LOSS-GUIDED PRETRAINING DATA SELECTION FOR TIME-SERIES FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series foundation models (TSFMs) are pretrained on heterogeneous collections containing billions of observations, yet their training windows are typically sampled without estimating whether they provide useful learning signal. We introduce a static data-selection framework that scores each window with a reference forecaster and retains an intermediate interval within every source dataset. Specifically, we connect forecasting loss to optimization difficulty by showing that normalized squared loss controls the per-sample gradient norm under a local Jacobian condition. We then define a reference loss score and apply dataset-stratified selection to preserve the diversity of samples. Across various TSFM architectures, our method outperforms random selection by an absolute margin and even improves both relative MASE and CRPS over full-data pretraining by retaining fewer candidate pretraining windows. Further analyses show strong cross-scale and cross-architecture score correlations, indicating that a small reference model can often select data for larger targets, provided that the reference and target share compatible difficulty orderings.

1 INTRODUCTION

Time-series foundation models (TSFMs) aim to transfer temporal structure learned from large, heterogeneous corpora to unseen forecasting tasks. Recent systems such as TimesFM, Chronos, and Moirai demonstrate that a single pretrained model can forecast across domains, sampling frequencies, context lengths, and prediction horizons without task-specific fitting (Das et al., 2024; Ansari et al., 2024; Woo et al., 2024). This universal forecasting paradigm shifts a major part of model development from downstream architecture design to corpus construction. As pretraining collections grow, however, the common assumption that every available window should receive equal attention becomes increasingly costly and scientifically questionable.

Time series samples are constructed by sliding context-future windows over longer sequences as illustrated in Figure 1 and neighboring windows can share part of their observations. Many windows are simple repetitions of seasonal or slowly varying patterns and add little influence once those regularities have been learned as shown in Figure 1(a). At the opposite extreme, a context can be almost constant while its future contains an abrupt, unannounced change. Such a window produces a large loss but supplies no stable relation for a forecaster to learn as in Figure 1(b). Between these extremes are nontrivial yet learnable windows that contain trends, changing seasonality, and moderate stochastic variation, illustrated by Figure 1(c). Therefore, the central question of this paper is how to identify useful training subset from a multi-source corpus that can match or exceed the performance of training on the full dataset.

Loss-based data pruning methods have already been adopted in computer vision and language models. In computer vision, importance-sampling methods prioritize samples using loss-derived bounds on per-sample gradient magnitude, while EL2N ranks images by their prediction-error norm as a tractable proxy for gradient-based importance (Katharopoulos & Fleuret, 2018; Paul et al., 2021). In LLM pretraining, a frozen reference model assigns each document a token-level cross-entropy or perplexity score and these scores are then used to rank and filter the corpus (Marion et al., 2023). A forecasting time series window has the same basic structure as a language sentence. Nonetheless, text perplexity cannot be directly transferred to time series, which have continuous and scale-dependent targets, may contain missing future observations, and are optimized by TSFMs toward

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.

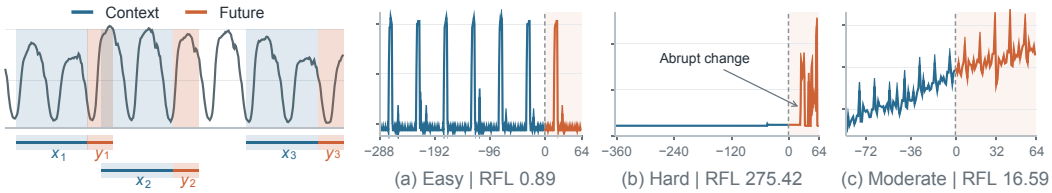


Figure 1: Sliding context-future windows over a long sequence and three real candidate windows. RFL is the reference loss of Chronos-Bolt on each original candidate window.

heterogeneous objectives such as squared error or quantile loss. Existing time-series selection methods focus mainly on task-specific training or foundation-model fine-tuning (Taga et al., 2025; Wu et al., 2026) and are not applicable to large-scale data due to high complexity, leaving data selection for TSFM pretraining largely unexplored.

We propose a loss-guided data selection framework for time series foundation model pretraining. A frozen reference forecaster assigns each candidate context–future window a reference loss score as in Figure 2 (2). The sample loss controls gradient norm under a local Jacobian condition, so loss is a principled proxy for how strongly a window can affect an update. Then dataset-stratified selection strategy retains an intermediate RFL interval within every source as in Figure 2 (3). Specifically, we convert RFL values into percentile ranks within each source dataset. For a retention ratio α, keep the central α fraction of every source. It requires one offline reference-model pass, stores only one scalar per window, and does not compute gradients or modify target training online. In conclusion, RFL identifies a useful difficulty region, whereas stratification preserves source coverage. We evaluate our method by training architecturally different TSFMs from scratch and achieving state-of-the-art results in the experiments.

Our contributions are as follows:

- To the best of our knowledge, we present the first systematic study of loss-based data selection at the window level for pretraining time series foundation models from scratch. This setting selects context–future windows once before target model optimization.
- We formalize reference loss (RFL) as a difficulty score computed by a frozen forecaster and clarify that normalized MSE controls the gradient magnitude under an explicit local-Jacobian condition. We also introduce dataset-stratified selection, which retains an intermediate RFL interval independently within each source dataset and therefore preserves source coverage and dataset-level diversity.
- We train architecturally distinct TSFM families from scratch and our method outperforms random selection by an absolute margin, and even surpasses training on the full sample set. Further studies of reference models and selection strategies characterize when reference scores transfer and validate the components of the framework.

2 Related Work

2.1 Time-series foundation models

Time-series foundation models (TSFMs) replace the conventional one-model-per-dataset workflow with a single forecaster pretrained on heterogeneous time series and transferred to unseen tasks. Representative models differ substantially in how they represent continuous observations and parameterize forecasts. TimesFM applies a decoder-only Transformer to patched continuous inputs (Das et al., 2024). Chronos scales and quantizes observations into a discrete vocabulary and trains a language-model backbone with cross-entropy (Ansari et al., 2024) and Moirai combines masked-encoder pretraining with an any-variate architecture and flexible probabilistic outputs (Woo et al., 2024). Timer-XL (Liu et al., 2025a) formulates univariate, multivariate, and covariate-informed forecasting as multivariate next-token prediction over long contexts. Time-MoE scales decoder-only forecasting through sparse mixture-of-experts layers and large multi-domain corpora (Shi et al.,

2025). Sundial uses a flow-matching objective to generate flexible continuous predictive distributions without a prespecified parametric family (Liu et al., 2025b). The primary focus of these models is architecture and data scaling and we focus on determining which context–future windows within a heterogeneous corpus should be used for pretraining.

2.2 Data selection for large language models

LLM pretraining has motivated data selection at both the document and domain levels. At the document level, large-scale pipelines combine heuristic or learned quality filters with exact and semantic deduplication to remove low-quality and redundant text (Albalak et al., 2024; Lee et al., 2022; Abbas et al., 2023). Model-based methods instead rank documents by reference-model cross-entropy or perplexity (Marion et al., 2023). At the domain level, DoReMi uses proxy-model excess loss to optimize mixture weights (Xie et al., 2023), DOGE estimates how training on one domain affects generalization to others (Fan et al., 2023), and data-mixing laws extrapolate the performance of candidate mixtures from smaller training runs (Ye et al., 2025). Online approaches further adapt sampling probabilities during training (Albalak et al., 2023; Jiang et al., 2025). Our study focuses on TSFM pretraining and we score context–future windows using reference-model loss.

2.3 Data selection for time series

Existing time series selection methods mostly address data valuation, task-specific model training, or foundation-model finetuning rather than pretraining a forecasting TSFM from scratch. TimeInf adapts influence functions to temporally dependent blocks and attributes model predictions to individual time points (Zhang et al., 2025). TSRating distills pairwise LLM judgments into a cross-domain quality rater and evaluates selected subsets with conventional models and TSFM fine-tuning (Wu et al., 2026). AdaRho uses the reducible-loss difference between a target and an adaptively updated reference model for online filtering and augmentation on individual forecasting datasets (Taga et al., 2025). These methods demonstrate the value of time-series-aware selection, but operate in task-specific training or fine-tuning regimes and require repeated adaptation or influence computations. Consequently, they are difficult to apply at TSFM pretraining scale. To the best of our knowledge, this is the first systematic study of loss-based sample-level selection for pretraining of time series foundation models.

3 Method

3.1 Problem formulation

Let the pretraining corpus be partitioned into K source datasets, $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$. A sampled forecasting sample is $z_i = (\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{m}_i)$, where $\boldsymbol{x}_i \in \mathbb{R}^C$ is a context of length C, $\boldsymbol{y}_i \in \mathbb{R}^H$ is the future window and H is the forecast horizon, $\boldsymbol{m}_i \in \{0,1\}^H$ marks observed future positions, where zero indicates a missing value. Our goal is to choose $\mathcal{S} \subset \mathcal{D}$ with $|\mathcal{S}| \approx \alpha|\mathcal{D}|$ such that training a target forecaster on $\mathcal{S}$ retains or improves zero-shot performance while reducing the number of distinct windows that must be stored and processed.

Figure 2 illustrates the complete framework. First, the reference loss score assigns every candidate window a difficulty value using a frozen forecaster. Second, dataset-stratified selection ranks these scores within each source dataset and constructs the retained pretraining subset while preserving source-level diversity.

3.2 Reference loss

Time-series forecasting models are commonly trained with point or probabilistic objectives. Let $M_i = \sum_{h=1}^{H} m_{i,h}$ be the number of observed positions in the future window. For a point forecaster $f_{\boldsymbol{\theta}}$, the masked mean-squared error (MSE) is

$$\mathcal{L}_i^{\mathrm{MSE}}(\boldsymbol{\theta}) = \frac{1}{M_i} \sum_{h=1}^{H} m_{i,h} \left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)_h - y_{i,h} \right)^2 . \tag{1}$$

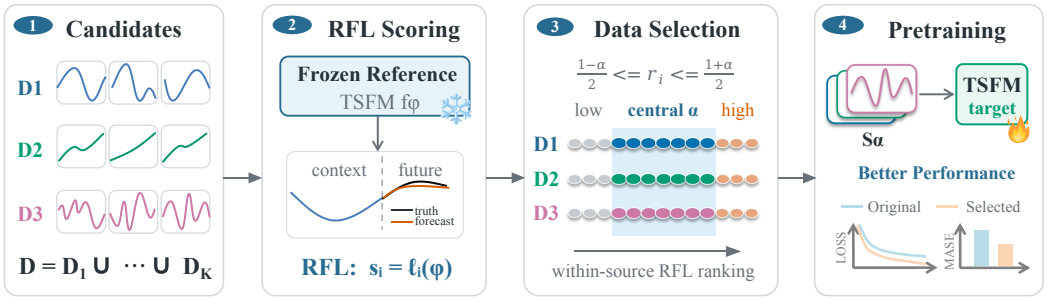


Figure 2: Overview of the selection framework. (1) Candidate context–future windows are collected by their source datasets. (2) A frozen reference TSFM assigns each window an RFL score. (3) Scores are ranked within each source, and the central α interval is retained. (4) The selected subset is used to pretrain the target TSFM, aiming to improve performance relative to full-data pretraining.

MSE measures point accuracy but does not evaluate the full predictive distribution. For a probabilistic forecaster with predictive cumulative distribution $F_{\boldsymbol{\theta},i,h}$, the continuous ranked probability score (CRPS) evaluates both calibration and sharpness. Weighted Quantile Loss (WQL) is a common approximation of CRPS and is commonly adopted in probabilistic forecasting (Gneiting & Raftery, 2007). For simplicity, we mainly use the MSE metric for the analysis.

Reference loss connects forecasting objectives to data selection. Let $f_{\boldsymbol{\phi}}$ be a frozen reference forecaster and let $\ell_i(\boldsymbol{\theta})$ denote the native masked loss on window z_i. We define the reference loss score (RFL) as

$$s_i^{\mathrm{RFL}} = \ell_i(\boldsymbol{\phi}). \tag{2}$$

RFL records the per-window value of an existing forecasting objective under fixed reference model parameters and a larger value indicates that the reference model explains the window less well. Motivated by standard first-order analyses of training-data influence (Pruthi et al., 2020; Xia et al., 2024), we now formalize what RFL reveals about optimization difficulty.

Consider the empirical risk of a model $f_{\boldsymbol{\theta}}$ over the complete candidate pool $\mathcal{D} = \{z_j\}_{j=1}^N$, $N = |\mathcal{D}|$ is the total number of candidate pretraining windows before selection. We have:

$$R(\boldsymbol{\theta}) = \frac{1}{N}\sum_{j=1}^{N} \ell_j(\boldsymbol{\theta}), \qquad \boldsymbol{g}_j = \nabla_{\boldsymbol{\theta}} \ell_j(\boldsymbol{\theta}), \qquad \bar{\boldsymbol{g}} = \nabla_{\boldsymbol{\theta}} R(\boldsymbol{\theta}). \tag{3}$$

To examine the utility of a single window, consider one SGD update $\boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta \boldsymbol{g}_i$. A first-order Taylor expansion of the candidate-pool risk gives $R(\boldsymbol{\theta}^+) - R(\boldsymbol{\theta}) = -\eta\langle \bar{\boldsymbol{g}}, \boldsymbol{g}_i\rangle + o(\eta)$. Thus, positive alignment with the candidate-pool gradient makes the leading-order risk change negative, whereas zero or negative alignment provides no first-order decrease. This separates two properties that a useful window should possess. Its gradient must be large enough to affect the parameters, but it must also align with the population gradient. A corrupted or intrinsically unpredictable future may have a large gradient yet weak or negative utility.

For the MSE objective in Equation 1, let $\boldsymbol{r}_i = \widetilde{\boldsymbol{y}}_i - f_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{x}}_i) \in \mathbb{R}^{M_i}$ collect residuals only at observed future positions. For algebraic convenience, define the half-MSE, which differs from Equation 1 by a positive constant and therefore leaves RFL rankings unchanged:

$$\ell_i^{\mathrm{MSE}}(\boldsymbol{\theta}) = \frac{1}{2}\mathcal{L}_i^{\mathrm{MSE}}(\boldsymbol{\theta}) = \frac{1}{2M_i}\|\boldsymbol{r}_i\|_2^2, \tag{4}$$

Then the connection between MSE loss and gradient magnitude is explicit.

Theorem 3.1 (MSE controls potential update magnitude)**.** *For MSE in Equation 4, the per-window gradient satisfies*

$$\boldsymbol{g}_i = -\frac{1}{M_i}\boldsymbol{J}_i^\top \boldsymbol{r}_i, \qquad \|\boldsymbol{g}_i\|_2^2 = \frac{1}{M_i^2}\boldsymbol{r}_i^\top \boldsymbol{J}_i \boldsymbol{J}_i^\top \boldsymbol{r}_i. \tag{5}$$

where $\boldsymbol{J}_i = \partial f_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{x}}_i)/\partial \boldsymbol{\theta}$ is the Jacobian of the observed predictions. Let $0 \le \lambda_i^- \le \lambda_i^+$ be the smallest and largest eigenvalues of the positive-semidefinite matrix $\boldsymbol{J}_i \boldsymbol{J}_i^\top$. Then

$$\frac{2\lambda_i^-}{M_i}\ell_i^{\mathrm{MSE}} \le \|\boldsymbol{g}_i\|_2^2 \le \frac{2\lambda_i^+}{M_i}\ell_i^{\mathrm{MSE}}. \tag{6}$$

The proof is provided in Appendix B. Theorem 3.1 states the precise sense in which normalized MSE is an optimization-difficulty score. If M_i and local Jacobian conditioning are comparable across windows, ranking their losses approximately ranks the magnitudes of the updates they can induce. For a nonnegative β-smooth objective, the standard self-bounding inequality $\|\nabla \ell_i\|_2^2 \le 2\beta \ell_i$ provides the one-sided result that very low loss rules out a large gradient (Srebro et al., 2010). Evaluating these relations at the frozen reference parameters $\boldsymbol{\phi}$ motivates using s_i^{RFL} as a cheap difficulty coordinate.

3.3 Dataset-stratified selection

Dataset-stratified selection applies RFL-based difficulty filtering while preserving the diversity of the pretraining corpus. We define the candidate corpus as $\mathcal{D} = \{z_i\}_{i=1}^N = \bigcup_{k=1}^K \mathcal{D}_k$, where K is the number of source datasets, $\mathcal{D}_k$ is the set of candidate windows from source k, and $n_k = |\mathcal{D}_k|$. Moreover, each window z_i has an RFL score s_i^{RFL} and a source label $g_i \in \{1, \dots, K\}$, such that $z_i \in \mathcal{D}_{g_i}$.

The preceding first-order Taylor expansion analysis motivates retaining an intermediate RFL interval. Low-RFL windows are expected to provide limited update magnitude, whereas the high-RFL windows are more likely to contain a mixture of rare useful patterns, distribution shifts, and unlearnable noise. By contrast, intermediate RFL windows retain nontrivial residuals while remaining close enough to learned temporal structure that their gradients may align with recurring patterns. This interpretation is consistent with empirical findings on moderate-score and intermediate-perplexity data selection in LLM and CV (Xia et al., 2023; Marion et al., 2023).

Therefore, we first consider *global selection*. It implements this conclusion by pooling all windows, ranking them by absolute RFL, and retaining the central α fraction. Its corpus-level empirical cumulative distribution and selected subset are

$$\widehat{F}_{\mathrm{all}}(t) = \frac{1}{N}\sum_{j=1}^{N} \mathbb{I}[s_j^{\mathrm{RFL}} \le t], \qquad \mathcal{S}_\alpha^{\mathrm{global}} = \left\{ z_i \in \mathcal{D} : \frac{1-\alpha}{2} \le \widehat{F}_{\mathrm{all}}(s_i^{\mathrm{RFL}}) \le \frac{1+\alpha}{2} \right\}. \tag{7}$$

Here, t is an RFL threshold and $\mathbb{I}[\cdot]$ is the indicator function. The desired retention ratio $\alpha \in (0, 1]$ specifies the fraction of the complete corpus to retain, and $\mathcal{S}_\alpha^{\mathrm{global}}$ contains the windows between the $(1-\alpha)/2$ and $(1+\alpha)/2$ quantiles of the pooled distribution.

Global selection controls $|\mathcal{S}_\alpha^{\mathrm{global}}| \approx \alpha N$, but it does not control the source-specific retention rate. Absolute RFL distributions can differ across datasets because of their noise levels, temporal patterns, and intrinsic predictability. Consequently, a source whose scores concentrate near either global tail can be severely underrepresented or removed, which may harm dataset diversity.

Dataset-stratified selection instead measures the difficulty of each window relative to other windows from the same source. For source k, we define the source-specific empirical cumulative distribution $\widehat{F}_k$ and the within-source percentile rank r_i as

$$\widehat{F}_k(t) = \frac{1}{n_k}\sum_{j:g_j=k} \mathbb{I}[s_j^{\mathrm{RFL}} \le t], \qquad r_i = \widehat{F}_{g_i}(s_i^{\mathrm{RFL}}). \tag{8}$$

Here, $r_i \in (0, 1]$ is the fraction of windows in $\mathcal{D}_{g_i}$ whose RFL score does not exceed that of window i. We then retain the central α fraction independently within every source:

$$\mathcal{S}_\alpha \equiv \mathcal{S}_\alpha^{\mathrm{strat}} = \left\{ z_i \in \mathcal{D} : \frac{1-\alpha}{2} \le r_i \le \frac{1+\alpha}{2} \right\}. \tag{9}$$

Equation 9 retains αn_k windows from every source k. The proportion of source k in the selected subset is therefore

$$p_k^{\mathcal{S}} = \frac{|\mathcal{S}_\alpha \cap \mathcal{D}_k|}{|\mathcal{S}_\alpha|} = \frac{\alpha n_k}{\alpha N} = \frac{n_k}{N} = p_k, \tag{10}$$

Algorithm 1 Reference-loss scoring and dataset-stratified selection

Require: Candidate corpus $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$; frozen reference model $f_{\boldsymbol{\phi}}$; retention ratio α
Ensure: Selected subset $\mathcal{S}_\alpha$ and trained target parameters $\boldsymbol{\theta}$
$\mathcal{S}_\alpha \leftarrow \varnothing$
for each candidate window $z_i \in \mathcal{D}$ **do**
  Compute its RFL score $s_i^{\mathrm{RFL}} \leftarrow \ell_i(\boldsymbol{\phi})$
end for
for each source dataset $k \in \{1, \ldots, K\}$ **do**
  Sort the n_k windows by increasing RFL score
  Denote the resulting order by $z_{\pi_k(1)}, \ldots, z_{\pi_k(n_k)}$
  $m_k \leftarrow \max\{1, \mathrm{round}(\alpha n_k)\}$
  $a_k \leftarrow \lfloor (n_k - m_k)/2 \rfloor + 1$
  $\mathcal{S}_\alpha \leftarrow \mathcal{S}_\alpha \cup \{z_{\pi_k(j)} : a_k \le j < a_k + m_k\}$
end for
Train the randomly initialized target model $f_{\boldsymbol{\theta}}$ on $\mathcal{S}_\alpha$
return $\mathcal{S}_\alpha, \boldsymbol{\theta}$

where $p_k = n_k/N$ is the proportion of source k in the original candidate corpus. Dataset-stratified selection thus filters windows by relative difficulty while preserving source coverage and dataset-level diversity. We analyze the benefit of the framework in Section 4.4.2.

3.4 Algorithm and computational cost

Algorithm 1 summarizes the two components of the framework. We first assign every candidate window an RFL score with a frozen reference model. We then rank the scores separately within each source dataset and retain the central interval. The reference and target architectures may differ, which permits an inexpensive proxy to amortize selection over several target models.

The offline algorithm requires one reference-model forward pass per candidate window. It additionally requires $\sum_{k=1}^{K} O(n_k \log n_k)$ time to sort the scores within sources and stores one scalar per window. The generated scores can be reused across target-model sizes and architectures, amortizing the scoring cost. Appendix D reports the measured GPU cost of reference scoring and target model pretraining.

4 Experiments

We organize the experiments around three questions. (Q1) can dataset-stratified RFL selection improve full-data pretraining while retaining fewer training samples? (Q2) can RFL scores be transferred across reference-model scales and architectures? (Q3) how important are the intermediate interval and dataset stratification?

4.1 Experimental setup

Data. The pretraining data mainly comes from the Chronos data collection (Ansari et al., 2024) and LOTSA (Woo et al., 2024). It contains more than 100 source datasets and 4.64M univariate series, spanning climate and weather, transportation, health and mobility data, industrial measurements, synthetic series generated via Gaussian processes(Ansari et al., 2024) and so on. We build validation windows from the last complete horizon of each series and keep them separate from the training windows. All architectural variants use the same validation samples to ensure comparable validation losses. Appendix C lists the source-level properties used by the training pipeline.

Training. We pretrain three representative TSFM families: Chronos-Bolt Base (Ansari et al., 2024) (encoder-decoder), Moirai-Base (Woo et al., 2024) (encoder), and TimesFM 2.5 (Das et al., 2024) (decoder). All models use AdamW, cosine learning-rate decay, gradient clipping, and an effective batch size of 512 on NVIDIA A100 GPUs. The initial learning rate is 3×10^{-4} for TimesFM and 1×10^{-3} for the other models, with 5,000 and 2,000 warm-up steps, respectively. We set the training

Table 1: GIFT-Eval comparison at different retention ratios. Lower MASE and CRPS are better. Full pretraining uses 100% of the candidate windows. Bold denotes the best selection method.

Target model	Metric ↓	Full (100%)	40% retention		60% retention		80% retention	
			Random	Ours	Random	Ours	Random	Ours
Chronos-Bolt Base	MASE	0.796	0.836	0.803	0.806	0.801	0.799	**0.777**
	CRPS	0.555	0.576	0.569	0.555	0.554	0.549	**0.542**
Moirai-Base	MASE	1.017	1.026	1.050	1.061	1.031	1.022	**1.005**
	CRPS	0.698	0.708	0.749	0.732	0.731	0.701	**0.680**
TimesFM 2.5	MASE	0.813	0.833	0.821	0.840	0.823	0.835	**0.804**
	CRPS	0.558	0.573	0.561	0.574	0.560	0.576	**0.554**

budget to 40,000 optimizer steps for TimesFM and 100,000 steps for others. The max context length is set to 2,048 and prediction length is architecture-specific: 64 for Chronos-Bolt, 96 for Moirai, and 128 for TimesFM. The implementation is available at https://anonymous.4open.science/r/tsfm-selection-EF61/.

Evaluation. We use GIFT-Eval (Aksu et al., 2024) and remove training-overlapping benchmarks. The current evaluation contains 91 task configurations. Following GIFT-Eval, we normalize MASE and CRPS by a seasonal-naive forecaster and report the geometric mean across tasks. We also report MSE and MAE on five commonly used benchmarks in ablation study: ETTh1, ETTh2, ETTm1, ETTm2, and weather. These datasets have been utilized for benchmarking and publicly available on (Wu et al., 2021). None of these datasets is included in pretraining datasets.

4.2 Overall Comparison

We compare dataset-stratified RFL selection with uniform random selection and full data pretraining under the same computational budget. To test whether the selection rule generalizes beyond model design, we pretrain three distinct target architectures and score their candidate pools using frozen references from the corresponding architecture. Table 1 shows that our method outperforms random selection by an absolute margin, indicating that RFL provides a useful selection signal across architectures. Since the random control retains the same number of windows, this consistency attributes the gain to which windows are retained. At 80% retention, our method improves over both full-data pretraining and a size-matched random subset on all target models. This is because we remove high-loss noise and harmful data and prune unnecessary low-loss data while maintaining diversity.

4.3 Reference-model analysis

We investigate how the choice of reference model affects data selection along two dimensions: scale transfer within an architecture and transfer across architectures. We use Chronos-Bolt Base and Moirai-Base as target models and retain 80% of the candidate windows. For within-architecture transfer, we vary the frozen reference among Chronos-Bolt Mini and Small, or between Moirai-Small and Base. For cross-architecture transfer, a frozen TimesFM reference selects data for each target. This design tests whether RFL requires a capacity-matched reference or can instead reuse a smaller or architecturally different pretrained forecaster.

Within-architecture references are largely robust to model scale. As shown in Table 2, all subset configurations improve MASE and CRPS over full-data pretraining, indicating small reference models produce competitive training subsets. The score analysis in Appendix E explains this robustness. In brief, the pairwise Spearman correlations among Chronos-Bolt families range from 0.977 to 0.986, and Moirai families obtain $\rho = 0.991$. This agreement indicates that models from the same architecture share a stable ordering of window difficulty. Consequently, a smaller pretrained reference can often approximate the ranking of a larger model and reduce the cost of the required forward scoring pass.

Cross-architecture transfer is conditional. A TimesFM reference selects a useful subset for Chronos-Bolt Base target, improving full-data pretraining and remaining competitive with the best reference.

Table 2: Evaluation results of cross-scale and cross-architecture reference models with 80% retention. Full pretraining uses 100% of the candidate windows. Bold and underlined values are best and second best for each target model.

Target Model	Reference Model	Retained	MASE	CRPS
Chronos-Bolt Base	Chronos-Bolt Mini	80%	**0.777**	<u>0.542</u>
	Chronos-Bolt Small	80%	<u>0.786</u>	0.545
	TimesFM	80%	<u>0.786</u>	**0.541**
	— (Full data)	100%	0.796	0.555
Moirai-Base	Moirai-Small	80%	**1.003**	<u>0.695</u>
	Moirai-Base	80%	<u>1.005</u>	**0.680**
	TimesFM	80%	1.022	0.708
	— (Full data)	100%	1.017	0.698

In contrast, the TimesFM-selected subset performs poorly in Moirai-Base pretraining, degrading to the level of random selection in Table 1. Indeed, TimesFM has strong RFL score rank agreement with Chronos-Bolt Mini on the candidate pool ($\rho = 0.828$), but only weak agreement with Moirai-Small ($\rho = 0.254$). For more analysis details, please see Appendix E. A plausible explanation is that TimesFM scores point and quantile errors, whereas Moirai uses a distributional negative log-likelihood, so they may not assign similar difficulty to high-uncertainty windows.

4.4 Ablation studies

4.4.1 RFL region

Dataset-stratified selection retains the central RFL interval by default. To isolate the effect of this choice, we compare three score-based subsets constructed independently within every source. Bottom retains the lowest-RFL tail, Top retains the highest-RFL tail, and Central retains the central interval. We additionally include a dataset-stratified random baseline that samples windows uniformly at random within each source, retaining the prescribed fraction from every source. All runs in this ablation use TimesFM as both the reference and target model. We evaluate 40% and 80% retention on five forecasting benchmarks in Table 3 and compare matched validation-loss trajectories through 40k optimization steps against source-matched random selection and full-data pretraining in Figure 3.

Table 3 demonstrates the advantages of the Central selection, consistent with the optimization interpretation in Section 3.2. Bottom retains low-RFL windows that the reference model already predicts well and are therefore relatively easy. Concentrating on this region can overallocate training capacity to redundant patterns with little residual learning signal. One possible explanation for this redundancy is that intermediate-difficulty windows contain the regularities needed for the easier cases. Once the target model learns these richer patterns, it can readily generalize to low-difficulty windows without repeatedly training on them. By contrast, Top retains high-difficulty windows. Although some of these windows may contain rare informative structure, a high loss can also arise from noise, abrupt unpredictable transitions, or distribution shifts, making the entire upper tail less reliably learnable. Central avoids both extremes and retains windows with nontrivial prediction error but sufficient regularity to provide stable learning signal.

The validation trajectories in Figure 3(a,b) reinforce this interpretation. Central remains among the lowest-loss trajectories over much of training and separates most clearly when the retained budget is small. This suggests that the benefit comes from excluding both simple windows and the most unstable tail, not from preferring uniformly easier or harder data.

4.4.2 Dataset stratification selection

Dataset stratification is the diversity-preserving component of our framework. We compare the proposed dataset-stratified rule in Equation 9 with global selection in Equation 7. Both rules retain the same central RFL interval and use the same total retention ratio. They differ only in whether percentile ranks are computed within each source or over the pooled corpus. We use the default reference model to score the candidate pool and pretrain a Chronos-Bolt Base target model on the

Table 3: RFL-region ablation at 40% and 80% retention, using TimesFM as both reference and target model. During downstream evaluation, each dataset uses a context window of 512 and a prediction window of 96. Lower MAE and MSE values indicate superior performance.

Ratio	Region	ETTh1		ETTh2		ETTm1		ETTm2		Weather	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
40%	Bottom	0.404	0.393	0.307	0.342	0.351	0.359	0.192	0.260	0.182	0.214
	Top	0.394	0.411	0.315	0.358	0.357	0.380	0.197	0.273	0.180	0.221
	Random	0.388	0.398	**0.302**	0.348	0.347	0.365	**0.182**	0.259	0.176	0.212
	Ours	**0.383**	**0.390**	**0.302**	**0.337**	**0.341**	**0.357**	0.186	**0.258**	**0.169**	**0.202**
80%	Bottom	0.386	0.391	0.300	0.337	0.348	**0.357**	0.184	**0.255**	0.172	0.206
	Top	0.391	0.400	0.305	0.349	0.346	0.366	0.182	0.259	0.171	0.209
	Random	0.379	0.390	0.294	0.347	0.345	0.367	0.189	0.262	0.173	0.210
	Ours	**0.373**	**0.388**	**0.287**	**0.334**	**0.339**	0.359	**0.181**	**0.255**	**0.170**	**0.204**
100%	Full	0.397	0.398	0.297	0.342	0.350	0.363	0.187	0.261	0.175	0.210

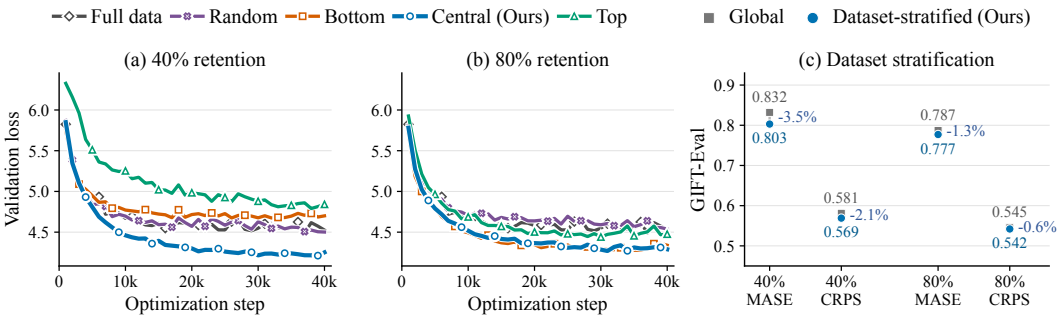


Figure 3: Ablations of the two components of our framework. (a,b) Validation-loss trajectories for full-data pretraining, random selection and the Bottom, Central (Ours), and Top RFL regions at 40% and 80% retention. Stars mark the minimum loss observed through 40k steps. (c) GIFT-Eval results for dataset-stratified selection and global selection at 40% and 80% retention on Chronos-Bolt Base target model. Annotated percentages are relative reductions from global selection.

selected subsets. Figure 3(c) shows that the stratified rule improves GIFT-Eval metrics at all retention ratios. This improvement indicates that global percentiles harm the data diversity, whereas within-source percentiles preserve coverage. The persistence of the advantage from 40% to 80% supports dataset stratification as an effective diversity constraint.

5 Conclusion

This paper presents a reference-loss selection framework for selecting training windows for time-series foundation-model pretraining. We introduce reference loss as a model-compatible difficulty score computed by a frozen forecaster. Our analysis connects loss to potential update magnitude while clarifying that actual one-step utility also depends on gradient alignment. Based on this, our dataset-stratified selection retains an intermediate interval within every source dataset. The resulting subset removes low-loss windows with limited residual learning signal and avoids overconcentrating on the unstable high-loss tail, while preserving source coverage and dataset-level diversity. Across distinct TSFM families, our method outperforms size-matched random selection by an absolute margin, and even surpasses the training on the full sample set. The central region and stratification ablations independently validate the use of intermediate difficulty and within-source ranking. Reference-model analysis further indicates the practicality of cross-scale and cross-architecture. Together, these findings establish pretraining-data selection as an important design axis alongside model architecture and corpus scale for time series foundation models.

AI USE STATEMENT

In this work, we used generative AI tools to assist in translating manuscript content, and interpreting experimental results. We did not use generative AI tools to formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, write proofs, design or provide feedback on the research methodology or experiments, implement methods, clean or reformat datasets, generate synthetic datasets, or support qualitative or thematic data analysis. Developing theoretical models or conceptual frameworks is not applicable to this work. Additionally, we used generative AI tools to inspect or debug software code, draft parts of the paper, edit the manuscript for readability, identify relevant literature, and format references.

We reviewed all AI-assisted work. AI-assisted debugging and editing suggestions were manually inspected and tested. Literature suggestions and formatted references were verified against the original publications, and all translated, drafted, or edited text was manually reviewed and revised. We take responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies data selection for forecasting models and does not collect new human-subject data or conduct user studies. Some source collections concern health, mobility, energy, or economic activity, but we use only the time-series records distributed by their maintainers and do not attempt to identify individuals.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce the method, along with an anonymous code link, in Section 4.1. Appendix B gives the complete proof of the theoretical result. Appendix C lists all pretraining datasets.

REFERENCES

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. SemDeDup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

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

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing for language model pre-training. In *NeurIPS Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models*, 2023.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Hae Won Jeong, et al. A survey on data selection for language models. *Transactions on Machine Learning Research*, 2024.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

Zalán Borsos, Mojmír Mutný, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2024.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. DOGE: Domain reweighting with generalization estimation. In *NeurIPS Workshop on Agent Learning in Open-Endedness*, 2023.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Yiding Jiang, Allan Zhou, Zhili Feng, Sadhika Malladi, and J. Zico Kolter. Adaptive data optimization: Dynamic sample selection with scaling laws. In *International Conference on Learning Representations*, 2025.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

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.

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*, 2025b.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining LLMs at scale. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

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.

Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Smoothness, low noise and fast rates. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

Ege Onur Taga, Halil Alperen Gozeten, Kutay Tire, Rahul Dalvi, Reinhard Heckel, and Samet Oymak. Filter, augment, forecast: Online data selection for robust time series forecasting. In *ICML Workshop on Foundation Models for Structured Data*, 2025.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

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*, 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.

Shunyu Wu, Dan Li, Wenjie Feng, Haozheng Ye, Jian Lou, and See-Kiong Ng. Rating quality of diverse time series data by meta-learning from LLM judgment. In *International Conference on Learning Representations*, 2026.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 54104–54132, 2024.

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *International Conference on Learning Representations*, 2023.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, 2023.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *International Conference on Learning Representations*, 2025.

Yizi Zhang, Jingyan Shen, Xiaoxue Xiong, and Yongchan Kwon. TimeInf: Time series data contribution via influence functions. In *International Conference on Learning Representations*, 2025.

A Related work

A.1 Data selection in computer vision

Geometry-based selection. Geometric methods construct a coreset that preserves coverage or removes redundancy in a feature space. Moderate-DS selects samples near the median distance to their class centers to obtain subsets that remain robust across pruning scenarios (Xia et al., 2023), whereas SemDeDup clusters pretrained embeddings and removes semantically redundant samples within each cluster (Abbas et al., 2023). Such methods make diversity explicit, but their behavior depends on the representation space and, in supervised variants, on class structure that is not naturally available for heterogeneous forecasting windows.

Score-based selection. Score-based methods assign each sample an independent importance or difficulty value and retain a prescribed region of the score distribution. Forgetting events identify samples whose predictions repeatedly switch from correct to incorrect during training (Toneva et al., 2019); GraNd and EL2N approximate sample importance using early-training gradient norms and prediction errors (Paul et al., 2021). These methods established that training dynamics can expose redundant, ambiguous, or noisy samples, but computing scores over multiple initializations or training epochs is expensive at foundation-model scale.

Optimization-based selection. Optimization-based methods select subsets according to the update or validation behavior they induce. CRAIG chooses a weighted coreset whose gradients approximate the full-data gradient (Mirzasoleiman et al., 2020). GLISTER formulates selection as a bilevel objective that improves held-out likelihood (Killamsetty et al., 2021) and bilevel coreset methods directly optimize subset weights for continual or streaming learning (Borsos et al., 2020). These approaches provide strong optimization semantics but typically require repeated gradients, validation targets, or inner-loop training.

B Proof of Theorem 3.1

Proof. Recall that $\boldsymbol{r}_i = \widetilde{\boldsymbol{y}}_i - f_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{x}}_i)$ contains the M_i residuals at observed future positions and that

$$\ell_i^{\mathrm{MSE}}(\boldsymbol{\theta}) = \frac{1}{2M_i} \boldsymbol{r}_i^\top \boldsymbol{r}_i. \tag{11}$$

Because $\partial \boldsymbol{r}_i / \partial \boldsymbol{\theta} = -\boldsymbol{J}_i$, the chain rule gives

$$\boldsymbol{g}_i = \nabla_{\boldsymbol{\theta}} \ell_i^{\mathrm{MSE}}(\boldsymbol{\theta}) = \frac{1}{M_i} \left(\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{\theta}} \right)^\top \boldsymbol{r}_i = -\frac{1}{M_i} \boldsymbol{J}_i^\top \boldsymbol{r}_i. \tag{12}$$

Taking its squared Euclidean norm yields

$$\|\boldsymbol{g}_i\|_2^2 = \frac{1}{M_i^2} (\boldsymbol{J}_i^\top \boldsymbol{r}_i)^\top (\boldsymbol{J}_i^\top \boldsymbol{r}_i) = \frac{1}{M_i^2} \boldsymbol{r}_i^\top \boldsymbol{J}_i \boldsymbol{J}_i^\top \boldsymbol{r}_i, \tag{13}$$

which proves Equation 5.

Let $\boldsymbol{A}_i = \boldsymbol{J}_i \boldsymbol{J}_i^\top$. This matrix is positive semidefinite because, for every vector $\mathbf{u}$, $\mathbf{u}^\top \boldsymbol{A}_i \mathbf{u} = \|\boldsymbol{J}_i^\top \mathbf{u}\|_2^2 \geq 0$. Applying the Rayleigh–Ritz inequality to $\boldsymbol{A}_i$ gives

$$\lambda_i^- \|\boldsymbol{r}_i\|_2^2 \leq \boldsymbol{r}_i^\top \boldsymbol{A}_i \boldsymbol{r}_i \leq \lambda_i^+ \|\boldsymbol{r}_i\|_2^2. \tag{14}$$

Finally, Equation 11 implies $\|\boldsymbol{r}_i\|_2^2 = 2M_i \ell_i^{\mathrm{MSE}}$. Substituting this identity into Equation 14 and dividing by M_i^2 yields

$$\frac{2\lambda_i^-}{M_i} \ell_i^{\mathrm{MSE}} \leq \|\boldsymbol{g}_i\|_2^2 \leq \frac{2\lambda_i^+}{M_i} \ell_i^{\mathrm{MSE}}, \tag{15}$$

which is Equation 6. □

C PRETRAINING DATASETS

Table 4: Pretraining datasets and key properties.

Dataset	Domain	Freq.	Series	MB
alibaba_cluster_trace_2018	Cloud/compute	5T	116,818	767.3
australian_electricity_demand	Energy/building	30T	5	4.6
azure_vm_traces_2017	Cloud/compute	5T	159,472	10632.0
bdg-2_bear	Energy/building	H	91	5.9
bdg-2_fox	Energy/building	H	135	9.3
bdg-2_panther	Energy/building	H	105	3.7
bdg-2_rat	Energy/building	H	280	18.9
beijing_air_quality	Climate/weather	H	132	18.5
BEIJING_SUBWAY_30MIN	Transportation	30T	552	3.5
bitcoin_with_missing	Forecasting benchmark	D	18	0.3
borealis	Climate/weather	H	15	0.3
borg_cluster_data_2011	Cloud/compute	5T	286,772	4314.8
buildings_900k	Energy/building	H	1,795,256	63020.1
bull	Ecology/other	H	41	11.5
cdc_fluview_ilinet	Health/mobility	W	375	1.3
cdc_fluview_who_nrevss	Health/mobility	W	296	0.7
china_air_quality	Climate/weather	H	2,622	137.9
cockatoo	Ecology/other	H	1	0.4
Commerce	Retail/economy	–	41,200	1546.3
covid19_energy	Energy/building	H	1	0.9
covid_mobility	Health/mobility	D	362	0.6
eco_data	Ecology/other	–	19,632	6785.4
elecdemand	Energy/building	30T	1	0.1
elf	Ecology/other	H	1	0.1
ercot	Energy/building	1H	8	5.0
exchange_rate	Forecasting benchmark	irregular	8	0.2
extended_web_traffic_with_missing	Forecasting benchmark	D	145,063	1487.7
favorita_sales	Retail/economy	D	111,840	560.2
favorita_transactions	Retail/economy	D	54	0.3
fred_md	Forecasting benchmark	M	107	0.3

Table 4: Pretraining datasets and key properties, continued.

Dataset	Domain	Freq.	Series	MB
gfc12_load	Energy/building	H	20	6.3
gfc14_load	Energy/building	H	1	0.1
gfc17_load	Energy/building	H	8	1.1
GREEND_1	Energy/building	–	264	3984.9
GREEND_2	Energy/building	–	321	6766.9
hog	Ecology/other	H	24	10.1
HZMETRO	Transportation	15T	160	1.5
ideal	Energy/building	H	217	5.0
kaggle_web_traffic_weekly	Forecasting benchmark	W-SUN	145,063	70.7
kdd2022	Cloud/compute	10T	134	189.1
largest_2017	Ecology/other	5T	8,196	3446.5
largest_2018	Ecology/other	5T	8,428	3544.1
largest_2019	Ecology/other	5T	8,600	3616.4
largest_2020	Ecology/other	5T	8,561	3609.8
largest_2021	Ecology/other	5T	8,548	3594.5
lcl	Energy/building	H	713	38.2
london_smart_meters_with_missing	Energy/building	30T	5,520	665.1
LOS_LOOP	Transportation	5T	207	28.4
m1_monthly	Forecasting benchmark	M	617	0.2
m4_daily	Forecasting benchmark	D	4,227	40.2
m4_hourly	Forecasting benchmark	H	414	1.5
m4_monthly	Forecasting benchmark	M	48,000	46.2
m4_quarterly	Forecasting benchmark	Q-DEC	24,000	10.3
m4_weekly	Forecasting benchmark	W-SUN	359	1.5
m4_yearly	Forecasting benchmark	A-DEC	22,974	4.1
m5	Retail/economy	D	30,490	234.8
mexico_city_bikes	Transportation	1H	493	154.8
monash_m3_monthly	Forecasting benchmark	M	1,428	0.6
monash_m3_other	Forecasting benchmark	Q-DEC	174	0.1
monash_m3_quarterly	Forecasting benchmark	Q-DEC	756	0.2
monash_m3_yearly	Forecasting benchmark	A-DEC	645	0.1
nn5_daily_with_missing	Forecasting benchmark	D	111	0.3
nn5_weekly	Forecasting benchmark	W-SUN	111	0.1
oikolab_weather	Climate/weather	H	8	3.2
pdb	Industrial/science	H	1	0.1
pedestrian_counts	Transportation	H	66	12.5
PEMS03	Transportation	5T	358	37.5
PEMS04	Transportation	5T	921	62.6
PEMS07	Transportation	5T	883	99.7
PEMS08	Transportation	5T	510	36.5
PEMS_BAY	Transportation	5T	325	67.8
phm_data	Industrial/science	–	340	5588.9
project_tycho	Health/mobility	W	1,258	5.6
Q-TRAFFIC	Transportation	15T	45,148	1059.1
residential_load_power	Energy/building	T	813	1752.0
residential_pv_power	Energy/building	T	699	1504.1
rideshare_with_missing	Transportation	H	2,304	3.5
sceaux	Energy/building	H	1	0.1
smart	Other	H	5	0.4
solar_power	Energy/building	4S	1	29.6
spain	Other	H	1	0.3
specific_humidity	Climate/weather	1H	26,624	37345.5
subseasonal	Climate/weather	D	3,448	227.3
subseasonal_precip	Climate/weather	D	862	39.1
sunspot_with_missing	Forecasting benchmark	D	1	0.3

Table 4: Pretraining datasets and key properties, continued.

Dataset	Domain	Freq.	Series	MB
synth	Synthetic	–	1,000,000	16404.0
taxi_30min	Transportation	30M	2,428	14.6
toa_incident_solar_radiation	Climate/weather	1H	2,048	2872.7
total_cloud_cover	Climate/weather	1H	2,048	2872.7
total_precipitation	Climate/weather	1H	2,048	2872.7
tourism_monthly	Retail/economy	M	366	0.4
tourism_quarterly	Retail/economy	Q-DEC	427	0.2
tourism_yearly	Retail/economy	A-DEC	419	0.1
traffic_hourly	Transportation	H	862	59.9
traffic_weekly	Transportation	W-SUN	862	0.4
uber_tlc_daily	Transportation	D	262	0.2
uber_tlc_hourly	Transportation	H	262	4.5
vehicle_trips_with_missing	Transportation	D	329	0.1
vorticity	Climate/weather	1H	26,624	37345.3
weatherbench_daily	Climate/weather	1D	225,280	13189.3
weatherbench_weekly	Climate/weather	7D	225,280	1904.8
wiki-rolling_nips	Forecasting benchmark	D	47,675	163.7
wind_farms_with_missing	Climate/weather	T	337	688.7
wind_power	Climate/weather	4S	1	29.6
10m_u_component_of_wind	Climate/weather	–	2,048	8617.5
10m_v_component_of_win	Climate/weather	–	2,048	8617.5
2m_temperature	Climate/weather	–	2,048	8617.5

D EFFICIENCY ANALYSIS

RFL selection introduces a one-time offline scoring pass over approximately 11M candidate windows per reference model. Table 5 reports the reference-scoring and target-pretraining costs on NVIDIA A100 GPUs. Each reference-scoring run uses one GPU. Owing to resource constraints, Moirai-Base pretraining uses one GPU, whereas TimesFM 2.5 and Chronos-Bolt Base use three GPUs in parallel for pretraining. Results show that small reference models substantially reduce the offline cost. The scoring pass is performed once, and its scalar score cache can be reused across retention ratios, selection regions, and target-training runs.

Table 5: Measured GPU cost of offline RFL scoring and target-model pretraining on A100 40GB GPUs. Each reference scores approximately 11M candidate windows. Moirai uses 1 GPU, while the other models use 3 GPUs in parallel for pretraining.

Offline reference scoring		**Target-model pretraining**		
Reference model	GPU-hours	Target model	Training steps	GPU-hours
Moirai-Small	0.92	Moirai-Base	100,000	17.47 (×1)
Moirai-Base	3.17			
TimesFM 2.5	2.50	TimesFM 2.5	40,000	7.14 (×3)
Chronos-Bolt Mini	0.67	Chronos-Bolt Base	100,000	15.59 (×3)
Chronos-Bolt Small	0.83			

E ADDITIONAL REFERENCE-MODEL ANALYSIS

Within-architecture RFL rankings are largely insensitive to reference scale. Figure 4(a) compares Chronos-Bolt Mini, Small, and Base on the same candidate pool. Their pairwise Spearman correlations remain high even though the models differ in capacity. Moreover, we also compared the RFL scores computed with Moirai-base and Moirai-small as reference models, respectively, and the

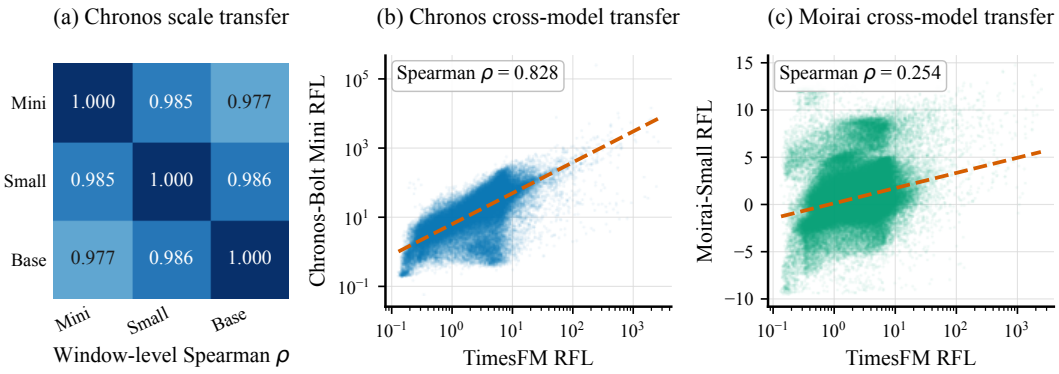


Figure 4: Reference-model agreement on target-specific candidate pools. (a) Window-level Spearman correlations among Chronos-Bolt Mini, Small, and Base over all candidate windows for the Chronos-Bolt-Base target. (b) TimesFM versus Chronos-Bolt Mini on the same Chronos target pool. (c) TimesFM versus Moirai-Small on the Moirai-Base target pool. Spearman coefficients use global ranks over every aligned window. For legibility, each scatter panel displays a deterministic sample of 150k windows, while the reported correlation and dashed ordinary-least-squares fit use all windows.

Spearman correlation between them is 0.991. This near-invariance indicates that the same architecture shares a stable notion of relative window difficulty. This agreement helps explain why the smaller reference models in Table 2 produce competitive training subsets.

Cross-architecture reuse depends on whether the reference and target families induce compatible difficulty orderings. On the Chronos-Bolt-Base candidate pool, TimesFM and Chronos-Bolt Mini retain strong window-level rank agreement ($\rho = 0.828$ in Figure 4(b)). Because dataset-stratified selection depends on ranks rather than absolute loss magnitudes, this monotonic agreement is sufficient for TimesFM to identify a useful Chronos subset, consistent with the successful Chronos transfer summarized in Section 4.3.

The Moirai pool illustrates a limitation of this transfer. TimesFM and Moirai-Small obtain only weak rank agreement ($\rho = 0.254$ in Figure 4(c)), and the TimesFM-selected subset underperforms both the Moirai-Small subset and full-data pretraining. A plausible explanation is that TimesFM scores point and quantile errors, whereas Moirai uses a distributional negative log-likelihood, so the two references need not assign the same difficulty to high-uncertainty windows.

F Qualitative Analysis

We qualitatively inspect the Chronos-Bolt Mini RFL scores used for dataset-stratified selection. The score file contains all candidate training windows from the original sample cache. For visualization only, we form three global score regions: low RFL (bottom 1%), middle RFL (49.5–50.5%), and high RFL (top 1%). Selection in the main experiments remains dataset-stratified and the global bands here are used only to make the score semantics visually interpretable.

Figure 5 supports the intended interpretation of RFL as a difficulty coordinate. Low-score samples tend to have small residual signal after context normalization. Middle-score samples include visible trend, seasonality, or amplitude changes but remain connected to the context. High-score samples frequently show future behavior that is hard to extrapolate from the context alone, such as sudden jumps or large tail events. This motivate removing both extremes when constructing a compact pretraining subset: low-score windows can be redundant, while the high-score tail mixes rare events with weakly learnable windows.

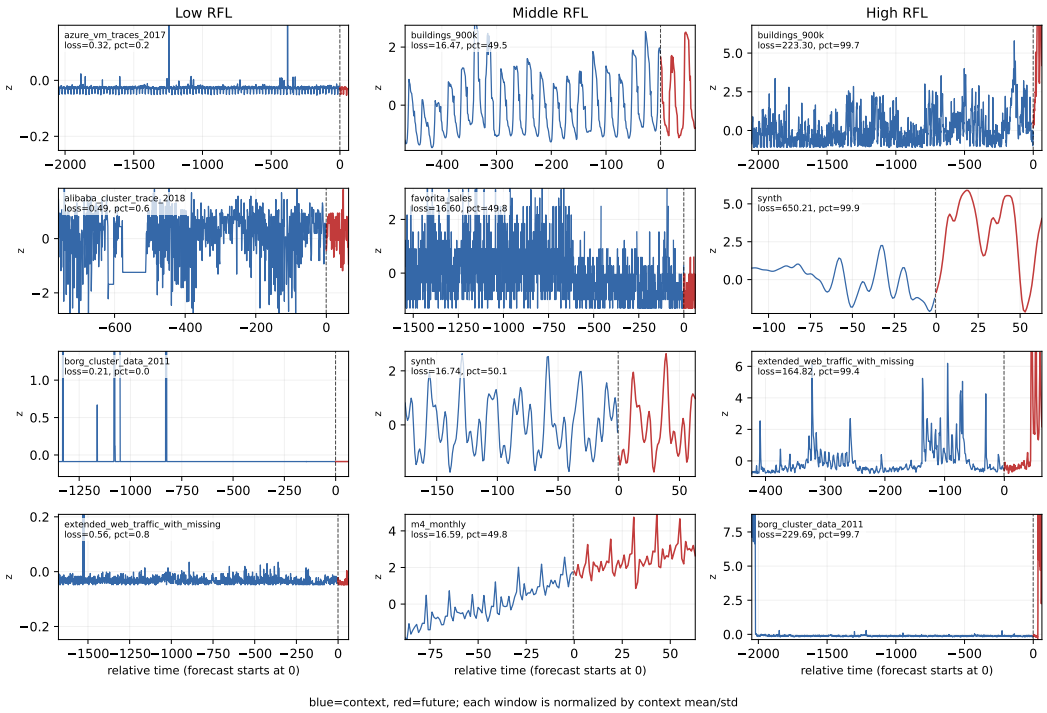


Figure 5: Representative windows from low, middle, and high Chronos-Bolt Mini RFL regions. Blue denotes the context and red denotes the forecast horizon. Each window is standardized by its valid context mean and standard deviation. The `pct` value in each panel is the global empirical percentile rank of that window's RFL score among all candidate windows. Low-RFL windows are often highly regular, middle-RFL windows retain learnable nontrivial structure, and high-RFL windows often contain abrupt shifts, extreme future values, or context–future mismatch.