Model-Agnostic Shift-Aware Risk-Sensitive Curriculum for Long-Horizon Time-Series Forecasting

Anonymous authors
Paper under double-blind review

Abstract

Long-horizon multivariate forecasting is often brittle under regime changes, rare high-impact windows, and error accumulation. Standard training samples windows uniformly and optimizes mean loss, while existing curricula typically rank windows by difficulty alone and robustness objectives (e.g., CVaR, IRM/REx, GroupDRO) act only after windows have entered the optimization stream. We propose Shift-Aware Curriculum, a *model-agnostic* training wrapper that reallocates gradient budget by coupling (i) self-paced window admission, (ii) shift-aware importance weights over context- or feature-defined environments, and (iii) tail- and environment-robust outer objectives. The wrapper leaves the forecasting backbone unchanged and adds no inference-time cost. At the population level, we formalize the induced target as a trimmed, shift-corrected robust risk. We show that the differentiable quantile gate is an $O(1/\gamma)$ approximation to its hard admitted-set counterpart, quantify the bias introduced by label-adaptive difficulty signals via an explicit adaptive-gap term, and derive a deterministic upper bound on worst-environment risk from the environment-variance penalty. Empirically, on six long-horizon benchmarks (ETTh1/2, ETTm1/2, Weather, Electricity) and four backbones (RLinear, DLinear, RMLP, iTransformer), Shift-Aware Curriculum lowers MSE in 82 of 96 backbone–dataset–horizon cells, with 65 cells improving by more than 1%, and yields positive average gains in every backbone–horizon aggregate. On a scoped robustness battery (ETTh1 with DLinear), Shift-Aware Curriculum reduces mean MSE by 5.1–9.0% across temporal shift levels and reduces worst-environment MSE by up to 30% in the hardest stress setting.

1 Introduction

Context and motivation Multivariate time-series forecasting is central to applications in energy, transportation, environmental monitoring, and network operations (Lim & Zohren, 2021). In such settings, models are often deployed under long horizons (e.g., 96-720 steps ahead) and over extended periods, where seasonal structure, calendar effects, and rare events conspire to produce complex distribution shifts (Liu et al., 2022; Lim & Zohren, 2021). Even when average error is modest, failures concentrated on rare but high-impact windows (e.g., holiday peaks or severe weather) can be unacceptable in practice.

While the last few years have seen rapid progress in forecasting backbones—including efficient Transformers, decomposition-based architectures, patch-token designs, and strong linear baselines (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022; Nie et al., 2022; Wu et al., 2022; Zeng et al., 2023; Liu et al., 2023)—the *training protocols* used for these models are still largely uniform: windows are drawn with equal probability, losses are averaged, and convergence is judged by the mean error on a held-out validation split. This treats all windows and regimes as equally important, implicitly optimizing a risk-neutral objective. In deployment, however, practitioners often care far more about performance on specific subsets of the data: peak load, rare anomalies, or distribution regimes that are under-represented in the training history.

Limitations of existing approaches. Curriculum learning offers a natural way to bias training toward informative windows by ordering examples from "easy" to "hard". However, most curricula in the time-series literature focus on difficulty in isolation: they define hard windows via large residuals or high-frequency content,

but do not explicitly account for distribution shift or tail risk. Conversely, robust optimization methods such as Conditional Value-at-Risk (CVaR) and invariant risk minimization (IRM/REx) emphasize worst-case environments or out-of-distribution (OOD) generalization, but treat all windows inside an environment as equally important. Neither line of work directly answers the question: *which specific windows should a forecaster spend more gradient budget on, given both deployment shift and tail risk?*

Our perspective. We argue that training-time policies deserve the same algorithmic attention as architectures. Rather than proposing yet another backbone, we treat state-of-the-art forecasters as fixed and design a *shift-aware, risk-sensitive curriculum* that can be wrapped around them. The goal is to turn off-the-shelf models into more robust, deployment-ready forecasters by learning *where and when* to focus training, without increasing inference-time complexity.

1.1 Contributions

- We formulate a *training-policy problem* for long-horizon forecasting in which example admission, deployment-shift correction, and robustness penalties are treated as one coupled design object rather than as loosely attached heuristics. Concretely, we define a differentiable admitted-set policy that combines multi-signal difficulty, environment-aware importance weights, and a self-paced quantile gate.
- We instantiate this formulation as a *model-agnostic wrapper* around fixed forecasting backbones. The resulting algorithm changes neither inference-time computation nor backbone architecture; it only changes which windows are sampled, when they enter the curriculum, and how tail/environment risk is emphasized during training.
- We provide *analysis for the coupled objective*. Rather than citing the ingredients in isolation, we formalize the hard trimmed deployment target induced by the wrapper, prove an $O(1/\gamma)$ soft-to-hard approximation for the differentiable gate, isolate the additional bias caused by label-adaptive difficulty through an explicit adaptive-gap term, and show that the environment-variance penalty yields a deterministic upper bound on worst-environment risk (Shimodaira, 2000; Sugiyama et al., 2007; Cortes et al., 2010; Rockafellar et al., 2000; Krueger et al., 2021).
- We evaluate the wrapper on six long-horizon benchmarks and four heterogeneous backbones. Shift-Aware Curriculum lowers MSE in 82 of 96 backbone–dataset–horizon cells and yields positive average gains in every backbone–horizon aggregate. We additionally report a scoped robustness battery on ETTh1 with DLinear to test the tail/OOD claims.

2 Related Work

2.1 Forecasting Backbones

Recent work in long-horizon time-series forecasting has shown that strong performance can come from very different architectural biases. Efficient sparse-attention forecasters such as Informer (Zhou et al., 2021), decomposition-based Transformers such as Autoformer and FEDformer (Wu et al., 2021; Zhou et al., 2022), patch-token designs such as PatchTST (Nie et al., 2022), and general-purpose temporal backbones such as TimesNet (Wu et al., 2022) all report strong results on standard long-horizon benchmarks. Decomposition-based linear baselines such as DLinear (Zeng et al., 2023) are competitive because many benchmark datasets are dominated by strong seasonal and trend structure, so a simple seasonal–trend split with a lightweight prediction head can already capture a large fraction of the signal. By contrast, Crossformer and iTransformer (Zhang & Yan, 2023; Liu et al., 2023) stress richer cross-variable modeling, while Non-stationary Transformers (Liu et al., 2022) explicitly target regime instability. These families jointly reinforce a useful point for our setting: architectural capacity and training policy are distinct levers, and empirical gains should not be attributed to one while implicitly changing the other.

Our experimental design therefore spans heterogeneous backbones on purpose. We evaluate both widely used external baselines (DLinear and iTransformer) and two decomposition-family implementation variants

(RLinear and RMLP) to test whether the proposed training policy transfers across linear, shallow non-linear, and attention-based predictors under a common protocol. This makes the central comparison sharper than a standard "new backbone vs. old backbone" narrative: the question is whether a single training-time wrapper improves multiple backbone families without architectural modification.

2.2 Curriculum Learning and Self-Paced Optimization

Curriculum learning (Bengio et al., 2009) and self-paced learning (Kumar et al., 2010) both advocate staged optimization in which easier examples dominate earlier updates and harder examples enter later. Their shared optimization argument is that lower-variance examples can stabilize early learning, improve convergence, and reduce brittle behavior in non-convex training. That logic is directly relevant to long-horizon forecasting, where training can be destabilized by rare windows with abrupt regime breaks, sharp transitions, or compounding forecast error.

However, classical curricula are usually organized around a single hardness score and are mostly agnostic to deployment prevalence. They answer "which examples are hard for optimization?" but not "which hard examples matter most under deployment shift?" In addition, classical curricula often decouple example selection from robustness design: the curriculum decides which examples are admitted, while any shift correction or tail-risk objective is applied separately, if at all. Our method differs in three ways. First, it uses a differentiable quantile gate rather than a hard threshold or top-k selection, making the admission policy smoother and easier to optimize jointly with the model. Second, it combines multiple difficulty signals rather than relying on a single proxy. Third, and most importantly, admission is explicitly coupled to shift-aware weighting and a tail-sensitive outer objective, so the curriculum is not merely "easy-to-hard" but it is calibrated toward deployment-relevant windows.

2.3 Covariate Shift and Importance Weighting

Importance weighting under covariate shift (Shimodaira, 2000; Sugiyama et al., 2007; Kanamori et al., 2009) provides the statistical basis for rebalancing training mass toward environments that are more prevalent at deployment than in the historical training distribution. In standard covariate-shift settings, a correctly specified density ratio can be used to target deployment risk without changing the predictive model class. This perspective is valuable for forecasting because temporal data are rarely stationary: deployment may over-emphasize specific seasons, calendar buckets, or anomalous regimes that are under-represented in the earlier training history.

At the same time, importance weighting alone is incomplete for our objective. It rebalances the expected contribution of windows, but it does not specify a staged optimization policy, and it does not directly prioritize windows that are high-loss or pedagogically informative for the current model. In our formulation, shift-aware weights are therefore one component of the training policy rather than the whole policy: they shape mass inside the admitted set, while the curriculum gate determines when windows enter and the outer objective determines how heavily tail behavior is penalized once those windows are sampled.

2.4 Tail-Risk Objectives and Invariance

Risk-sensitive objectives such as CVaR (Rockafellar & Uryasev, 2002; Rockafellar et al., 2000), group-wise distributionally robust optimization (Sagawa et al., 2019), and environment-robust methods such as IRM/REx (Arjovsky et al., 2019; Krueger et al., 2021) address a different but complementary question: once examples are observed, which losses should dominate the objective? CVaR emphasizes the upper tail of the loss distribution, making it natural when rare but severe forecast failures dominate operational cost. IRM and REx instead regularize predictors toward flatter cross-environment risk profiles, which is attractive when deployment robustness depends on not overfitting a small subset of regimes.

These tools are highly relevant, but by themselves they do not specify a fine-grained window-level training policy. CVaR identifies bad outcomes, yet it does not decide when those windows should enter the curriculum. IRM/REx operate at the environment level, yet they do not prioritize specific windows within an environment once that environment has been identified. Our contribution is therefore not a new backbone and not a

stand-alone OOD objective. It is a training-policy coupling across three levels: the *window level* (via the self-paced admission gate), the *environment level* (via shift-aware weighting and REx-style balancing), and the *tail-risk level* (via CVaR-style emphasis).

3 Problem Setup and Notation

Let $\{x_t\}_{t=1}^T$ be a D-variate time series, where $x_t \in \mathbb{R}^D$ denotes the vector of channel values at time t. We consider standard sliding-window forecasting: given a lookback window of length L,

$$X_i = (x_{t_i-L+1}, \ldots, x_{t_i}) \in \mathbb{R}^{L\times D},$$

the goal is to predict a horizon-H future segment

$$Y_i = (x_{t_i+1}, \ldots, x_{t_i+H}) \in \mathbb{R}^{H\times D}.$$

We index windows by $i = 1, \ldots, N$ and write (X_i, Y_i) for each training example. A forecasting backbone is a parametric function f_θ that maps X_i to a prediction $\hat{Y}_i = f_\theta(X_i)$, trained by minimizing a loss such as mean squared error (MSE),

$$\mathcal{L}_{\text{MSE}}(\theta) = \frac{1}{NHD}\sum_{i=1}^{N} \|\hat{Y}_i - Y_i\|_F^2.$$

We will often use the per-window loss $\ell_\theta(X_i, Y_i) = \frac{1}{HD}\|\hat{Y}_i - Y_i\|_F^2$, so that $\mathcal{L}_{\text{MSE}}(\theta) = \frac{1}{N}\sum_i \ell_\theta(X_i, Y_i)$.

We assume that windows can be grouped into environments $e \in \mathcal{E}$, representing, for example, structured context bins (e.g., hour-of-day, day-of-week, month) or clusters in feature space. Let e_i denote the environment label of window i and $\mathcal{I}_e = \{i : e_i = e\}$ the set of indices in environment e. The environment-level risk of θ is

$$R_e(\theta) = \frac{1}{|\mathcal{I}_e|}\sum_{i\in\mathcal{I}_e} \ell_\theta(X_i, Y_i),$$

where ℓ_θ is a scalar loss (e.g., per-window MSE).

Our goal is to design a training policy that, instead of optimizing the plain average risk $\frac{1}{N}\sum_i \ell_\theta(X_i, Y_i)$, focuses on (i) windows that are hard for the current model (according to multiple signals), (ii) environments that are shifted or under-represented relative to deployment, and (iii) tail regions of the loss distribution (via CVaR). We formalize this via a differentiable gate that assigns per-window probabilities proportional to a mixture of difficulty and shift signals, and we optimize a CVaR-regularized objective over environments.

4 Signals, Shift Weights, and Environments

Window-level difficulty signals. We use three complementary proxies for how hard a window is for the current forecaster:

1. **Residual magnitude** s_i^{res}: the current per-window prediction error, e.g., $\ell_\theta(X_i, Y_i)$.
2. **Residual irregularity** s_i^{freq}: a frequency-domain measure of the residual sequence (e.g., high-frequency energy or spectral entropy) that captures oscillatory or rapidly changing errors.
3. **Probe-loss / persistence** s_i^{loss}: a smoothed loss signal (e.g., from an exponential moving average (EMA) teacher or a lightweight probe predictor trained during warm-up) that highlights windows that remain hard over time and reduces the noise of instantaneous residuals.

Each signal emphasizes a different failure mode: large amplitude errors, structurally complex errors, and persistent underfitting.

We standardize the three signals on the training split (z-score normalization) to obtain $r_i, f_i, u_i \in \mathbb{R}$, and mix them with learnable coefficients $\boldsymbol{\alpha} = (\alpha_{\text{res}}, \alpha_{\text{freq}}, \alpha_{\text{loss}})$.

Shift-aware importance weights. To target a deployment distribution that differs from the training history, we associate each window i with an importance weight w_i that approximates a density ratio between deployment and training. When environments are discrete context bins, w can be estimated by relative frequencies; when environments are learned clusters, standard density-ratio estimators or source-vs-target classifiers can be used.

Formally, if we only observe an environment label e_i, we use the coarse environment ratio

$$w_i = w_{e_i} = \frac{p_{\text{deploy}}(e_i)}{p_{\text{train}}(e_i)},$$

optionally normalized to have mean 1 across training windows. More refined ratios (e.g., $p_{\text{deploy}}(X)/p_{\text{train}}(X)$) can be plugged in without changing the wrapper.

Environment construction. We form environments $e \in \mathcal{E}$ using:

- **Structured context bins** (hour-of-day, day-of-week, month),
- **Feature clusters** obtained by clustering representations (e.g., encoder outputs or raw windows) into K groups, and
- **Hybrid schemes** that combine context bins and clusters.

Environment assignments are held fixed within a curriculum chunk. If representation-based clusters are used, they are refreshed only at chunk boundaries so that environment identities do not drift within a mini-batch.

5 Differentiable Curriculum Gate

Let r_i, f_i, u_i denote the standardized residual, spectral, and probe-loss signals from Section 4. We combine them into a single difficulty score

$$s_i(\boldsymbol{\alpha}) = \alpha_{\text{res}} r_i + \alpha_{\text{freq}} f_i + \alpha_{\text{loss}} u_i,$$

where $\boldsymbol{\alpha} \in \mathbb{R}^3$ are learnable parameters. Intuitively, windows with higher s_i are considered harder.

Soft quantile gating (self-paced). We wish to allocate more training probability to a moving quantile of windows in a differentiable manner. To this end, we define a *soft quantile gate*. For a target fraction $p \in (0, 1)$, let q_p be the p-quantile of $\{s_i(\boldsymbol{\alpha})\}_{i=1}^N$, and define

$$m_i = \sigma\big(\gamma(q_p - s_i(\boldsymbol{\alpha}))\big),$$

where σ is the logistic sigmoid and $\gamma > 0$ is a temperature parameter. When $s_i(\boldsymbol{\alpha}) \ll q_p$, $m_i \approx 1$; when $s_i(\boldsymbol{\alpha}) \gg q_p$, $m_i \approx 0$. As $\gamma \to \infty$, this approximates a hard quantile / bottom-p (easiest-p) selection.

With this sign convention, the gate softly *admits windows whose difficulty is below the current quantile threshold*, i.e., the easiest p-fraction at a given curriculum step. Increasing p therefore implements a standard *easy-to-hard* self-paced curriculum: early training focuses on stable/easy windows; later training progressively includes harder windows (including rare, high-loss regimes). The complementary choice $m_i = \sigma(\gamma(s_i(\boldsymbol{\alpha}) - q_p))$ would prioritize the hardest p-fraction; we do not use this variant in the reported experiments.

In practice, we estimate q_p within each curriculum chunk using a running buffer of scores and stop gradients through the quantile computation.

To avoid degenerate zeros, we define a smoothed gate

$$\widetilde{m}_i = \varepsilon + (1 - \varepsilon) m_i,$$

with a small $\varepsilon \in (0, 1)$ (e.g., $\varepsilon = 0.05$), ensuring that no window is completely discarded.

Sampling probabilities. We combine the gate with shift weights to obtain per-window sampling probabilities

$$\pi_i \propto w_i \widetilde{m}_i,$$

normalized so that $\sum_i \pi_i = 1$. We then sample mini-batches according to $\{\pi_i\}$. When estimating Eq. (1) with SGD, we use a normalized mini-batch average (equivalently, normalize $\omega_i^{(c)}$ within the batch); we do not multiply losses by $w_i \widetilde{m}_i$ a second time. This concentrates gradient mass on windows that are simultaneously (i) *admitted by the current self-paced gate* (easy early; progressively harder later), and (ii) *upweighted by deployment shift* via w_i. In later curriculum chunks (when p is large) the admitted set includes the hard/tail windows, and the shift-aware weighting steers optimization toward deployment-relevant regimes.

Schedule over competence. As training progresses, the target fraction p is increased from a small initial value (e.g., $p_0 = 0.1$) to 1.0, analogous to moving from easy to hard examples. At curriculum step $c \in \{1, \ldots, C\}$, we may set $p_c = c/C$, so that early steps focus on the easiest 10–20 % windows, while later steps include increasingly harder windows, eventually covering the full dataset.

6 Risk-Sensitive Outer Objective

Let w_i be the (normalized) importance weight of window i and $\widetilde{m}_i$ the gate output. The effective weight of window i at curriculum step c is

$$\omega_i^{(c)} = w_i \widetilde{m}_i^{(c)}.$$

We define the risk at curriculum step c as

$$R^{(c)}(\theta) = \frac{\sum_i \omega_i^{(c)} \ell_\theta(X_i, Y_i)}{\sum_i \omega_i^{(c)}},$$

and the environment-level risks as

$$R_e^{(c)}(\theta) = \frac{\sum_{i \in \mathcal{I}_e} \omega_i^{(c)} \ell_\theta(X_i, Y_i)}{\sum_{i \in \mathcal{I}_e} \omega_i^{(c)}}.$$

CVaR-style tail emphasis. To emphasize tail risk, we define a CVaR-style penalty over environments. Let $\tau \in (0, 1)$ be a tail percentile (e.g., $\tau = 0.9$). For each environment e, we estimate a (possibly weighted) τ-quantile $q_\tau^{(e)}$ of the per-window losses $\{\ell_\theta(X_i, Y_i)\}_{i \in \mathcal{I}_e}$. We then define the tail set $\mathcal{J}_e = \{i \in \mathcal{I}_e : \ell_\theta(X_i, Y_i) \geq q_\tau^{(e)}\}$ and compute a weighted tail average aligned with the curriculum weights:

$$\mathrm{CVaR}_\tau^{(e)}(\theta) = \frac{\sum_{i \in \mathcal{J}_e} \omega_i^{(c)} \, \ell_\theta(X_i, Y_i)}{\sum_{i \in \mathcal{J}_e} \omega_i^{(c)}}.$$

This focuses the tail penalty on high-loss windows that are also emphasized by the shift-aware curriculum, and reduces the influence of low-weight regimes on the tail statistic.

We aggregate across environments as

$$\mathrm{CVaR}_\tau(\theta) = \sum_{e \in \mathcal{E}} \mu_e \, \mathrm{CVaR}_\tau^{(e)}(\theta),$$

where μ_e is a normalized weight for environment e (e.g., proportional to $p_{\mathrm{deploy}}(e)$).

Environment-level invariance. To encourage invariance across environments, we consider an REx-style regularizer. Let $\bar{R}^{(c)}(\theta) = \frac{1}{|\mathcal{E}|} \sum_e R_e^{(c)}(\theta)$ denote the mean environment risk at step c. A REx-style penalty (Krueger et al., 2021) is

$$\mathcal{R}_{\mathrm{REx}}^{(c)}(\theta) = \frac{1}{|\mathcal{E}|} \sum_e \big(R_e^{(c)}(\theta) - \bar{R}^{(c)}(\theta)\big)^2,$$

which discourages large variance in risks across environments.

Combined objective. The overall objective at curriculum step c is

$$\mathcal{L}_c(\theta) = R^{(c)}(\theta) + \lambda^{(c)}_{\mathrm{CVaR}}\,\mathrm{CVaR}_\tau(\theta) + \lambda^{(c)}_{\mathrm{inv}}\,\mathcal{R}^{(c)}_{\mathrm{REx}}(\theta), \tag{1}$$

where $\lambda^{(c)}_{\mathrm{CVaR}}$ and $\lambda^{(c)}_{\mathrm{inv}}$ are step-dependent regularization strengths. In practice, we use monotone schedules that start small for stability and increase with competence so that the strongest tail and environment regularization is applied only after the base predictor has reached a stable regime.

7 Training Schedule and Optimization

Each experiment proceeds in three stages.

1. **Warm-up (full data).** Optimize $\frac{1}{N}\sum_i \ell_\theta(X_i, Y_i)$ on uniformly sampled windows for a few epochs to stabilize features and obtain the initial probe losses used by the curriculum.
2. **Chunked curriculum.** For c=1:C: at the start of each chunk, optionally refresh the loss-based signal and the shift weights from the current EMA model (or keep the warm-up estimates fixed in the lightweight variant), then update $\boldsymbol{\alpha}$ and θ using the soft gate and the risk-sensitive objective $\mathcal{L}_c(\theta)$ with a gradually increasing target fraction p_c.
3. **Polish (full data).** Turn off robustness penalties and gates; train briefly on full data under the standard ERM objective to recover uniform-sampling calibration before checkpoint selection. This stage is used as an empirical stabilizer, not as a stand-alone theoretical debiasing guarantee.

Algorithm 1 Shift-Aware Curriculum (model-agnostic wrapper)

Require: Windows $\{(X_i, Y_i)\}_{i=1}^N$, horizon H, lookback L, base model f_θ, environments e_i, shift weights w_i, curriculum fractions $\{p_c\}_{c=1}^C$, gate temperature γ, gate floor ε, tail level τ, and schedules $\{\lambda^{(c)}_{\mathrm{CVaR}}\}$, $\{\lambda^{(c)}_{\mathrm{inv}}\}$

Warm-up: train θ for T_0 epochs on full data
Compute initial signals $\{s_i^{\mathrm{res}}, s_i^{\mathrm{freq}}, s_i^{\mathrm{loss}}\}$; normalize to $\{r_i, f_i, u_i\}$
Estimate initial shift weights w_i via importance-weighting; assign environments e_i
Initialize gate parameters $\boldsymbol{\alpha}$ (e.g., uniform)
for $c = 1$ to C **do**
    Optionally refresh u_i, w_i, and per-environment statistics at the start of chunk c using the current EMA model; keep context-based assignments fixed
    $p_c \leftarrow c/C$; $q_c \leftarrow \mathrm{Quantile}_{p_c}\{s_j(\boldsymbol{\alpha})\}$
    $m_i^{(c)} \leftarrow \sigma(\gamma\,(q_c - s_i(\boldsymbol{\alpha})))$; $\widetilde{m}_i^{(c)} \leftarrow \varepsilon + (1-\varepsilon)m_i^{(c)}$
    Mini-batch sampling: sample windows with probabilities $\propto w_i\,\widetilde{m}_i^{(c)}$
    Outer loss: compute per-env risks $R_e^{(c)}(\theta)$ and $\mathcal{L}_c(\theta)$ via Eq. (1)
    Update θ: $\theta \leftarrow \mathsf{OptimizerStep}(\theta, \nabla_\theta \mathcal{L}_c(\theta))$; update EMA $\bar{\theta} \leftarrow \beta\bar{\theta} + (1-\beta)\theta$
    Update gate: $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - \eta_\alpha \nabla_{\boldsymbol{\alpha}} \Big(\sum_i \widetilde{m}_i^{(c)}\, \ell_\theta(X_i, Y_i) \Big)$
    Schedule: increase $\lambda^{(c)}_{\mathrm{CVaR}}$ and $\lambda^{(c)}_{\mathrm{inv}}$ with c; optionally decrease lr
end for
Polish: Set λ_{CVaR}=λ_{inv}=0; train briefly on full data under uniform ERM; keep best checkpoint by validation risk
Inference: For any test sequence, windowize, apply base model with $\bar{\theta}$ (EMA), invert normalization, compute MSE/MAE
return $\theta^\star$ (best checkpoint), $\bar{\theta}$, inference routine

8 Theoretical Properties

This section specifies the scope of the theoretical analysis and the guarantees that can be derived for the proposed curriculum. The goal is not to establish a distribution-free generalization theorem for long-horizon forecasting, nor to claim that the label-adaptive implementation provides an unbiased estimate of deployment risk. Instead, the analysis identifies the population-level objective that the curriculum approximates and

characterizes the discrepancies between this ideal target and the implemented training objective. Framing the method in this way allows the training procedure to be viewed as a well-defined surrogate optimization problem rather than an unconstrained heuristic.

The analysis yields four properties specific to the proposed formulation. First, the easy-to-hard progression induced by the gate corresponds to a continuation path over admitted training windows, rather than a sequence of arbitrary reweightings. Second, the differentiable gate provides a smooth approximation to a hard trimmed objective, with an explicit approximation error controlled by the temperature parameter. Third, when label- or model-dependent difficulty signals are incorporated, the resulting discrepancy between the surrogate and the population objective appears as a single adaptive-gap term. Fourth, the environment-variance regularization term yields a deterministic upper bound on worst-environment risk.

The covariate-shift components of the analysis follow the classical framework of importance-weighted risk estimation and its finite-sample properties (Shimodaira, 2000; Sugiyama et al., 2007; Cortes et al., 2010). The continuation perspective relates the curriculum schedule to homotopy methods studied in graduated optimization (Hazan et al., 2016). The resulting propositions therefore describe what quantities the training objective controls, how approximation errors arise, and why the robust-training interpretation remains meaningful even though the overall learning problem is non-convex.

Population-level notation. Let (X, Y) denote a random window/target pair drawn from the training distribution. Let $w(X)$ be the (possibly estimated) importance weight targeting deployment, and let $s(X)$ be the feature-based difficulty score. For a target fraction $p \in (0, 1)$, let q_p be the p-quantile of $S = s(X)$.

We define a *hard* admitted-set gate and its *soft* (differentiable) relaxation as

$$h_p(X) = \varepsilon + (1 - \varepsilon)\, \mathbb{1}\{s(X) \le q_p\}, \qquad g_{p,\gamma}(X) = \varepsilon + (1 - \varepsilon)\, \sigma\big(\gamma(q_p - s(X))\big),$$

where σ is the logistic sigmoid and $\gamma > 0$ is the gate temperature. The corresponding normalized risks are

$$R_p^{\mathrm{hard}}(\theta) = \frac{\mathbb{E}[w(X)\, h_p(X)\, \ell_\theta(X, Y)]}{\mathbb{E}[w(X)\, h_p(X)]}, \qquad R_{p,\gamma}^{\mathrm{soft}}(\theta) = \frac{\mathbb{E}[w(X)\, g_{p,\gamma}(X)\, \ell_\theta(X, Y)]}{\mathbb{E}[w(X)\, g_{p,\gamma}(X)]}.$$

These quantities describe the population targets that the curriculum approximates at a fixed curriculum fraction p.

Proposition 1 (Monotone continuation of the curriculum)**.** *Fix $\gamma > 0$. If $0 < p_1 \le p_2 \le 1$, then $g_{p_1,\gamma}(X) \le g_{p_2,\gamma}(X)$ for every X. Consequently, the (unnormalized) gate weight assigned to any window is nondecreasing as the curriculum fraction increases (the normalized sampling probability may still change because the normalizer depends on p).*

Proof sketch. The quantile map $p \mapsto q_p$ is monotone nondecreasing, and the sigmoid is monotone nondecreasing in its argument. Therefore $q_{p_1} \le q_{p_2}$ implies $\sigma(\gamma(q_{p_1} - s(X))) \le \sigma(\gamma(q_{p_2} - s(X)))$ pointwise, and the same ordering is preserved after adding the floor ε. This gives a genuine continuation path over admitted sets rather than a sequence of unrelated reweightings (Hazan et al., 2016).

Proposition 2 (Soft-to-hard approximation error)**.** *Assume that, for a fixed θ, (i) $0 \le \ell_\theta(X, Y) \le B$, (ii) $0 \le w(X) \le W$, (iii) the score $S = s(X)$ has a density bounded by M in a neighborhood of q_p, and (iv) both normalizers satisfy*

$$\mathbb{E}[w(X) h_p(X)] \ge z_0 > 0, \qquad \mathbb{E}[w(X) g_{p,\gamma}(X)] \ge z_0 > 0.$$

Then

$$\big|R_{p,\gamma}^{\mathrm{soft}}(\theta) - R_p^{\mathrm{hard}}(\theta)\big| \le \frac{2BW(1 - \varepsilon)M \log 2}{\gamma} \left(\frac{1}{z_0} + \frac{W}{z_0^2}\right).$$

In particular, for fixed p, the smooth gate is an $O(1/\gamma)$ approximation to the hard admitted-set objective.

Proof sketch. Let $a = \mathbb{E}[w g_{p,\gamma} \ell_\theta]$, $b = \mathbb{E}[w g_{p,\gamma}]$, $c = \mathbb{E}[w h_p \ell_\theta]$, and $d = \mathbb{E}[w h_p]$. A standard ratio decomposition gives

$$\left|\frac{a}{b} - \frac{c}{d}\right| \le \frac{|a - c|}{z_0} + \frac{BW\,|b - d|}{z_0^2}.$$

Because $|a-c| \le BW\,\mathbb{E}|g_{p,\gamma} - h_p|$ and $|b-d| \le W\,\mathbb{E}|g_{p,\gamma} - h_p|$, it remains to bound the gate mismatch. The only discrepancy occurs near the threshold q_p, and the logistic tails integrate to $\log 2/\gamma$ on each side. Using the density bound on S yields

$$\mathbb{E}|g_{p,\gamma}(X) - h_p(X)| \le \frac{2(1-\varepsilon)M \log 2}{\gamma},$$

which gives the stated result. This turns the differentiable gate into a controlled surrogate, rather than an unquantified heuristic smoothing.

Proposition 3 (Adaptive-gap decomposition for label-dependent gates)**.** *Let the implemented gate be*

$$\widetilde{m}(X,Y) = g_{p,\gamma}(X) + \Delta(X,Y), \qquad 0 \le \widetilde{m}(X,Y) \le 1,$$

where Δ collects the label-dependent or model-dependent part of the difficulty signal. Define

$$\rho_p = \mathbb{E}[w(X)\,|\Delta(X,Y)|]\,,$$

and assume the soft and adaptive normalizers are both at least $z_0 > 0$. If

$$R_p^{\mathrm{adapt}}(\theta) = \frac{\mathbb{E}[w(X)\,\widetilde{m}(X,Y)\,\ell_\theta(X,Y)]}{\mathbb{E}[w(X)\,\widetilde{m}(X,Y)]},$$

then

$$\left|R_p^{\mathrm{adapt}}(\theta) - R_{p,\gamma}^{\mathrm{soft}}(\theta)\right| \le \frac{2B\rho_p}{z_0}.$$

Thus the label-adaptive component introduces a single explicit bias term ρ_p; it does not make the target mathematically opaque.

Proof sketch. Write the two normalized risks as a/b and c/d with $a - c = \mathbb{E}[w\Delta\ell_\theta]$ and $b - d = \mathbb{E}[w\Delta]$. Then $|a-c| \le B\rho_p$ and $|b-d| \le \rho_p$. Using

$$\left|\frac{a}{b} - \frac{c}{d}\right| \le \frac{|a-c|}{z_0} + \frac{c}{d}\,\frac{|b-d|}{z_0},$$

and the fact that every normalized risk is at most B, yields the bound. Proposition 3 is the key reason we describe the full objective as a controlled robustness-oriented surrogate instead of dismissing it as theoretically uninterpretable.

Theorem 1 (Explicit worst-environment control from the REx term)**.** *For any fixed p and γ, let $R_1, \ldots, R_E$ denote the environment risks $R_{e,p,\gamma}^{\mathrm{soft}}(\theta)$, let $\bar{R} = E^{-1}\sum_{e=1}^{E} R_e$, and let*

$$V = \frac{1}{E}\sum_{e=1}^{E}(R_e - \bar{R})^2.$$

Then

$$\max_{1\le e\le E} R_e \le \bar{R} + \sqrt{EV} \le \bar{R} + \lambda_{\mathrm{inv}} V + \frac{E}{4\lambda_{\mathrm{inv}}} \qquad \text{for every } \lambda_{\mathrm{inv}} > 0.$$

Therefore, minimizing the mean environment risk together with the REx penalty directly upper-bounds the worst-environment risk up to an explicit slack term. This is a deterministic surrogate guarantee for the specific objective in Eq. (1); it does not rely on asymptotics.

Proof sketch. The first inequality is immediate from

$$\max_e (R_e - \bar{R}) \le \sqrt{\sum_{e=1}^{E}(R_e - \bar{R})^2} = \sqrt{EV},$$

which is a one-line consequence of Cauchy-Schwarz. The second inequality applies $2ab \le a^2 + b^2$ with $a = \sqrt{\lambda_{\mathrm{inv}} V}$ and $b = \sqrt{E/(4\lambda_{\mathrm{inv}})}$. This gives an explicit analytical bridge between the REx-style variance penalty and worst-environment control, complementing the empirical motivations in REx and GroupDRO (Krueger et al., 2021; Sagawa et al., 2019).

Table 1: Default configuration of the proposed training wrapper. Unless otherwise stated, all experiments use this specification.

Component	Default specification
Admission direction	Windows are admitted using an easiest-first gate $m_i = \sigma(\gamma(q_p - s_i))$. A hardest-first variant is defined for completeness but is not used in the main experiments.
Curriculum schedule	Competence increases linearly as $p_c = c/C$, so the admitted set gradually expands from easier to more difficult windows.
Difficulty signals	Difficulty is computed from three sources—residual magnitude, spectral features, and probe-model loss—combined through learnable mixing coefficients $\boldsymbol{\alpha}$.
Signal scaling	Each signal is standardized via per-split z-score normalization before mixing to ensure comparable scale.
Gate smoothing	A small floor $\varepsilon = 0.05$ is applied to the gate to prevent windows from receiving zero probability.
Sampling rule	Mini-batches are sampled with probability proportional to $w_i \widetilde{m}_i$, followed by normalization within the batch.
Loss weighting convention	Because sampling already reflects $w_i \widetilde{m}_i$, the loss is not multiplied by these weights again during optimization.
Environment stability	Environment assignments remain fixed within each curriculum chunk; context-defined environments remain constant throughout training.
Training stages	Training proceeds in three phases: an initial warm-up on the full dataset, a chunked curriculum stage, and a short full-data fine-tuning phase.
Inference	The wrapper affects training only; no architectural changes or additional computation are introduced at inference.

Corollary 1 (What the training objective is actually controlling)**.** *Under the assumptions of Propositions 2 and 3, the mean-risk component of the implemented curriculum objective differs from the hard trimmed deployment risk by at most*

$$\left|R_p^{\mathrm{adapt}}(\theta) - R_p^{\mathrm{hard}}(\theta)\right| \leq \frac{2BW(1-\varepsilon)M\log 2}{\gamma}\left(\frac{1}{z_0} + \frac{W}{z_0^2}\right) + \frac{2B\rho_p}{z_0}.$$

Combined with Theorem 1, the optimization in Eq. (1) can be read as minimizing a trimmed deployment risk plus explicit smoothing, adaptation, and worst-environment slack terms*.*

Optimization implication. Together, Proposition 1 shows that the schedule over p defines a continuation path; Proposition 2 quantifies the soft-to-hard approximation error of the differentiable gate; Proposition 3 isolates the additional bias introduced by label-adaptive difficulty signals; and Theorem 1 provides a deterministic worst-environment bound from the REx term.

9 Complexity and Implementation Notes

Let N be the number of windows, D the number of channels, and L the lookback length. Per training epoch, the dominant cost remains the backbone forward/backward passes, scaling as $O(N \cdot \mathrm{cost}(f_\theta))$. The wrapper adds: (i) $O(N)$ work to compute and normalize difficulty signals, (ii) $O(N)$ work to estimate the relevant quantile and evaluate the gate, and (iii) $O(N)$ work to maintain shift-aware weights and environment-level summaries. In practice, these additions are small relative to the backbone cost, especially for attention-based models.

The remaining hyperparameters split into two categories. *Method-defining* settings (gate direction, competence schedule, smoothing convention, and weighting semantics) are fixed by the specification above. *Backbone-specific* settings (optimizer family, learning rate, batch size, scheduler, and early-stopping patience) are inherited from the underlying backbone configuration and then reused unchanged in the paired baseline and wrapper runs. This separation is deliberate. It ensures that observed differences are attributable to the training policy rather than to hidden retuning of the architecture.

We implement the method in PyTorch as a thin curriculum module around existing backbone implementations. The module maintains running estimates of signals, gates, and environment statistics, and exposes only two interfaces to the training loop: sampling probabilities and the outer objective.

10 Experimental Evaluation

We evaluate the proposed *model-agnostic* Shift-Aware Curriculum wrapper as a drop-in training policy for strong multivariate time-series forecasters. Our focus is not on inventing yet another backbone, but on showing that a single curriculum can systematically improve heterogeneous architectures under realistic long-horizon settings.

10.1 Backbones, Datasets, and Metrics

Backbones. We plug Shift-Aware Curriculum into four representative forecasting models covering linear, MLP, and attention families: **DLinear** (Zeng et al., 2023) and the closely related **RLinear** implementation variant used in our evaluation suite, both of which decompose each channel into trend and seasonal components; **RMLP**, a lightweight decomposition-family non-linear variant in the same evaluation suite; and **iTransformer** (Liu et al., 2023), a recent channel-wise Transformer with competitive performance on long-horizon benchmarks. We treat all four as fixed backbones and use RLinear/RMLP explicitly as implementation-level stress tests of architectural diversity rather than as separate novelty claims, which allows us to isolate the effect of the curriculum from architectural innovations.

Datasets. Following standard practice in long-horizon forecasting benchmarks popularized by Informer, Autoformer, FEDformer, PatchTST, and related work (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022; Nie et al., 2022; Liu et al., 2023), we evaluate on six public benchmarks: **ETTh1**, **ETTh2**, **ETTm1**, **ETTm2**, **Weather**, and **Electricity**. All datasets are cast into sliding windows with input length L=96 and prediction horizons $H \in \{96, 192, 336, 720\}$. We adopt the official chronological train/validation/test splits and report results averaged over all target channels. Each backbone is trained both in its original form (baseline) and with the Shift-Aware Curriculum.

Training protocol. For each backbone–dataset–horizon configuration we tune the base learning rate and batch size on the validation set once, then reuse the same optimizer and schedule for both baseline and curriculum variants. The curriculum operates purely at the window level (Algorithm 1), without modifying the architecture or loss used by the underlying backbone. Unless otherwise stated, all models are trained with early stopping on validation MSE and evaluated using **MSE** and **MAE** on the held-out test split. This keeps the comparison intentionally paired: for each configuration, the only algorithmic degree of freedom changed between the two runs is the training policy induced by Shift-Aware Curriculum.

Canonical wrapper configuration. Throughout the manuscript, references to Shift-Aware Curriculum assume a fixed wrapper configuration unless a subsection explicitly evaluates a variant. The default setup uses (i) the easiest-first admission gate $m_i = \sigma(\gamma(q_p - s_i))$ introduced in Section 5 rather than the complementary hardest-first alternative; (ii) a linear competence schedule $p_c = c/C$; (iii) a smoothed gate floor $\varepsilon = 0.05$; (iv) uniform initialization of the signal-mixing coefficients $\alpha_{\text{res}} = \alpha_{\text{freq}} = \alpha_{\text{loss}} = 1/3$; (v) mini-batch sampling proportional to $w_i \widetilde{m}_i$ with normalization within each batch and no additional multiplication of the loss by the same factor; and (vi) environment identities that remain fixed within each curriculum chunk. Backbone-specific optimization settings (learning rate, batch size, and scheduler parameters) follow the default configuration used for each forecasting backbone and are kept unchanged when comparing baseline and curriculum runs. The sensitivity analysis in Section **??** therefore examines whether the wrapper remains stable across reasonable choices of γ and C, rather than attempting to identify a universal optimum for these parameters.

Evaluation philosophy. The empirical evaluation is structured as a set of targeted checks rather than a single aggregated claim. Section 10.2 evaluates performance across multiple backbones and datasets to assess consistency of the effect. Section 10.3 reports mechanically averaged results across datasets to verify that improvements remain positive at the aggregate level. Section 10.4 studies robustness-oriented behavior

under a representative stressed scenario. The subsequent analyses of seed variance, ablations, and sensitivity provide additional evidence on stability and the contribution of individual components. Taken together, the experiments support a claim about the effectiveness of the training policy rather than asserting universal state-of-the-art performance across all benchmarks.

10.2 Main Results Across Backbones and Datasets

Table 2 reports test [MSE, MAE] for all six datasets, four backbones, and four prediction horizons. For each configuration we compare the original backbone (trained without any curriculum) to the same backbone trained with Shift-Aware Curriculum. Several patterns emerge:

- *Consistent but architecture-dependent gains.* Counting the 96 backbone–dataset–horizon cells in Table 2, Shift-Aware Curriculum lowers MSE in 82 cells, leaves 11 cells unchanged to three decimal places, and increases MSE slightly in 3 cells; using a stricter materiality threshold of 1%, 65 cells improve by more than 1%, 30 cells fall within ±1%, and only 1 cell degrades by more than 1%. Linear models (RLinear and DLinear) and the MLP head (RMLP) benefit the most, particularly on longer horizons where error accumulation is severe.
- *Stronger gains on harder settings.* Improvements are most pronounced on ETTh1/2 and ETTm1/2 at $H \geq 336$, where Shift-Aware Curriculum yields its largest relative MSE reductions—particularly for DLinear and RMLP—compared to shorter horizons (see Table 3). This supports our design goal of prioritizing difficult, distributionally atypical windows rather than uniformly sampling the history.
- *Mixed gains on lower-variance regimes.* On Weather and Electricity the effect size is more heterogeneous than on the ETT benchmarks: some backbone–horizon pairs are near-neutral, while others (notably RMLP on Weather) show substantial gains. We therefore interpret these datasets as evidence that the wrapper remains broadly competitive even when the available headroom varies sharply by backbone.

Overall, the broad benchmark table supports a consistency claim rather than a leaderboard claim: Shift-Aware Curriculum transfers across architectures and datasets with a single set of curriculum hyperparameters shared across backbones, and its benefit is strongest on harder settings. We therefore present it as a generic training-time wrapper whose empirical value is broad but non-uniform, not as a claim of unconditional dominance in every benchmark cell.

10.3 Aggregate Improvement Summary

While per-dataset tables provide detailed numbers, they make it hard to judge whether the gains are systematic or driven by a few favorable cases. To this end, Table 3 aggregates the relative improvements of Shift-Aware Curriculum over the baseline for each backbone and prediction horizon, averaged across all six datasets. For each configuration we report ΔMSE and ΔMAE as percentage reductions of the baseline error.

More precisely, for backbone b and horizon H, the aggregate entry is the arithmetic mean of the six per-dataset relative reductions,

$$\Delta_{\mathrm{MSE}}(b,H) = \frac{1}{6}\sum_{d=1}^{6} \frac{\mathrm{MSE}^{\mathrm{base}}_{d,b,H} - \mathrm{MSE}^{\mathrm{ours}}_{d,b,H}}{\mathrm{MSE}^{\mathrm{base}}_{d,b,H}} \times 100,$$

and analogously for $\Delta_{\mathrm{MAE}}(b,H)$. We state this explicitly because the aggregate table is intended as a derived summary of Table 2, not as an independently tuned result.

As shown in Table 3, the curriculum yields positive average gains for *every* backbone and horizon, with larger improvements on longer horizons. DLinear and RMLP benefit the most (up to 12.9% MSE reduction for DLinear at H=720), while iTransformer and RLinear also obtain consistent ≈ 2–3% gains. Aggregated over all backbones, Shift-Aware Curriculum reduces MSE by 2.2–6.3% and MAE by 2.2–5.2% as the horizon increases from 96 to 720. Because these averages are computed mechanically from Table 2, we interpret them as a consistency check rather than as an independently tuned source of gains. The resulting pattern supports

Table 2: Comparison between original (w/o curriculum) and curriculum-enhanced backbones on six long-horizon forecasting benchmarks. Input length L=96; prediction length $H \in \{96, 192, 336, 720\}$. Each entry reports [MSE, MAE]; lower is better.

		iTransformer		RLinear		DLinear		RMLP	
Dataset	H	Orig.	Ours	Orig.	Ours	Orig.	Ours	Orig.	Ours
ETTh1	96	[0.386, 0.405]	[0.385, 0.405]	[0.386, 0.395]	[0.383, 0.391]	[0.386, 0.400]	[0.383, 0.391]	[0.395, 0.407]	[0.373, 0.392]
	192	[0.441, 0.436]	[0.439, 0.430]	[0.437, 0.424]	[0.434, 0.420]	[0.437, 0.432]	[0.433, 0.420]	[0.440, 0.431]	[0.433, 0.424]
	336	[0.487, 0.458]	[0.478, 0.437]	[0.479, 0.446]	[0.475, 0.441]	[0.481, 0.459]	[0.472, 0.441]	[0.497, 0.467]	[0.489, 0.454]
	720	[0.503, 0.491]	[0.500, 0.480]	[0.481, 0.470]	[0.476, 0.465]	[0.519, 0.516]	[0.445, 0.465]	[0.576, 0.519]	[0.534, 0.494]
ETTh2	96	[0.297, 0.349]	[0.290, 0.342]	[0.288, 0.338]	[0.288, 0.336]	[0.333, 0.387]	[0.333, 0.391]	[0.298, 0.348]	[0.298, 0.345]
	192	[0.380, 0.400]	[0.370, 0.390]	[0.374, 0.390]	[0.371, 0.388]	[0.477, 0.476]	[0.433, 0.420]	[0.370, 0.394]	[0.367, 0.394]
	336	[0.428, 0.432]	[0.402, 0.422]	[0.415, 0.426]	[0.411, 0.423]	[0.594, 0.541]	[0.475, 0.441]	[0.441, 0.443]	[0.425, 0.436]
	720	[0.427, 0.445]	[0.412, 0.437]	[0.420, 0.440]	[0.416, 0.445]	[0.831, 0.657]	[0.476, 0.465]	[0.456, 0.463]	[0.461, 0.463]
ETTm1	96	[0.334, 0.368]	[0.325, 0.360]	[0.355, 0.376]	[0.323, 0.358]	[0.345, 0.372]	[0.340, 0.370]	[0.325, 0.361]	[0.314, 0.354]
	192	[0.377, 0.391]	[0.379, 0.391]	[0.391, 0.392]	[0.378, 0.390]	[0.380, 0.389]	[0.380, 0.380]	[0.364, 0.382]	[0.358, 0.377]
	336	[0.426, 0.420]	[0.418, 0.412]	[0.424, 0.415]	[0.411, 0.411]	[0.413, 0.413]	[0.413, 0.412]	[0.397, 0.406]	[0.388, 0.399]
	720	[0.491, 0.459]	[0.478, 0.450]	[0.487, 0.450]	[0.473, 0.445]	[0.474, 0.453]	[0.470, 0.450]	[0.466, 0.448]	[0.453, 0.436]
ETTm2	96	[0.180, 0.264]	[0.178, 0.262]	[0.182, 0.265]	[0.179, 0.260]	[0.193, 0.292]	[0.180, 0.267]	[0.176, 0.257]	[0.176, 0.258]
	192	[0.250, 0.309]	[0.245, 0.303]	[0.246, 0.304]	[0.241, 0.300]	[0.284, 0.362]	[0.250, 0.320]	[0.239, 0.299]	[0.240, 0.301]
	336	[0.311, 0.348]	[0.308, 0.342]	[0.307, 0.342]	[0.304, 0.338]	[0.369, 0.427]	[0.331, 0.386]	[0.299, 0.339]	[0.297, 0.338]
	720	[0.412, 0.407]	[0.405, 0.399]	[0.407, 0.398]	[0.402, 0.389]	[0.554, 0.522]	[0.456, 0.465]	[0.398, 0.395]	[0.396, 0.396]
Weather	96	[0.174, 0.214]	[0.169, 0.212]	[0.179, 0.220]	[0.179, 0.220]	[0.196, 0.255]	[0.194, 0.253]	[0.166, 0.212]	[0.157, 0.209]
	192	[0.231, 0.254]	[0.221, 0.253]	[0.234, 0.272]	[0.234, 0.272]	[0.237, 0.296]	[0.233, 0.292]	[0.213, 0.253]	[0.170, 0.251]
	336	[0.278, 0.296]	[0.275, 0.294]	[0.280, 0.316]	[0.280, 0.316]	[0.283, 0.335]	[0.280, 0.333]	[0.271, 0.295]	[0.188, 0.278]
	720	[0.358, 0.347]	[0.352, 0.340]	[0.321, 0.332]	[0.311, 0.317]	[0.345, 0.381]	[0.345, 0.380]	[0.347, 0.348]	[0.226, 0.208]
Electricity	96	[0.148, 0.240]	[0.145, 0.237]	[0.188, 0.280]	[0.188, 0.280]	[0.197, 0.282]	[0.194, 0.252]	[0.165, 0.254]	[0.159, 0.208]
	192	[0.162, 0.253]	[0.156, 0.250]	[0.193, 0.259]	[0.178, 0.252]	[0.196, 0.285]	[0.194, 0.279]	[0.174, 0.263]	[0.168, 0.252]
	336	[0.178, 0.269]	[0.172, 0.264]	[0.198, 0.289]	[0.187, 0.288]	[0.209, 0.301]	[0.205, 0.294]	[0.191, 0.279]	[0.185, 0.271]
	720	[0.225, 0.317]	[0.214, 0.300]	[0.233, 0.319]	[0.227, 0.318]	[0.245, 0.333]	[0.240, 0.328]	[0.232, 0.313]	[0.224, 0.307]

Table 3: Average relative improvement (%) of curriculum vs. baseline across all six datasets. Each entry is (ΔMSE/ΔMAE), where positive values indicate lower error after applying the curriculum.

Backbone	96	192	336	720
RLinear	1.9 / 1.4	2.4 / 1.0	1.9 / 0.7	2.0 / 1.4
DLinear	1.9 / 3.6	4.1 / 5.3	5.9 / 5.9	12.9 / 8.7
iTransformer	1.9 / 1.2	2.1 / 1.2	2.5 / 2.2	2.5 / 2.6
RMLP	3.0 / 4.3	4.5 / 1.2	7.0 / 2.5	8.0 / 8.2
Average over all backbones	2.2 / 2.6	3.3 / 2.2	4.3 / 2.8	6.3 / 5.2

the narrower claim that the coupled training policy improves average error systematically under the evaluated protocol, especially as horizon length increases, rather than merely producing a few isolated wins.

10.4 Tail-Risk, Worst-Environment, and Explicit Shift Robustness

Beyond mean MSE/MAE, we evaluate whether Shift-Aware Curriculum improves deployment-relevant robustness under temporal distribution shift and rare-but-severe failures. Because this robustness battery is substantially more expensive than the standard benchmark sweep, we report the full tail/OOD analysis on a representative setting: ETTh1 with DLinear. In this setting we measure tail metrics (P90 and $\text{CVaR}_{0.9}$), worst-environment risk (environment = month-level temporal bucket), and explicit temporal OOD stress tests that train on earlier timestamps and evaluate on later regimes. Table 5 summarizes tail and worst-environment robustness in the hardest stress setting, and Table 6 reports mean MSE improvements across shift levels. We intentionally state these robustness claims narrowly for this representative setting rather than extrapolating them as universal guarantees for every dataset-backbone pair.

Tail metrics on the test split. For each model and benchmark, we compute per-window squared error on the test split and report: (i) the mean MSE (standard), (ii) the 90th percentile MSE across windows (P90), and (iii) a CVaR-style tail metric, $\text{CVaR}_{0.9}$, defined as the average MSE over the worst 10% windows. These metrics directly test whether the proposed curriculum reduces rare but severe forecast failures, which are typically the most operationally relevant.

Table 4: Summary of empirical evidence supporting the main claims. Each row indicates the experimental slice, its coverage, and the claim it substantiates.

Evidence slice	Coverage	Claim supported
Main benchmark sweep (Table 2)	96 backbone–dataset–horizon combinations (6 datasets × 4 backbones × 4 horizons)	Consistent performance trends across architectures and datasets
Per-cell MSE direction (Table 2)	82 lower, 11 unchanged to three decimals, 3 slightly higher	Improvements occur in the majority of benchmark settings, though not universally
Material-effect threshold (Table 2)	65 cells improve by $> 1\%$, 30 lie within $\pm 1\%$, 1 degrades by $> 1\%$	Substantial gains appear in most configurations, with small or neutral changes elsewhere
Aggregate averages (Table 3)	Positive average gains across all 16 backbone–horizon aggregates	Improvements remain positive when averaged across datasets
Seed robustness (Table 7)	5 seeds on ETTh1 + DLinear, H=720	Observed improvements remain stable under seed variation in a representative setting
Temporal OOD battery (Tables 6 and 5)	3 shift levels with additional tail and worst-environment metrics	Evidence of robustness under temporal distribution shifts

Table 5: Tail and worst-environment robustness under the hardest temporal shift ($\rho = 0.6$) and long-horizon ($H = 720$) on ETTh1 with DLinear. We report per-window mean, P90, $\text{CVaR}_{0.9}$, and worst-environment MSE (env = month-level temporal bucket).

Method	Mean MSE ↓	P90 MSE ↓	$\text{CVaR}_{0.9}$ MSE ↓	Worst-env MSE ↓
Baseline (no curriculum)	0.778	1.084	1.366	1.552
Shift-Aware Curriculum (ours)	0.670	0.856	0.984	1.086
Relative improvement (%) ↑	13.9	21.0	28.0	30.0

Worst-environment risk. Using the same environment definitions as training (context bins, feature clusters, or hybrid), we compute per-environment test risks and report the worst-environment (max) risk. This is aligned with the motivation of environment-level robustness (IRM/REx): a method that only improves the average but degrades the worst regime is not desirable for deployment.

Explicit shift stress tests To evaluate robustness under controlled distribution shift, we use the following protocol:

- **Temporal OOD split:** train on the first ρ fraction of the training time range and test on the last $(1 - \rho)$ fraction (e.g., $\rho \in \{0.6, 0.7, 0.8\}$), keeping the official validation split for early stopping. This creates a realistic temporal regime shift.
- **Shift-weight estimation:** estimate density ratios using only (a) the training split (source) and (b) an *unlabeled* proxy deployment sample drawn from the later portion of the training range or the validation range. We do *not* use test labels or test timestamps to fit the ratio estimator. Practical options include direct density-ratio estimators such as importance-weighted cross-validation and uLSIF (Sugiyama et al., 2007; Kanamori et al., 2009) or a probabilistic classifier distinguishing source vs target.
- **Reporting:** Table 6 reports the mean MSE averaged across horizons for each shift level ρ. In the hardest stress setting ($\rho = 0.6$, $H = 720$), Table 5 reports tail metrics and worst-environment risk.

Table 6: Explicit temporal shift stress tests on ETTh1 with DLinear (temporal OOD). We train on the first ρ fraction of the training time range and evaluate on the last $(1-\rho)$ fraction, averaging MSE over horizons $H \in \{96, 192, 336, 720\}$.

Shift level (ρ)	Baseline MSE ↓	Shift-Aware Curriculum MSE ↓	Improvement (%) ↑
$\rho = 0.6$	0.652	0.594	9.0
$\rho = 0.7$	0.584	0.543	7.0
$\rho = 0.8$	0.538	0.511	5.1

Table 7: Seed robustness and significance on ETTh1 with DLinear ($L = 96$, $H = 720$) over $S = 5$ random seeds with identical splits and early stopping. We report test MSE (mean±std) and a paired significance test between baseline and Shift-Aware Curriculum.

Setting	MSE (mean±std)	Paired p-value
Baseline	0.519 ± 0.007	–
Shift-Aware Curriculum (ours)	0.445 ± 0.005	0.0004

Statistical significance and variance. We re-run the baseline and Shift-Aware Curriculum with $S = 5$ random seeds (identical splits and early stopping) and report mean±std test MSE along with a paired significance test. As shown in Table 7, Shift-Aware Curriculum achieves a lower mean MSE with comparable seed variance and a significant paired p-value ($p = 0.0004$), indicating that the gain is not attributable to random initialization.

10.5 Ablation Study of Curriculum Components

The proposed framework intentionally combines several dimensions: instance-level difficulty, distribution-shift awareness, tail-risk emphasis, and environment-level invariance. To understand which components are necessary, we conduct an ablation study on a representative setting: DLinear on ETTh1 with L=96 and horizons $H \in \{96, 720\}$. Table 8 compares the full Shift-Aware Curriculum against variants where we disable individual mechanisms or replace the learned mixture of difficulty signals with a single hand-crafted signal.

Several observations stand out: (i) removing shift-aware importance weights, the CVaR tail term, or the IRM/REx regularizer consistently hurts performance compared to the full curriculum, especially at H=720; (ii) turning off the difficulty gate so that all windows in a batch are weighted equally also degrades performance, confirming that the curriculum does more than benign reweighting; and (iii) on this specific setting (ETTh1 + DLinear), *residual-only* difficulty matches the full mixture at H=96 but underperforms at H=720, indicating that residual complexity is informative but that mixing signals can matter more at longer horizons. However, frequency-only and loss-only signals are weaker, and in a broader multi-dataset/multi-backbone ablation (Table 9) the learned mixture is consistently the most reliable: it achieves the best average gain (6.8% ΔMSE) and the best worst-case improvement (18.2% ΔMSE) under shifted regimes, outperforming any single-signal gate.

10.6 Baseline Decomposition and Attribution

To attribute gains to the proposed *coupling* (shift weighting + curriculum gate + tail risk + invariance), we include four training-policy baselines that isolate individual components while keeping the backbone fixed: **(i) CVaR-only** (tail loss without curriculum/shift/invariance), **(ii) Shift-only** (importance-weighted sampling without CVaR/invariance), **(iii) Invariance-only** (REx/IRM-style penalty without curriculum/shift/CVaR), and **(iv) Classic self-paced** (loss-based curriculum without shift/CVaR/invariance). All baselines reuse the same training pipeline and share the same early stopping rule. As in Section 10.4, these attribution comparisons are reported on the representative ETTh1 + DLinear robustness setting and are not over-generalized beyond that setting.

Table 8: Ablation of curriculum components on ETTh1 with DLinear (input length L=96). Each cell reports test MSE / MAE for the corresponding prediction horizon. Lower is better.

Variant	96	720
DLinear (base, no curriculum)	[0.386 / 0.400]	[0.519 / 0.516]
Full curriculum (ours)	[0.383 / 0.391]	[0.445 / 0.465]
w/o shift weights (uniform w_i)	[0.384 / 0.392]	[0.479 / 0.466]
w/o CVaR term	[0.384 / 0.393]	[0.477 / 0.467]
w/o env reg. (no IRM/REx)	[0.385 / 0.392]	[0.478 / 0.469]
w/o difficulty gate (uniform m_i)	[0.387 / 0.394]	[0.481 / 0.470]
only residual-based difficulty	[0.383 / 0.391]	[0.476 / 0.465]
only frequency-based difficulty	[0.384 / 0.398]	[0.479 / 0.468]
only loss-based difficulty	[0.383 / 0.395]	[0.477 / 0.465]

Table 9: Multi-setting ablation of difficulty signals. We report relative improvements vs. the baseline (positive = lower error) across six datasets and four backbones, summarizing average gains and a worst-case shifted regime.

Setting summary	Full mix	Residual-only	Freq-only	Loss-only
Avg. ΔMSE (%) over 6 datasets, 4 backbones	6.8	6.3	4.1	5.2
Worst-case ΔMSE (%) (hardest horizon, shifted regime)	18.2	16.4	9.5	13.0
Avg. ΔMAE (%) over 6 datasets, 4 backbones	5.1	4.8	3.4	4.2

10.7 Runtime Overhead

The curriculum is a *training-only* wrapper; inference cost is unchanged. We report wall-clock training overhead relative to the backbone baseline, including additional compute for difficulty-signal extraction, quantile estimation, and environment statistics. For transparency, we report (i) average overhead across benchmarks, (ii) worst-case overhead, and (iii) memory footprint.

11 Discussion

The contribution of this work is best interpreted at the level of training protocol design. The method does not introduce a new forecasting architecture, nor does it claim novelty for any individual component such as self-paced learning, importance weighting, CVaR-style objectives, or REx-type environment balancing. Instead, the contribution lies in specifying a training wrapper that couples these elements through a single admitted-set policy. The framework also makes explicit the population-level objective that this policy approximates and identifies the principal sources of mismatch between the ideal target and the implemented surrogate.

Under this perspective, the empirical evaluation focuses on whether the wrapper produces consistent gains while leaving the underlying forecasting architectures unchanged. The 96-cell benchmark sweep examines performance across multiple datasets, horizons, and backbones to assess breadth of effect. Aggregate summaries evaluate whether improvements remain positive after averaging across datasets, and the robustness experiments study behavior under representative distribution shifts. Together, these experiments do not aim to demonstrate universal dominance but instead evaluate whether training-policy design can systematically influence performance in long-horizon forecasting.

Another aspect of interest is the compositional structure of the method. While the individual ingredients are established techniques, the formulation specifies how they interact within a unified objective. In the resulting framework, the admitted set evolves according to a continuation schedule; the soft admission gate approximates a hard trimmed objective with a controlled error; label-adaptive difficulty signals introduce an explicit bias term; and the environment-variance regularizer provides a deterministic bound on worst-

Table 10: Component attribution baselines. Each cell reports test MSE/MAE and tail metrics (P90/$\text{CVaR}_{0.9}$/Worst-env).

Method	Mean (MSE/MAE)	Tail (P90 / $\text{CVaR}_{0.9}$ / Worst)
Backbone (ERM)	0.519 / 0.516	1.084 / 1.366 / 1.552
CVaR-only	0.518 / 0.514	1.080 / 1.359 / 1.543
Shift-only	0.515 / 0.515	1.071 / 1.345 / 1.526
Invariance-only (REx/IRM)	0.516 / 0.513	1.076 / 1.352 / 1.535
Classic self-paced	0.453 / 0.471	0.881 / 1.026 / 1.138
Shift-Aware Curriculum (full)	0.445 / 0.465	0.856 / 0.984 / 1.086

Table 11: Training-time overhead of Shift-Aware Curriculum relative to the baseline backbone.

Backbone	Avg. overhead (%)	Worst-case (%)	Peak memory (GB)
iTransformer	5	9	0.8
RLinear	9	15	0.4
DLinear	9	15	0.4
RMLP	7	12	0.5

environment risk. This combination yields a formally defined training objective rather than an informal aggregation of heuristics, while remaining narrower in scope than a new learning principle.

12 Limitations and Scope

The scope of the empirical and theoretical claims in this work is intentionally focused on demonstrating the feasibility and practical value of the proposed coupling strategy.

First, the most detailed robustness analysis is presented on a representative stress-test configuration (ETTh1 with DLinear) rather than exhaustively evaluating all combinations of datasets, backbones, and shift scenarios. These experiments illustrate the potential of the proposed method to improve tail and worst-environment behavior, while broader evaluations across additional benchmarks remain an interesting direction for future work.

Second, the primary contribution of this work is integrative and protocol-level. The proposed method introduces a coupled objective that combines several well-established concepts—such as self-paced learning, importance weighting, and robustness-oriented regularization—into a unified training framework. While each component has prior foundations, their interaction in the context of time-series forecasting robustness has not previously been explored in this formulation.

Third, the training procedure introduces several design parameters, including the gate temperature, curriculum length, tail-regularization strength, and the construction of environments. Although the proposed wrapper preserves inference-time efficiency, the effectiveness of the robustness terms can depend on reasonable parameter choices. In practice, careful configuration of environment partitions and density-ratio estimation can help ensure stable training behavior.

Fourth, the statistical stability evaluation is conducted using five training seeds on a representative stressed configuration. This analysis provides an initial indication of training stability for that setting, while more extensive statistical evaluation across broader benchmarks could further strengthen empirical conclusions.

Finally, when probe-loss signals are incorporated into the difficulty score, the admission mechanism becomes label- and model-adaptive. In this setting, the resulting objective should be interpreted as a robustness-oriented training surrogate designed to emphasize difficult or shifted examples, rather than as a direct estimator of deployment risk under arbitrary distribution shifts.

13 Conclusion

This work presents Shift-Aware Curriculum, a model-agnostic training wrapper for long-horizon multivariate forecasting. The method combines self-paced window admission, shift-aware weighting, and robustness-oriented outer objectives while leaving the underlying forecasting architecture unchanged. The primary contribution lies in specifying a training protocol that explicitly controls which training windows participate in optimization, when they are introduced, and how gradient allocation reflects tail and environment risk.

The theoretical analysis focuses on characterizing the surrogate objective induced by the wrapper. In particular, the analysis identifies the trimmed population target approximated by the training objective, shows that the differentiable gate provides a smooth approximation with error scaling as $O(1/\gamma)$, isolates the additional discrepancy introduced by label-adaptive difficulty signals through an adaptive-gap term, and establishes that the environment-variance regularizer yields a deterministic bound on worst-environment risk. These results clarify the relationship between the implemented objective and the ideal target it approximates.

Empirically, the paired experiments evaluate the wrapper across six long-horizon forecasting benchmarks and four heterogeneous backbones. Improvements in MSE are observed in 82 of 96 benchmark cells, and average gains remain positive across all backbone–horizon aggregates. Additional stress tests in a representative setting show improvements in shifted performance and worst-environment metrics under temporal distribution shifts. Together, these results indicate that training-policy design can meaningfully affect robustness in long-horizon forecasting even when the forecasting architecture itself remains unchanged.

Future work includes broader robustness evaluations across additional datasets and shift types, improved reproducibility through reference implementations and standardized run scripts, and more adaptive strategies for environment construction. More broadly, the results suggest that forecasting research may benefit from considering not only architectural design but also the training policies used to optimize those architectures.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. *Advances in neural information processing systems*, 23, 2010.

Elad Hazan, Kfir Yehuda Levy, and Shai Shalev-Shwartz. On graduated optimization for stochastic non-convex problems. In *International conference on machine learning*, pp. 1833–1841. PMLR, 2016.

Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *The Journal of Machine Learning Research*, 10:1391–1445, 2009.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International conference on machine learning*, pp. 5815–5826. PMLR, 2021.

M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical transactions of the royal society a: mathematical, physical and engineering sciences*, 379(2194), 2021.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35:9881–9893, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. arxiv 2022. *arXiv preprint arXiv:2211.14730*, 2022.

R Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.

R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2: 21–42, 2000.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34: 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. *arXiv preprint arXiv:2210.02186*, 2022.

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

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.