets and predicted means and uniform p_ℓ moments of $\log(\bar{v}/\epsilon)$ suffice by Jensen's inequality and $\bar{v} \ge \epsilon$, without globally bounding targets, means, or scales.

The statistical-loss moment condition also has a concrete sufficient condition. Sorting minimizes the one-dimensional transport cost, so the original time pairing and unit projections give $\mathcal{L}_{\mathrm{marg}}(\boldsymbol{\mu}, \mathbf{Y}) \le H^{-1}\|\boldsymbol{\mu} - \mathbf{Y}\|_F^2$. Normalized power lies in $[0,1]$, hence $\mathcal{L}_{\mathrm{freq}} \le \log((1+\epsilon)/\epsilon)$. ACF coefficients lie in $[-1,1]$; the stabilized recursion clips higher-order PACF coefficients to $[-0.999, 0.999]$, with the first equal to ACF. Thus $\mathcal{L}_{\mathrm{corr}} \le 2(1+\eta)$ for normalized lag weights, where η is the PACF weight in Equation 7. Fixed loss weights and uniform $2p_d$ moments of targets and means therefore suffice for the statistical-loss moment bound. All moment conditions concern the loss's standardized scale and the entire predictor class, including normalization states; they are not assertions verified for the reported runs.

Proposition 2 bounds risk under $\mathbb{P}$. The total-variation bound in Equation 35 still requires uniform boundedness on deployment examples, and Corollary 1 requires a Lipschitz bound on the original predictive loss. These deployment conditions do not follow from training-distribution moments alone.