PHaSe: ADAPTIVE TIME SERIES DATA SELECTION VIA PROTOTYPE HARDNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world time series data vary in quality, yet conventional training typically grants equal training opportunities to all instances. Although data selection can prioritize useful instances, assessing instances individually does not explicitly account for the overall learning state of their shared temporal patterns. In this work, we propose Prototype HArdness-guided SElection (PHaSe), an adaptive time series data selection framework that aligns training allocation with the model's evolving learning needs across temporal patterns. PHaSe organizes instances into representative pattern groups and estimates their relative learning needs, termed hardness, from recent loss trajectories, providing a pattern-level basis for identifying instances that require further learning. It then transfers pattern-level hardness to instance priorities, modulates these priorities by assignment concentration, and converts them into sampling probabilities. To adapt pattern organization to changing training priorities, we refit prototypes on selected instances and combine the fitted centers with historical centers through normalized momentum updates. Compared with the evaluated baselines in individual settings, PHaSe achieves up to 16.4% lower long-term MSE, 4.8% lower short-term sMAPE, and 74.2% higher classification accuracy, while using only half of the training instances per epoch.

1 INTRODUCTION

Time series analysis is widely applied in energy, transportation, and healthcare (Lim et al., 2019; Li et al., 2017; Che et al., 2016). Existing research has made sustained progress in architectural design and feature modeling, focusing on how to represent the data (Zeng et al., 2022; Nie et al., 2023; Wu et al., 2023; Liu et al., 2024). Real-world time series vary in quality, yet conventional training typically grants equal training opportunities to all instances without fully accounting for their different roles in model learning, potentially leading to less effective use of training resources (Taga et al., 2026; Katharopoulos & Fleuret, 2018). More importantly, a model's mastery of different instances evolves throughout training (Zhou et al., 2020; Toneva et al., 2018; Swayamdipta et al., 2020). An instance that is currently difficult to predict may be predicted more accurately after further training. Even if an instance is not selected for training in the current epoch, model updates can still change its prediction. This motivates selecting data according to the model's current state.

Existing approaches commonly guide data selection by assessing the role of individual instances in model learning. For example, TimeInf uses influence functions to quantify the contribution of time series data to model predictions (Zhang et al., 2025), while AdaRho uses a dynamically updated reference model to guide selection based on reducible loss (Taga et al., 2026). These studies highlight the value of allocating training data according to model state. From the perspective of learning objectives, training should not only fit individual instances but, more importantly, capture their shared temporal patterns to support generalization. Differences in pattern frequency, temporal characteristics, and compatibility with the model mean that different patterns may be learned at different rates. As illustrated in Figure 1, full-data training allocates equal training opportunities, whereas instance-level selection assigns priorities to individual instances. We instead use the learning state shared by related instances to guide selection at the pattern level. The key challenges are to characterize the model's learning state across temporal patterns and to translate this pattern-level assessment into instance selection criteria.

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.

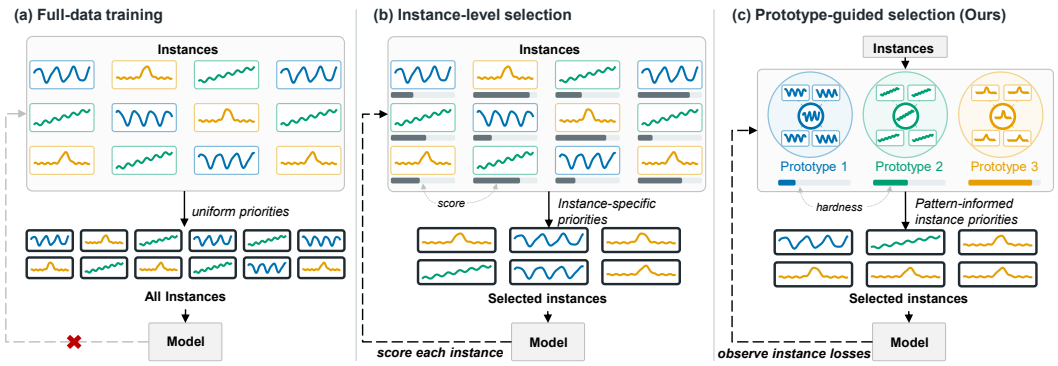

Figure 1: Training data allocation. (a) Full-data training uses all instances. (b) Instance-level selection scores instances individually. (c) PHaSe aggregates learning states through prototypes and transfers prototype hardness to instance priorities.

To address these challenges, we propose **PHaSe** (**P**rototype **Ha**rdness-guided **Se**lection), an adaptive data selection framework that responds to the model's evolving learning needs across temporal patterns. Clustering backbone-independent descriptors organizes related instances into representative pattern groups. Their prototype hardness is estimated from loss trajectories as a model-dependent, time-varying measure of relative learning needs, not intrinsic data difficulty (Seedat et al., 2024). Because selection determines the next epoch's updates, we use linear trends to distinguish instances with similar current errors but different trajectories. Soft memberships aggregate these signals into prototype hardness and transfer it back to instance priorities. Assignment concentration modulates this transfer, giving a pattern's learning state greater influence when the association is clearer. The resulting sampling distribution selects the next training subset under a fixed budget.

Prototypes determine which instances' learning states are pooled and thus influence training allocation. To adapt this organization to the training focus, PHaSe refits prototypes on the instances used for model updates. Historical-center initialization and normalized momentum balance adaptation to the current subset with retention of previous pattern information. Meanwhile, end-of-epoch observation refreshes losses for all training instances, including unselected ones. Prototype updates therefore reflect actual training allocation, while hardness estimation retains full-data coverage. Our contributions include:

- We propose a data selection framework guided by prototype hardness that dynamically allocates training opportunities according to the model's learning needs across temporal patterns.
- We develop a mechanism for estimating and transferring prototype hardness from loss trajectories, quantifying the model's mastery of temporal patterns and translating this assessment into instance sampling priorities.
- We propose a dynamic prototype update mechanism linked to learning needs, retaining historical pattern information while adapting to the current training focus.

2 Related Work

This section focuses on data selection and adaptive training. Broader work on time series modeling, foundation models, and data valuation is reviewed in Appendix A.

Data selection prioritizes training instances according to their estimated utility. For time series, TimeInf uses influence functions to attribute model predictions to training data (Zhang et al., 2025), while TSRating learns a cross-domain quality rating model from pairwise LLM judgments (Wu et al., 2025). Rather than estimating data quality alone, PHaSe conditions selection on the model's changing learning needs across shared temporal patterns.

Adaptive training accounts for changes in instance relevance during optimization. DIHCL constructs curricula from historical instance hardness (Zhou et al., 2020), and InfoBatch dynamically prunes

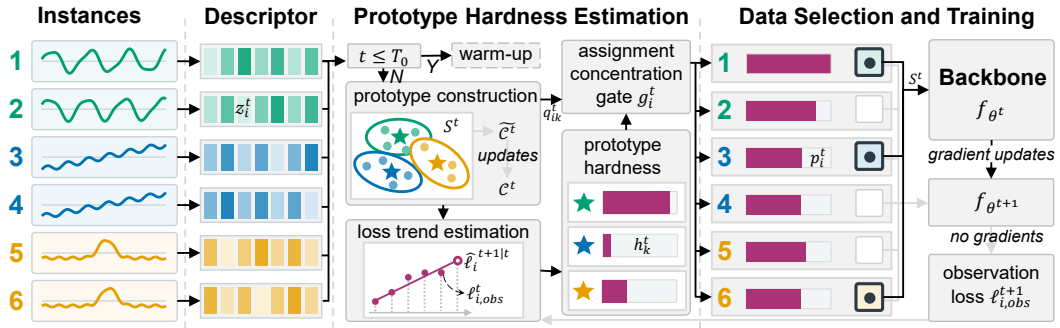

Figure 2: Overview of PHaSe. Loss trends are aggregated into prototype hardness and transferred to instance priorities. Selected instances update the backbone, while full-data observation refreshes losses for the next selection round.

instances using loss-based scores and rescales their training contributions (Qin et al., 2023). RHO-LOSS prioritizes reducible loss (Mindermann et al., 2022), while AdaRho updates a reference model to support selection for time series (Taga et al., 2026). These approaches establish the value of model-dependent selection. PHaSe organizes the selection signal at the pattern level, aggregating loss trends into prototype hardness and transferring it back to instance priorities. Prototypes adapt to the selected training data, while full-data loss observations refresh the learning states used for subsequent selection.

3 Method

3.1 Framework and Initialization

Framework overview. Let $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{N}$ contain time series inputs and their targets. For forecasting, $X_i \in \mathbb{R}^{L\times C}$ and $Y_i \in \mathbb{R}^{H\times C}$. For classification, Y_i is a class label. A backbone f_θ outputs a forecast or class logits. Selection operates on complete instances using scalar losses defined below for both tasks. Appendix B.1 gives their explicit forms and variable length handling.

A model's mastery varies across temporal patterns and evolves throughout training. PHaSe therefore allocates training opportunities according to the model's learning state. As shown in Figure 2, we construct temporal pattern prototypes, estimate their hardness from observation loss trends, and use this hardness to select instances for model updates. The prototypes and their hardness are refreshed throughout training to guide subsequent selection.

Let θ^t be the model parameters after epoch t, S^t the indices of instances used for model updates in that epoch, and p^t the sampling weights prepared at its end. During T_0 warm-up epochs, $S^t = \{1, \dots, N\}$. Starting at epoch $T_0 + 1$, a new set S^t of $M = \max\{1, \mathrm{round}(rN)\}$ distinct instance indices is sampled without replacement using p^{t-1}, where $r \in (0, 1]$ controls the update budget. At the end of every epoch, all instances are observed with f_{θ^t}, after which the scheduler prepares p^t for the next epoch. Algorithm 1 gives the complete ordering.

Warm-up and prototype initialization. Full data warm-up for T_0 epochs establishes a learning state baseline before selective training. Each instance receives a backbone independent descriptor $d_i = \phi(X_i)$ that captures temporal profiles, local variations, and magnitude statistics (Appendix B.3). To make descriptor distances comparable, we standardize each coordinate and apply ℓ_2 normalization:

$$\widetilde{d}_i^t = \frac{d_i - \bar{d}^t}{\max\{\sigma_d^t, \epsilon\}}, \qquad z_i^t = \frac{\widetilde{d}_i^t}{\max\{\|\widetilde{d}_i^t\|_2, \epsilon\}}. \tag{1}$$

Here, $\bar{d}^t$ and σ_d^t are the coordinate-wise mean and population standard deviation of descriptors indexed by S^t, and $\epsilon > 0$ ensures numerical stability.

We apply k-means (Lloyd, 1982) to $\mathcal{Z}^t = \{z_i^t\}_{i \in S^t}$ to organize instances with similar input characteristics into temporal patterns. The fitted centers are normalized to form the prototype set:

$$c_k^t = \frac{\widetilde{c}_k^t}{\max\{\|\widetilde{c}_k^t\|_2, \epsilon\}}, \qquad k = 1, \ldots, K, \tag{2}$$

where $\widetilde{c}_k^t$ is the fitted center and the resulting prototype set is $\mathcal{C}^t = \{c_k^t\}_{k=1}^K$.

To associate instance-level learning states with temporal patterns, we compute soft memberships for all training instances:

$$q_{ik}^t = \frac{\exp\big(-\|z_i^t - c_k^t\|_2^2/\tau_a\big)}{\sum_{j=1}^K \exp\big(-\|z_i^t - c_j^t\|_2^2/\tau_a\big)}, \qquad \sum_{k=1}^K q_{ik}^t = 1. \tag{3}$$

The temperature $\tau_a > 0$ controls membership concentration. Soft memberships allow an instance to be associated with multiple temporal patterns while preserving the strength of each association.

3.2 Prototype Hardness Estimation

Prototype hardness h_k^t is a model dependent, time varying score of relative learning needs, estimated from the observation losses of associated instances. Higher hardness indicates a larger trend-adjusted loss signal relative to other patterns. Because selection allocates updates for the next epoch, we use recent loss trends to distinguish instances with similar current errors but different learning trajectories.

Loss trend estimation. Let $\widehat{Y}_i = f_\theta(X_i)$. Define the training loss $\ell_{i,\mathrm{tr}}$ and assessment error e_i for each task as

$$\ell_{i,\mathrm{tr}}(\theta) = e_i(\theta) = \begin{cases} \mathrm{MSE}(\widehat{Y}_i, Y_i), & \text{forecasting}, \\ \mathrm{CE}(\widehat{Y}_i, Y_i), & \text{classification}. \end{cases} \tag{4}$$

MSE averages over target entries and CE denotes cross entropy. The observation loss uses the same transformation for both tasks:

$$\ell_{i,\mathrm{obs}}^t = \log\big(1 + e_i(\theta^t)\big)\,. \tag{5}$$

We retain the latest $m = \min\{t, W\}$ observations, where W is the maximum history length in epochs, and write them chronologically as $u_{ij} = \ell_{i,\mathrm{obs}}^{t-m+1+j}$ for $j = 0, \ldots, m-1$. With $\bar{j} = (m-1)/2$ and $\bar{u}_i = m^{-1}\sum_j u_{ij}$, we estimate a linear slope and combine it with the historical mean to form a loss-trend signal $\widehat{\ell}_i^{\,t+1|t}$ for the next selection round:

$$b_i^t = \frac{\sum_{j=0}^{m-1}(j - \bar{j})(u_{ij} - \bar{u}_i)}{\max\Big\{\sum_{j=0}^{m-1}(j-\bar{j})^2, \epsilon\Big\}}, \qquad \widehat{\ell}_i^{\,t+1|t} = \max\{0, \bar{u}_i + m b_i^t\}. \tag{6}$$

For $m = 1$, the slope is zero. The term $m b_i^t$ scales the per-epoch slope by the available history length. A decreasing trend lowers the signal, an increasing trend raises it, and a flat trend leaves it unchanged.

Prototype hardness aggregation. We aggregate the loss-trend signals through soft memberships to obtain a pattern-level state:

$$a_k^t = \frac{\sum_{i=1}^N q_{ik}^t \widehat{\ell}_i^{\,t+1|t}}{\max\{\sum_{i=1}^N q_{ik}^t, \epsilon\}}, \qquad h_k^t = \exp\left[\mathrm{clip}\left(\frac{a_k^t - \bar{a}^t}{\tau_h \max\{\sigma_a^t, \epsilon\}}, -\alpha, \alpha\right)\right], \tag{7}$$

where $\bar{a}^t$ and σ_a^t are the mean and population standard deviation of the K aggregated loss signals a_k^t, and $\tau_h > 0$ is a hardness temperature. $\alpha > 0$ bounds the exponential input within $[-\alpha, \alpha]$. Exponentiation converts standardized loss states into hardness scores, and clipping limits amplification of extreme estimates.

Although only S^t determines the fitted centers, all instances contribute to Equation (7). This distinction lets the method adapt its pattern representation to the training subset without restricting the estimation of pattern-level learning demand to that subset. Aggregation also shares trajectory information among related inputs, so selection is informed by their common pattern-level state rather than solely by isolated instance scores.

3.3 HARDNESS-GUIDED DATA SELECTION AND TRAINING

Instance prioritization and selection. We prioritize instances associated with high hardness prototypes to focus model updates on insufficiently learned temporal patterns, rather than relying solely on individual instance losses. Prototype hardness is transferred through the same soft memberships. A clearer association makes a pattern's learning state a more specific basis for instance priority, so we modulate this transfer by assignment concentration:

$$E_i^t = -\sum_{k=1}^{K} q_{ik}^t \log q_{ik}^t, \qquad g_i^t = \max\left\{g_{\min}, 1 - \lambda \frac{E_i^t}{\log K}\right\}, \qquad v_i^t = g_i^t \sum_{k=1}^{K} q_{ik}^t h_k^t. \tag{8}$$

Here $K \geq 2$, E_i^t is membership entropy, and $\lambda \in [0, 1]$ controls the assignment concentration gate g_i^t. Its lower bound $g_{\min} \in (0, 1]$ preserves positive priorities. The resulting v_i^t is the unnormalized sampling priority for the next epoch.

To balance targeted learning with coverage of the training population, we mix the hardness-guided sampling distribution with a uniform component. This preserves selection opportunities for lower scored instances and limits excessive concentration on those currently assigned high priorities:

$$\widetilde{p}_i^t = \frac{(v_i^t)^{1/\tau_s}}{\sum_{j=1}^{N} (v_j^t)^{1/\tau_s}}, \qquad p_i^t = \frac{1-\rho}{N} + \rho \widetilde{p}_i^t. \tag{9}$$

The sampling temperature $\tau_s > 0$ controls concentration. The mixing coefficient $\rho \in [0, 1)$ controls the emphasis on hardness-guided prioritization relative to uniform exploration. We sample M instances without replacement according to the weights p^t and denote their index set by S^{t+1}. This allocates the update budget across distinct instances within each epoch. The selected instances are then shuffled and grouped into minibatches for training.

Selective optimization and global observation. Prioritized instances receive more opportunities for model updates. To moderate the resulting concentration of training influence, we adjust their loss contributions using tempered inverse sampling weights. Clipping limits extreme inverse weights, while normalization maintains a mean weight of one within the current minibatch. For a minibatch $\mathcal{B} \subseteq S^{t+1}$, we define inverse sampling weights as:

$$\widetilde{w}_i^t = \min\{w_{\max}, (N p_i^t)^{-\beta}\}, \qquad w_i^t(\mathcal{B}) = \frac{\widetilde{w}_i^t}{|\mathcal{B}|^{-1} \sum_{j \in \mathcal{B}} \widetilde{w}_j^t}, \tag{10}$$

where $w_{\max} \geq 1$ caps the weights before normalization, and $\beta \in [0, 1)$ controls the adjustment strength. $\beta = 0$ recovers unweighted training. Sampling weights remain fixed throughout the epoch, and no gradients propagate through the selection mechanism. The loss function is:

$$\mathcal{L}_{\mathcal{B}}^{t+1}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} w_i^t(\mathcal{B}) \ell_{i,\mathrm{tr}}(\theta), \tag{11}$$

where $\ell_{i,\mathrm{tr}}$ is defined in Equation (4). Model updates can change predictions on both selected and unselected instances. To avoid outdated learning states, we evaluate $f_{\theta^{t+1}}$ on all training instances at the end of the epoch using the task specific observation loss. This evaluation uses no gradients and does not update model parameters. Selective training thus allocates gradient updates, while global observation refreshes the information used for subsequent selection.

3.4 DYNAMIC PROTOTYPE UPDATES

Training conditioned clustering. Prototypes organize learning states for selection rather than merely summarize input geometry. To make this organization responsive to training priorities, we refit prototypes on the instances trained on in the current epoch, while retaining all instance hardness estimation. At the end of epoch $t > T_0$, we recompute descriptor statistics from S^t, transform all instances using Equation (1), and fit k-means only on $\{z_i^t\}_{i \in S^t}$.

We initialize the fit with $\mathcal{C}^{t-1}$ so that centers adapt from their previous positions rather than a new random configuration. This encourages continuity across fits and provides the starting correspondence for blending same index centers.

Momentum update. The selected subset reflects training priorities but also finite budget sampling fluctuations. Directly adopting its fitted centers could propagate these fluctuations through hardness aggregation into subsequent selection. We therefore blend fitted and historical centers, balancing adaptation with continuity, and normalize the result for membership computation:

$$\xi_k^t = \mu c_k^{t-1} + (1-\mu)\widetilde{c}_k^t, \qquad c_k^t = \frac{\xi_k^t}{\max\{\|\xi_k^t\|_2, \epsilon\}}, \tag{12}$$

where $\mu \in [0, 1)$ controls retention of historical centers. Appendix B.8 gives a local regularization interpretation of this update. We then recompute all instance memberships and hardness to prepare p^t for the next epoch.

4 Experiments

To examine whether pattern-level learning states provide a useful basis for allocating training instances, we evaluate PHaSe across long-term forecasting, short-term forecasting, and classification.

4.1 Experimental Setup

Datasets. For long-term forecasting, we use ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, and Traffic. For short-term forecasting, we use the Yearly, Monthly, and Daily subsets of M4 (Makridakis et al., 2018). For classification, we use JapaneseVowels, Handwriting, Heartbeat, and Ethanol Concentration from the UEA archive (Bagnall et al., 2018). Dataset descriptions and preprocessing are given in Appendix C.

Backbones and comparison methods. We use DLinear (Zeng et al., 2022), TimesNet (Wu et al., 2023), and iTransformer (Liu et al., 2024) to cover linear, convolutional, and Transformer-based architectures. We compare PHaSe with **TimeInf** (Zhang et al., 2025), which uses influence-based data valuation, and **AdaRho** (Taga et al., 2026), which selects data via target minus reference losses and updates its reference online. We also include **Full**, the reproduced backbone trained on all instances, and **Random**, uniform instance sampling without replacement.

Metrics and implementation details. We assess data selection through downstream performance, using mean squared error (MSE) for long-term forecasting, symmetric mean absolute percentage error (sMAPE) for short-term forecasting, and accuracy (Acc.) for classification. Lower forecasting errors and higher accuracy indicate better performance. Experiments use PyTorch (Paszke et al., 2019) on two NVIDIA H800 GPUs with 80 GB of memory each. The backbone configuration, optimizer, batch size, learning rate schedule, maximum epoch count, and early stopping patience are shared for each combination of dataset and backbone. Training settings and external method adaptations are detailed in Appendix D.

4.2 Main Results

Table 1 compares PHaSe with Full, Random, TimeInf, and AdaRho across three time series tasks using DLinear, TimesNet, and iTransformer. These backbones cover linear, CNN-based, and Transformer-based architectures, allowing us to examine whether the benefits of PHaSe extend across different model families.

Long-term forecasting. PHaSe achieves the lowest MSE in all 21 combinations of dataset and backbone, including one tie with AdaRho on Electricity using iTransformer. It consistently improves over both full data training and random sampling across all seven datasets. For example, on ETTh2 with DLinear, PHaSe reduces MSE from 0.348 to 0.301 compared with Full, a relative reduction of 13.5%, and outperforms the strongest competing method, AdaRho, by 5.6%. These results support the effectiveness of using pattern-level learning states to guide training instance allocation across different forecasting architectures.

Short-term forecasting. On the M4 subsets, PHaSe achieves the lowest sMAPE in seven of the nine settings and ranks second in the remaining two. With TimesNet, it outperforms all comparison

Table 1: Comparison of data selection methods across forecasting and classification on multiple datasets and backbones. All results are averaged over five seeds. MSE and sMAPE are lower-is-better, and accuracy is in percent and higher-is-better. Long-term input/horizon are 96/96, and M4 horizons are 6/18/14. JV, HW, HB, and EC denote JapaneseVowels, Handwriting, Heartbeat, and Ethanol Concentration.

	Long-term forecasting: MSE ↓							M4: sMAPE ↓			Classification: Acc. ↑			
Method	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Elec.	Traffic	Yearly	Monthly	Daily	JV	HW	HB	EC
DLinear														
Full	0.396	0.348	0.343	0.194	0.196	0.195	0.650	14.06	14.13	3.07	96.94	12.47	73.33	**29.28**
Random	0.405	0.359	0.345	0.195	0.196	0.195	0.649	14.06	14.13	3.00	96.94	12.31	73.33	27.63
TimeInf	0.414	0.360	0.346	0.196	0.197	0.198	0.653	14.06	13.90	3.00	95.77	8.94	73.33	27.88
AdaRho	0.412	0.319	0.347	0.183	0.199	0.196	0.662	**14.00**	14.08	3.02	96.85	14.71	73.82	27.38
PHaSe	**0.395**	**0.301**	**0.331**	**0.182**	**0.182**	**0.193**	**0.641**	14.02	**13.85**	**2.96**	**96.96**	**15.57**	**73.93**	28.62
TimesNet														
Full	0.412	0.321	0.336	0.187	0.175	0.167	0.590	13.60	13.09	3.09	**95.68**	18.63	72.36	27.50
Random	0.416	0.318	0.342	0.186	0.175	0.167	0.591	13.58	13.08	3.10	95.23	17.96	75.60	27.50
TimeInf	0.431	0.338	0.340	0.187	0.177	0.170	0.597	13.70	13.08	3.10	92.79	15.73	71.38	26.36
AdaRho	0.411	0.315	0.330	0.177	0.172	0.163	0.596	13.63	13.07	3.09	95.32	20.82	75.61	26.62
PHaSe	**0.408**	**0.310**	**0.328**	**0.170**	**0.166**	**0.160**	**0.586**	**13.54**	**13.00**	**3.05**	95.50	**21.76**	**75.65**	**28.14**
iTransformer														
Full	0.393	0.299	0.342	0.185	0.175	0.148	0.393	13.85	14.20	3.50	97.36	19.25	71.22	27.63
Random	0.395	0.299	0.343	0.185	0.176	0.149	0.398	13.83	14.20	3.52	97.21	17.02	73.66	25.86
TimeInf	0.408	0.300	0.350	0.184	0.177	0.153	0.410	13.83	**14.04**	3.52	94.77	12.24	73.17	**27.76**
AdaRho	0.393	0.298	0.342	0.178	0.176	**0.147**	0.401	13.83	14.19	3.51	96.76	19.53	73.98	25.73
PHaSe	**0.380**	**0.296**	**0.326**	**0.176**	**0.169**	**0.147**	**0.390**	**13.77**	14.13	**3.35**	**97.40**	**20.43**	**74.85**	27.74

Table 2: Training efficiency on ETTh1 and Weather with input and prediction lengths of 96. We report mean ± standard deviation over three seeds, in seconds. Lower is better.

Dataset	Backbone	Full	Random	TimeInf	AdaRho	PHaSe
ETTh1	DLinear	**17.0 ± 0.1**	**17.0 ± 0.4**	17.7 ± 0.2	95.0 ± 11.8	17.5 ± 0.2
	TimesNet	110.8 ± 0.8	82.8 ± 0.6	84.4 ± 0.3	232.3 ± 0.8	**82.4 ± 0.3**
	iTransformer	32.0 ± 0.1	**27.2 ± 0.1**	29.1 ± 0.3	121.1 ± 0.6	27.9 ± 0.1
Weather	DLinear	74.3 ± 20.2	56.4 ± 19.1	56.2 ± 15.6	559.7 ± 47.7	**35.7 ± 0.2**
	TimesNet	225.4 ± 23.9	215.1 ± 76.7	215.5 ± 23.3	739.2 ± 200.7	**212.9 ± 29.9**
	iTransformer	90.8 ± 1.5	76.9 ± 0.5	77.1 ± 8.2	396.7 ± 57.4	**65.9 ± 5.0**

methods on Yearly, Monthly, and Daily data. On M4-Daily with iTransformer, PHaSe reduces sMAPE from 3.50 under Full to 3.35. These results demonstrate its applicability across different temporal granularities.

Classification. PHaSe achieves the highest accuracy in nine of the twelve settings, with three wins for each backbone. It performs best on both Handwriting and Heartbeat across all three architectures. On Handwriting, PHaSe improves accuracy over Full by 3.10, 3.13, and 1.18 percentage points with DLinear, TimesNet, and iTransformer, respectively. Overall, these results show that the benefits of PHaSe extend beyond forecasting to classification.

Training efficiency. Table 2 reports training time on ETTh1 and Weather. PHaSe achieves the lowest mean training time in four of the six settings and is faster than TimeInf and AdaRho in every setting. Compared with Full, it reduces training time in five settings, with reductions ranging from 5.5% to 52.0%. The largest reduction occurs on Weather with DLinear, where training time decreases from 74.3 to 35.7 seconds. On ETTh1 with DLinear, PHaSe takes slightly longer than Full, at 17.5 versus 17.0 seconds. Across all six settings, PHaSe achieves a speedup factor ranging from 2.8 to 15.7 over AdaRho. Together with the predictive results in Table 1, these findings indicate that PHaSe can improve model performance while reducing training time in most evaluated settings.

Table 3: Ablation study of PHaSe across long-term forecasting, short-term forecasting, and classification. Results are reported as mean ± standard deviation over five seeds, after equal-weight averaging across the datasets and three backbones within each task. Accuracy is reported in percent. The best mean in each column is bolded.

Variant	Long-term MSE ↓	Long-term MAE ↓	Short-term sMAPE ↓	Classification Acc. ↑
Random	0.3087 ± 0.0012	0.3240 ± 0.0007	10.277 ± 0.023	53.35 ± 0.63
w/o assignment gating	0.2980 ± 0.0006	0.3082 ± 0.0004	10.190 ± 0.047	54.55 ± 0.46
w/o loss trend estimation	0.3004 ± 0.0004	0.3116 ± 0.0003	10.220 ± 0.032	54.19 ± 0.35
w/o momentum smoothing	0.3012 ± 0.0008	0.3130 ± 0.0006	10.214 ± 0.058	54.28 ± 0.42
w/o prototype updates	0.3033 ± 0.0004	0.3167 ± 0.0003	10.241 ± 0.035	53.92 ± 0.37
w/o prototype aggregation	0.3100 ± 0.0011	0.3253 ± 0.0008	10.283 ± 0.064	53.20 ± 0.58
PHaSe	**0.2970 ± 0.0005**	**0.3076 ± 0.0003**	**10.186 ± 0.051**	**54.71 ± 0.43**

4.3 Analysis

Ablation study. We evaluate the contributions of PHaSe's components by removing assignment gating, loss trend estimation, momentum smoothing, prototype updates, or prototype aggregation. As shown in Table 3, the full framework achieves the best mean performance on all four metrics across the three tasks. Removing prototype aggregation leads to the largest degradation, increasing long-term MSE from 0.2970 to 0.3100 and reducing classification accuracy from 54.71% to 53.20%. This comparison supports sharing learning-state information among instances with related temporal patterns. Because this variant also removes the associated assignment gate, it evaluates the prototype-guided mechanism as a whole.

Dynamic prototype maintenance also contributes to performance. Freezing the prototypes increases long-term MSE to 0.3033 and reduces classification accuracy to 53.92%, suggesting that pattern groups benefit from adapting to changes in training allocation. Removing momentum smoothing worsens all four mean metrics, supporting the use of historical information when updating prototypes from the current selected instances. Replacing the loss-trend signal with the latest observation loss increases long-term MSE to 0.3004 and short-term sMAPE to 10.220. These results support incorporating recent loss trends rather than relying only on the current loss. Assignment gating provides smaller improvements. Removing it changes short-term sMAPE from 10.186 to 10.190 and classification accuracy from 54.71% to 54.55%.

Data pruning. To assess the importance of prioritized instances, we remove them in descending score order and retrain iTransformer on the remaining data, increasing the removal ratio from 0% to 45% in steps of 5%. Figure 3 shows greater degradation under PHaSe, particularly at larger removal ratios. At 45% removal, Electricity MSE rises from 0.148 to 0.178, compared with 0.153, 0.170, and 0.162 for Random, TimeInf, and AdaRho. M4-Daily sMAPE increases from 3.50 to 4.10, exceeding all comparison methods. JapaneseVowels accuracy falls from 97.36% to 65.42%, a drop of 31.94 percentage points, while the other methods retain 68.50% to 71.80% accuracy. These results support PHaSe's ability to prioritize instances whose removal has a larger impact on downstream performance.

Finetuning time series foundation models. To examine whether the benefits of PHaSe extend to foundation model adaptation, we evaluate Time-MoE (Shi et al., 2024), Moirai-small (Woo et al., 2024), and MOMENT-small (Goswami et al., 2024) on ETTh1 and Weather. We compare finetuning with PHaSe against full-data finetuning and random instance selection, and include zero-shot performance as a reference.

Figure 4 shows that PHaSe achieves the lowest MSE in all six settings. Compared with full-data finetuning, it reduces Time-MoE's MSE from 0.3893 to 0.3651 on ETTh1 and from 0.1740 to 0.1634 on Weather, improvements of 6.2% and 6.1%. The largest reduction is 7.8% for Moirai-small on ETTh1, from 0.4126 to 0.3805. MOMENT-small improves by 3.0% and 1.1% on ETTh1 and Weather. Random selection performs similarly to Full, whereas PHaSe improves on both throughout, supporting its applicability to the evaluated foundation models.

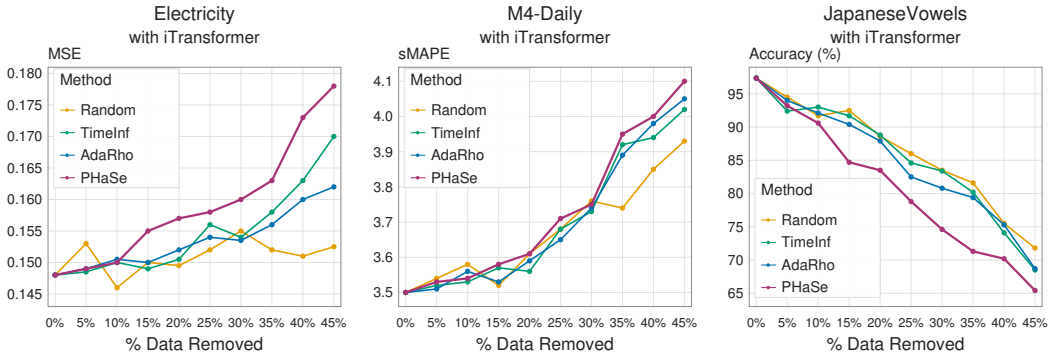

Figure 3: Data pruning with iTransformer. Instances are removed in descending score order, or uniformly for Random, before retraining. Higher forecasting errors and lower classification accuracy indicate greater degradation.

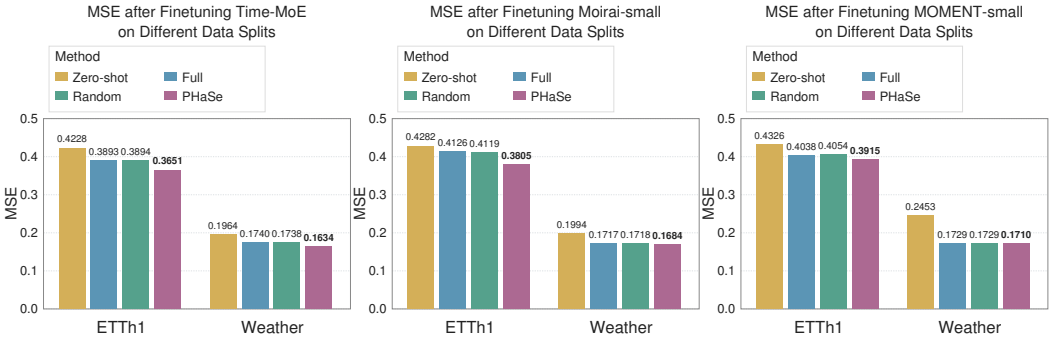

Figure 4: MSE of Time-MoE, Moirai-small, and MOMENT-small on ETTh1 and Weather under zero-shot inference, Full, Random, and PHaSe finetuning.

5 Limitations

Despite its effectiveness across the evaluated tasks, PHaSe leaves room for improving the efficiency and precision of learning-state assessment. Refreshing losses over the full training set and maintaining prototypes introduce additional overhead, so runtime savings depend on the backbone and sampling budget. Moreover, loss-based hardness reflects relative learning needs rather than directly measuring the benefit of additional updates, and may assign high priority to instances with persistent noise. A promising direction is to combine less frequent or incremental state updates with noise-aware hardness estimation, reducing assessment costs while better identifying instances that can benefit from further training.

6 Conclusion

We present PHaSe, an adaptive time series data selection framework that allocates training opportunities according to evolving pattern-level learning needs. It estimates prototype hardness from loss trajectories and transfers these signals to instance priorities. Dynamic prototype updates balance current training focus with historical information, while global observation refreshes learning states across the training set. Experiments on forecasting, classification, and foundation model finetuning support the effectiveness of this pattern-level approach across different backbones.

AI USE STATEMENT

We used AI for manuscript polishing and translation, and to assist with literature review, experimental design and analysis, and figure preparation. The authors take responsibility for the final text, code, results, and claims.

REFERENCES

David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, 2007. URL https://research.google/pubs/k-means-the-advantages-of-careful-seeding/.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive, 2018, 2018. URL https://arxiv.org/abs/1811.00075.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent Neural Networks for Multivariate Time Series with Missing Values, 2016. URL https://arxiv.org/abs/1606.01865.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2242–2251. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ghorbani19c.html.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16115–16152. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/goswami24a.html.

Angelos Katharopoulos and Francois Fleuret. Not All Samples Are Created Equal: Deep Learning with Importance Sampling. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2525–2534. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/katharopoulos18a.html.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based Data Subset Selection for Efficient and Robust Learning, 2020. URL https://arxiv.org/abs/2012.10630.

Krishnateja Killamsetty, Durga S, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5464–5474. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/killamsetty21a.html.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting, 2017. URL https://arxiv.org/abs/1707.01926.

Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting, 2019. URL https://arxiv.org/abs/1912.09363.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting, 2024. URL https://arxiv.org/abs/2310.06625.

S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2): 129–137, March 1982. ISSN 1557-9654. doi: 10.1109/tit.1982.1056489. URL http://dx.doi.org/10.1109/TIT.1982.1056489.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4): 802–808, October 2018. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2018.06.001. URL http://dx.doi.org/10.1016/j.ijforecast.2018.06.001.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15630–15649. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/mindermann22a.html.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6950–6960. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/mirzasoleiman20a.html.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers, 2023. URL https://arxiv.org/abs/2211.14730.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019. URL https://arxiv.org/abs/1912.01703.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep Learning on a Data Diet: Finding Important Examples Early in Training, 2021. URL https://arxiv.org/abs/2107.07075.

Garima Pruthi, Frederick Liu, Mukund Sundararajan, and Satyen Kale. Estimating Training Data Influence by Tracing Gradient Descent, 2020. URL https://arxiv.org/abs/2002.08484.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, and Yang You. InfoBatch: Lossless Training Speed Up by Unbiased Dynamic Data Pruning, 2023. URL https://arxiv.org/abs/2303.04947.

Nabeel Seedat, Fergus Imrie, and Mihaela van der Schaar. Dissecting sample hardness: A fine-grained analysis of hardness characterization methods for data-centric ai. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 9495–9564, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/285918fa156e68e988b537efd836bdba-Paper-Conference.pdf.

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, 2024. URL https://arxiv.org/abs/2409.16040.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning, 2022. URL https://arxiv.org/abs/2206.14486.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics, 2020. URL https://arxiv.org/abs/2009.10795.

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 Emtiyaz Khan, Yingzhen Li, Arno Solin, and Aaditya Ramdas (eds.), *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 1081–1089. PMLR, 02–05 May 2026. URL https://proceedings.mlr.press/v300/taga26a.html.

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, 2018. URL https://arxiv.org/abs/1812.05159.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting, 2021. URL https://arxiv.org/abs/2106.13008.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis, 2023. URL https://arxiv.org/abs/2210.02186.

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, 2025. URL https://arxiv.org/abs/2506.01290.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. TS2Vec: Towards Universal Representation of Time Series, 2021. URL https://arxiv.org/abs/2106.10466.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are Transformers Effective for Time Series Forecasting?, 2022. URL https://arxiv.org/abs/2205.13504.

Yizi Zhang, Jingyan Shen, Xiaoxue Xiong, and Yongchan Kwon. TimeInf: Time Series Data Contribution via Influence Functions, 2025. URL https://arxiv.org/abs/2407.15247.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

Tianyi Zhou, Shengjie Wang, and Jeffrey Bilmes. Curriculum learning by dynamic instance hardness. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8602–8613. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/62000dee5a05a6a71de3a6127a68778a-Paper.pdf.