LEARNING FROM COMPLEXITY: EXPLORING DYNAMIC SAMPLE PRUNING OF SPATIO-TEMPORAL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatio-temporal forecasting is fundamental to intelligent systems in transportation, climate science, and urban planning. However, training deep learning models on the massive, often redundant, datasets from these domains presents a significant computational bottleneck. Existing solutions typically focus on optimizing model architectures or optimizers, while overlooking the inherent inefficiency of the training data itself. This conventional approach of iterating over the entire static dataset each epoch wastes considerable resources on easy-to-learn or repetitive samples. In this paper, we explore a *novel training-efficiency techniques, namely learning from complexity with dynamic sample pruning, `ST-Prune`, for spatio-temporal forecasting*. Through dynamic sample pruning, we aim to intelligently identify the most informative samples based on the model's real-time learning state, thereby accelerating convergence and improving training efficiency. Extensive experiments conducted on real-world spatio-temporal datasets show that `ST-Prune` significantly accelerates the training speed while maintaining or even improving the model performance, showcasing its effectiveness, efficiency, and generalizability.

1 INTRODUCTION

With the rapid advancement of sensing technologies, massive volumes of spatio-temporal (ST) data are being collected and leveraged in data-driven forecasting scenarios, enabling critical services ranging from traffic management to weather forecasting and power grid operations (Avila & Mezić, 2020; Nguyen et al., 2023; Yang et al., 2024; Li et al., 2025a). To capture the complex nonlinear dynamics inherent in such data, spatio-temporal neural networks (Jin et al., 2023; 2024) have become the dominant and most powerful paradigm. However, while research community has largely focused on designing sophisticated architectures for marginal performance gains (Shao et al., 2023) in recent years, *a fundamental and costly bottleneck has been overlooked: the training process itself.*

Standard spatio-temporal training protocols require iterating over the entire samples in every training epoch (Liu et al., 2024b; Jin et al., 2024). As a result, the widely adopted benchmarks (Li et al., 2018b; Song et al., 2020) are typically restricted to limited spatial regions and temporal spans, severely limiting the scalability of spatio-temporal nueral networks training and inflating their computational cost. This raises a natural research question: *do we really need to compute over all available spatio-temporal samples during model training stage?*

To answer this question, we first conduct a detailed analysis of the widely used spatio-temporal benchmark (PEMS08). As shown in Figure 1, the left panel visualizes the Pearson correlation matrix across spatial nodes, revealing that the vast majority exhibit high similarity (≥ 0.8). Even node pairs with lower similarity in left panel still display recurring periodic temporal patterns, as illustrated in the middle panel. Principal component analysis (PCA) along both spatial and temporal dimensions (right panel) further shows that a small number of components are sufficient to reconstruct most of the variance. *Together, these visualizations and statistics reveal a high degree of spatio-temporal redundancy, which opens up new opportunities for us to make more efficient use of training data.*

Recent data-centric acceleration research (Zha et al., 2025) have devoted significant effort to identifying representative subsets within training datasets. Typical techniques include data pruning (Raju et al., 2021; Wang et al., 2023; Qin et al., 2024; Zhang et al., 2024b;a; Wang et al., 2025), dataset distillation (Nguyen et al., 2021; Wang et al., 2022; Cazenavette et al., 2022; Zhao & Bilen, 2023; Li et al., 2023; Zhang et al., 2024c), and coreset selection (Har-Peled & Mazumdar, 2004; Chen,

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.

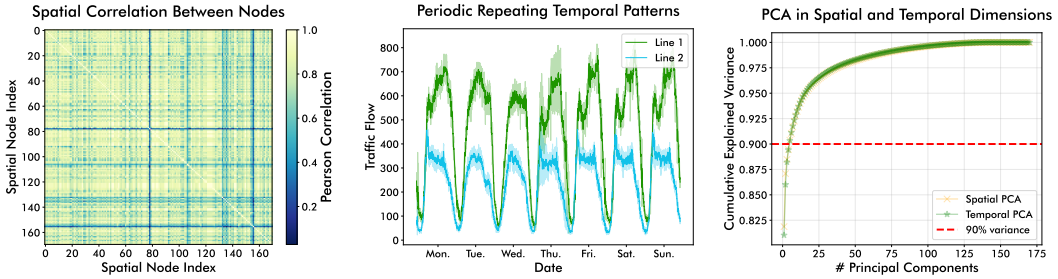
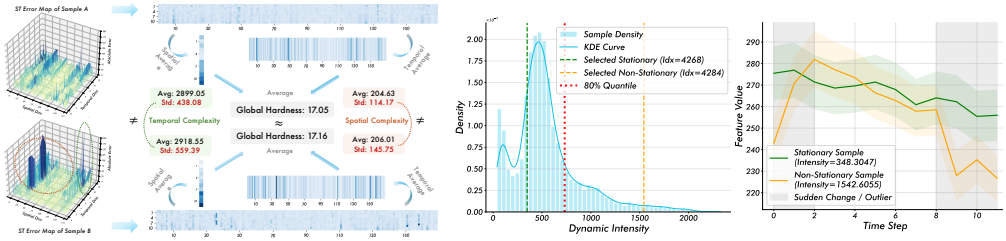


Figure 1: The spatio-temporal data redundancy characteristics and statistical properties along the spatial and temporal dimensions, exemplified by the PEMS08 (Song et al., 2020). For additional visualizations and statistical information on other ST datasets, please refer to the Appendix A.1.



(a) Illustration of the Averaging Masking Effect.

(b) Distribution of Dynamic Intensity & Patterns of Selected Samples

Figure 2: Further analysis of spatio-temporal data insights. (a) Averaging Masking Effect: Low-error nodes dilute critical localized anomalies, necessitating a structural scoring mechanism beyond simple mean error. (b) Long-tail Stationarity Distribution: The dominance of stationary patterns motivates our stationarity-aware rescaling to prevent distribution shift and maintain representativeness.

2009; Toneva et al., 2018; Shim et al., 2021), which respectively retain, synthesize, or select a smaller yet information-rich subset from the raw data. Despite their promise, these methods are primarily developed for computer vision and natural language tasks and *fail to fully exploit the unique redundancy characteristics of spatio-temporal data highlighted above.*

To address this critical gap, we propose `ST-Prune` , a novel dynamic sample pruning framework specifically tailored for spatio-temporal training. Distinct from traditional paradigms that passively process the entire dataset, `ST-Prune` actively curates high-value data subsets to optimize training efficiency. Our fundamental insight is that directly applying generic dynamic pruning strategies (Raju et al., 2021; Qin et al., 2024; Moser et al., 2025) to spatio-temporal data proves ineffective. This failure stems from two specific phenomena: the *Averaging Masking Effect* (Figure 2 a) , where critical localized failures are obscured by low global errors, and the inherent *Long-tail Stationarity Distribution* (Figure 2 b), where standard pruning induces severe distribution shifts. To this end, `ST-Prune` operationalizes these insights via two innovative components: complexity-informed scoring metric, which incorporates a spatio-temporal heterogeneity penalty to identify structural samples that appear ”globally trivial yet locally intractable“; and stationarity-aware gradient rescaling, which adaptively adjusts weights based on dynamic intensity. This ensures that the preserved samples maintain an unbiased representation of the original data distribution while discarding substantial redundant stationary samples. Ultimately, this design not only significantly mitigates computational overhead but also ensures the robust capture of diverse and complex spatio-temporal patterns. In summary, our contributions are:

- We propose `ST-Prune` , a dynamic sample pruning method of spatio-temporal training, that shifts the ST research focus from optimizing the model to intelligently optimizing the training data flow.
- We design a novel framework consisting of a complexity-informed metric for real-time informativeness assessment and a stationarity-aware distribution rescaling for stable, unbiased training.
- Extensive experiments on multiple real-world spatio-temporal datasets demonstrate that `ST-Prune` drastically reduces training time while maintaining or even improving the predictive accuracy of various backbones, proving its effectiveness, efficiency, scalability and universality.

2 Related Work

Efficient Spatio-Temporal Training. Spatio-temporal forecasting tasks typically rely on neural networks that integrate spatial and temporal operators (Chen & Liang, 2025a;b) to capture spatial dependencies and temporal dynamics. However, the large number of nodes and extended time horizons (Liu et al., 2024b; Yin et al., 2025) result in substantial computational costs during training. Previous studies have sought efficiency improvements through model-level optimizations, such as prior structure Shao et al. (2022); Cini et al. (2023); Fu et al. (2025), graph sparsification (Wu et al., 2025; Ma et al., 2025), and model distillation (Tang et al., 2024; Chen et al., 2025), or sample-level transformations, including subgraph (Jiang et al., 2023; Liu et al., 2024a; Wang et al., 2024; Weng et al., 2025; Zhao et al., 2026) and input window (Fang et al., 2025; Ockerman et al., 2025; Shao et al., 2025) sampling. While these approaches reduce computational demand at the model or optimization level, they rarely provide explicit control over the number or distribution of training samples and often tightly couple network component design. *In contrast, we aim to develop a general dynamic spatio-temporal sample pruning strategy that accelerates the training of arbitrary spatio-temporal neural networks while maintaining competitive performance.*

Data-Centric Acceleration Strategy. Data-centric acceleration strategies (Zha et al., 2025) have drawn considerable attention because they enhance model training and inference efficiency from a data management perspective. Representative approaches include data distillation (Lei & Tao, 2023; Yu et al., 2023), data pruning, and data selection (Moser et al., 2025). Data distillation seeks to compress the original data distribution into a smaller yet information-equivalent synthetic dataset to reduce training cost, whereas data pruning and selection reduce computational burden by removing redundant samples or selecting the most representative subsets. However, these methods are primarily designed for vision (Wang et al., 2018) or language (Albalak et al., 2024) tasks and struggle to exploit the high redundancy and strong correlations inherent in spatio-temporal data. Moreover, they are often static and sensitive to sample size (Guo et al., 2022), which may limit generalization under dynamically evolving spatio-temporal distributions. *Notably, our method exploits the complexity of ST samples and dynamically adjusts sample selection to maximize computational efficiency.*

3 Preliminaries

Probelm Definition (Spatio-Temporal Forecasting). From an optimization perspective, spatio-temporal forecasting aims to solve an empirical risk minimization problem. Given a large-scale spatio-temporal training dataset $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{|\mathcal{D}|}$ (and an optional graph structure prior $\mathcal{G}$), the objective is to find the optimal model parameters θ^* by minimizing a global loss function $\mathcal{J}(\theta; \mathcal{D})$:

$$\min_{\theta} \mathcal{J}(\theta; \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathcal{L}(f_\theta(X_i; \mathcal{G}), Y_i), \tag{1}$$

where f_θ is the forecasting spatio-temporal neural network, and $\mathcal{L}$ is a loss metric (*e.g.*, MAE). $(X_i \in \mathbb{R}^{N \times T_p \times F}, Y_i \in \mathbb{R}^{N \times T_f \times F})$ represent the sensor readings of C measurement features at N spatial locations over T_p past and T_f future consecutive time steps, respectively. This formulation treats every sample (X_i, Y_i) equally and implies that the gradient for each training epoch is computed over the entire, often redundant, dataset $\mathcal{D}$. This exhaustive summation is the primary source of computational inefficiency in the spatio-temporal training process.

Task Definition (Dynamic Sample Pruning). We formulate dynamic sample pruning as an optimization problem aimed at approximate the convergence of the objective in Equation 1. Instead of using the full dataset $\mathcal{D}$ at each epoch e, we seek to identify an optimal training subset $\mathcal{D}_e^* \subset \mathcal{D}$ with a constrained size $|\mathcal{D}_e| = k_e < |\mathcal{D}|$. The model parameters are then updated based only on this subset: $\theta_e = \text{Update}(\theta_{e-1}, \mathcal{D}_e)$. Ideally, the optimal subset $\mathcal{D}_e^*$ should be the one that provides the steepest descent direction for the true, underlying data distribution $\mathcal{P}$:

$$\mathcal{D}_e^* = \underset{\mathcal{D}_e \subset \mathcal{D}, |\mathcal{D}_e| = k_e}{\text{argmin}} \mathbb{E}_{(X,Y)\sim\mathcal{P}}[\mathcal{L}(f_{\theta_e}(X; \mathcal{G}), Y)]. \tag{2}$$

However, minimizing this objective is intractable as the true distribution $\mathcal{P}$ is unknown. However, we have empirically shown above that this distribution exhibits a significant low-rank property, that is, the training samples are highly redundant. Thus, the core challenge of dynamic sample pruning is to design a tractable and efficient proxy strategy to select an informative subset $\mathcal{D}_e$ at each epoch.

4 METHODOLOGY

4.1 COMPLEXITY-INFORMED PRUNING

4.1.1 BEYOND THE AVERAGING MASKING EFFECT

Before elaborating on our proposed framework, we first analyze the limitations of standard dynamic pruning techniques (Katharopoulos & Fleuret, 2018; Raju et al., 2021; Qin et al., 2024) when applied to spatio-temporal domains. Existing methods typically quantify sample difficulty using a global loss metric, $\ell_i = \|f_\theta(X_i) - Y_i\|$, where a lower ℓ_i indicates an "easy" sample. However, in spatio-temporal forecasting, the scalar loss ℓ_i is an aggregate metric averaged over N spatial nodes and T time steps. This aggregation introduces a critical pathology we term the *Averaging Masking Effect*.We empirically demonstrate this phenomenon in Figure 2 (a), which visualizes the error landscapes of two distinct samples from a real-world traffic dataset:

- **Sample A (Global Noise):** The error is distributed relatively uniformly across the spatio-temporal field. The global hardness (mean MAE) is calculated as 17.05.
- **Sample B (Local Anomaly):** As highlighted by the red circle in Figure 2 (a), this sample contains critical localized failures (*e.g.*, severe congestion spikes at specific hubs), while the remaining nodes exhibit low errors. Surprisingly, its global hardness is 17.16.

Despite the fundamental difference in structural information, standard pruning methods viewing only the global hardness (≈ 17.1) would treat these two samples as identical. Consequently, the informative Sample B, which drives the learning of local spatial dynamics, may risks being pruned as "easy" data. This observation confirms that magnitude-based metrics fail to distinguish structural complexity from background noise. To address this, we propose *Spatio-Temporal Complexity Scoring*, a metric designed to capture the *non-uniformity* (σ) of the error distribution, ensuring that samples with high spatial heterogeneity (like Sample B) are preserved even if their global mean is low.

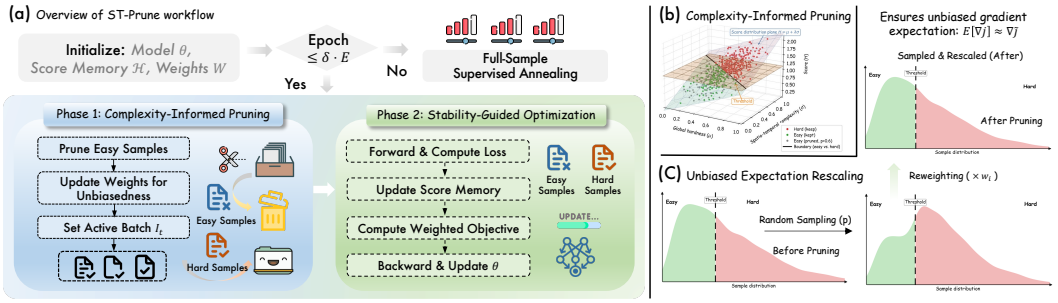


Figure 3: The overview of the ST-Prune for efficient data pruning during spatio-temporal training.

4.1.2 SPATIO-TEMPORAL COMPLEXITY SCORING

Formally, we propose ST-Prune . As shown in Fig. 3, during the forward propagation process, each sample retains a score related to its loss value. The average of these scores is set as the pruning threshold. In each epoch, a corresponding proportion of samples with low scores are pruned.

To operationalize the insight from Figure 2 (a), we propose a composite scoring function that explicitly penalizes performance heterogeneity. Let $\mathbf{E}_t^{(i)} \in \mathbb{R}^{N \times T}$ denote the absolute error matrix for the i-th sample at epoch t. We define the structural informativeness score $\mathcal{H}_t(i)$ as:

$$\mathcal{H}_t(i) = \underbrace{\mu(\mathbf{E}_t^{(i)})}_{\text{Global Hardness}} + \lambda \cdot \left[\underbrace{\sigma_{\text{space}}(\mathbf{E}_t^{(i)}) + \sigma_{\text{time}}(\mathbf{E}_t^{(i)})}_{\text{Spatio-Temporal Complexity}} \right], \tag{3}$$

where $\mu(\cdot)$ denotes the global mean function. The terms σ_{space} and σ_{time} represent the standard deviations calculated along the spatial and temporal dimensions of the error matrix, respectively.

4.1.3 RANDOMIZED PRUNING POLICY

To reduce computational overhead while maintaining data diversity, we categorize samples into an *Informative Set* $\mathcal{S}_{inf}$ and a *Redundant Set* $\mathcal{S}_{red}$ at the beginning of each epoch t. Let $\bar{\mathcal{H}}_t$ be the average score of the entire dataset.

$$z_i \in \begin{cases} \mathcal{S}_{inf}, & \text{if } \mathcal{H}_t(i) \geq \bar{\mathcal{H}}_t \\ \mathcal{S}_{red}, & \text{if } \mathcal{H}_t(i) < \bar{\mathcal{H}}_t \end{cases} \tag{4}$$

Unlike static pruning which permanently discards $\mathcal{S}_{red}$, we adopt a randomized "soft" pruning strategy. All samples in $\mathcal{S}_{inf}$ are retained. For samples in $\mathcal{S}_{red}$, we retain them with a probability $p \in (0, 1)$, determined by the target pruning ratio. This ensures that even currently "easy" samples have a chance to be revisited, preventing the model from catastrophic forgetting of basic patterns.

4.2 STABILITY-GUIDED OPTIMIZATION

4.2.1 STATIONARITY-AWARE GRADIENT RESCALING

While the scoring metric identifies which samples to prune, a naive removal of "easy" samples would inevitably alter the training data distribution. As highlighted in Figure 2 (b), spatio-temporal datasets exhibit a long-tail distribution of dynamic intensity: the vast majority of samples are stationary (low temporal variance), while high-dynamic events are rare. Standard pruning methods Qin et al. (2024) use a uniform rescaling weight $w = \frac{1}{1-r}$, where r is the pruning rate, and thus ignore this imbalance. They over-prune stationary samples, shift the training distribution toward the "tail" (non-stationary events), and make the model overfit to extreme dynamics at the expense of robustness to regular patterns.

To rectify this distribution shift, we propose a *Stationarity-Aware Rescaling* strategy. We first quantify the *dynamic intensity* δ_i of each sample i as the temporal variance of its ground truth targets: $\delta_i = \mathrm{Var}_t(Y_i)$. For the subset of retained "inf" samples $\mathcal{S}'_{inf}$, we assign an adaptive weight w_i that is inversely proportional to their dynamic intensity:

$$w_i = \frac{1}{1-r} \cdot \underbrace{\left(\frac{\bar{\delta}_{\mathcal{D}}}{\delta_i + \epsilon}\right)^{\alpha}}_{\text{Stationarity Correction}}, \tag{5}$$

where $\bar{\delta}_{\mathcal{D}}$ is the global average intensity, and $\alpha \geq 0$ controls the correction strength. This mechanism ensures that preserved stationary samples (low δ_i) receive higher weights, effectively representing the population of pruned stationary data. Conversely, highly dynamic samples (high δ_i) have been naturally retained through our scoring metric and will receive standard weights. Formally, the modified objective $\tilde{\mathcal{J}}(\theta)$ for the current epoch becomes:

$$\tilde{\mathcal{J}}(\theta) = \sum_{i \in \mathcal{S}_{red}} \mathcal{L}(x_i, y_i; \theta) + \sum_{j \in \mathcal{S}'_{inf}} w_j \cdot \mathcal{L}(x_j, y_j; \theta). \tag{6}$$

This reweighting guarantees that the gradient expectation remains unbiased not only in magnitude but also in terms of *dynamic regime distribution*, ensuring convergence consistency with the full dataset.

Remark. "Low temporal variance" here refers to the local stationarity of raw ST data within short windows. However, Eq. 3 measures the variance of prediction error maps, not raw data. These are inherently decoupled: even with smooth inputs, errors can exhibit extreme, localized fluctuations. Our complexity term captures these error instabilities. Consequently, loss-level ST structural information is vital for pruning, while sample-level dynamic intensity necessitates our weight rescaling strategy.

4.2.2 TRAINING SCHEDULE WITH ANNEALING

While the unbiased estimator holds in expectation, the variance of the gradients inevitably increases due to downsampling. To mitigate this effect and ensure stable convergence in the final phase of training, we introduce a deterministic annealing strategy. Given a total training budget of E epochs and a cutoff ratio δ (*e.g.*, $\delta = 0.9$), we perform the pruning strategy only for the first $\delta \cdot E$ epochs. For the remaining epochs $t > \delta \cdot E$, we revert to full-dataset training. This allows the model to perform lightweight fine-tuning on all samples, eliminating residual variance and ensuring that the final model performance is more strictly lossless compared to that of the original strategy.

5 EXPERIMENTS

In this section,we conduct extensive experiments to answer the following research questions (RQs):

- **RQ1: (Effectiveness)** Can `ST-Prune` outperform existing pruning methods on common datasets?
- **RQ2: (Efficiency)** How efficient is `ST-Prune` compared with existing data pruning baselines?
- **RQ3: (Scalability)** How well does `ST-Prune` scale to large datasets and foundation models?
- **RQ4: (Universality)** Is `ST-Prune` effective across diverse backbones, optimizers, and tasks?
- **RQ5: (Mechanism & Robustness)** How does `ST-Prune` work? Which components or strategies crucial, and are they sensitive to different designs or hyper-parameters? How is the interpretability?

5.1 EXPERIMENTAL SETUP

Dataset and Evaluation Protocol. We adopt widely used public benchmarks commonly employed in the literature, covering representative spatio-temporal forecasting scenarios, including the traffic flow dataset *PEMS08* (Song et al., 2020) and the charging station dataset *UrbanEV* (Li et al., 2025a). In addition, we use the large-scale spatio-temporal benchmark *LargeST* (Liu et al., 2024b), which contains *SD*, *GBA*, and *GLA*, to evaluate the training efficiency of our method on large datasets. Following standard practice (Shao et al., 2023), all datasets are chronologically split into training, validation, and test sets with a ratio of $6:2:2$. However, to ensure fair comparison of efficiency experiments, we disable early stopping mechanism and train for a fixed 100 epochs. Following the default settings (Jin et al., 2023), we use the past 12 steps to predict the average of the next 12 steps, and the mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) are used as metrics to evaluate the performance. Furthermore, we vary the dataset remaining ratio across $\{10\%, 30\%, 50\%, 70\%\}$ to analyze model performance under different pruning levels. For more details about datasets and protocols, please refer to Appendix A.1.

Baselines and Parameter Settings. We benchmark against a comprehensive suite of pruning strategies adapted for spatio-temporal tasks, categorized into: (1) *Static pruning methods*: random (*Hard Random*), geometry-based (*CD*, *Herding*, *K-Means*), uncertainty-based (*Least Confidence*, *Entropy*, *Margin*), loss-based (*Forgetting*, *GraNd*), decision boundary-based (*Cal*), bi-level (*Glister*), and submodular (*GraphCut*, *FaLo*); and (2) *Dynamic pruning methods*: Random(*Soft Random*), uncertainty-based (ϵ-*greedy*, *UCB*), and loss-based (*InfoBatch*). We employ classical and popular *GWNet* (Wu et al., 2019) as the default ST backbone, alongside *STAEformer* (Liu et al., 2023) and *STID* (Shao et al., 2022) for cross-architecture evaluation, and the foundation model *OpenCity* Li et al. (2025b) to assess scalability. To ensure fair comparison, all models are trained using SGD (momentum 0.9, weight decay 1e-4) with a cosine annealing scheduler. More details on baselines, models, and parameters are reported in Appendix A.2.

5.2 EFFECTIVENESS STUDY (RQ1)

Table 1 compares the MAPE results of `ST-Prune` against representative data pruning and selection methods (Full metric results can be found in Appendix B). The best results are highlighted in **bold pink**, and the second-best in underlined in blue. Besides, we report the relative performance change ($_{\downarrow degradation}$ or $_{\uparrow improvement}$) compared to the whole dataset baseline. We observe that: ❶ *Dynamic pruning strategies generally outperform static ones.* Across various baselines, dynamic methods achieve superior performance by leveraging training dynamics to better assess sample importance. Interestingly, both Hard Random and Soft Random remain competitive at low retention rates, mirroring phenomena observed in vision datasets Guo et al. (2022). We hypothesize that for strategy-based methods, specific algorithmic inductive biases may introduce a harmful sampling expectation bias. ❷ *Dataset redundancy varies across domains.* We observe that *PEMS08* is more sensitive to pruning than *UrbanEV*, indicating differing redundancy levels. For example, at 10% retention (excluding our method), performance degradation on *PEMS08* ($20.40\% \sim 45.26\%$) significantly exceeds that of *UrbanEV* ($7.34\% \sim 28.63\%$). However, performance recovers rapidly as the retention ratio increases; at a 50% pruning level, degradation across both datasets remains within single digits. ❸ *`ST-Prune` consistently excels across all pruning ratios.* `ST-Prune` outperforms both static and dynamic baselines in all settings. Even at 10% retention rate, it keeps practical utility with minimal degradation. Notably, `ST-Prune` surpasses whole dataset performance on *UrbanEV*.

Table 1: Performance comparison to state-of-the-art dataset pruning methods when remaining $\{10\%, 30\%, 50\%, 70\%\}$ of the full training set. All methods are trained using same ST backbone.

	Dataset	PEMS08 (MAPE (%) ↓)				URBANEV (MAPE (%) ↓)			
	Remaining Ratio %	10	30	50	70	10	30	50	70
Static	Hard Random	13.93 ↓26.87%	14.80 ↓34.79%	12.07 ↓9.93%	12.06 ↓9.84%	35.89 ↓20.32%	31.90 ↓6.94%	31.52 ↓5.67%	30.69 ↓2.88%
	CD (Agarwal et al., 2020)	14.55 ↓32.51%	12.25 ↓11.57%	11.96 ↓8.93%	11.83 ↓7.74%	38.37 ↓28.63%	32.20 ↓7.95%	30.40 ↓1.91%	31.01 ↓3.96%
	Herding (Welling, 2009)	15.79 ↓43.81%	12.11 ↓10.29%	11.58 ↓5.46%	12.84 ↓16.94%	34.63 ↓16.09%	30.97 ↓3.82%	30.94 ↓3.72%	30.97 ↓3.82%
	K-Means (Sener & Savarese, 2018)	15.71 ↓43.08%	12.24 ↓11.48%	12.48 ↓13.66%	11.52 ↓4.92%	38.45 ↓28.90%	31.85 ↓6.77%	30.96 ↓3.79%	30.69 ↓2.88%
	Least Confidence (Coleman et al., 2019)	14.31 ↓30.33%	12.62 ↓14.94%	11.74 ↓6.92%	11.61 ↓5.74%	34.23 ↓14.75%	36.87 ↓23.60%	30.98 ↓3.86%	30.55 ↓2.41%
	Entropy (Coleman et al., 2019)	13.22 ↓20.40%	12.82 ↓16.76%	12.83 ↓16.85%	11.28 ↓2.73%	36.42 ↓22.09%	31.20 ↓4.59%	30.65 ↓2.75%	30.49 ↓2.21%
	Margin (Coleman et al., 2019)	14.06 ↓28.05%	12.55 ↓14.30%	11.60 ↓5.65%	11.38 ↓3.64%	37.21 ↓24.74%	31.89 ↓6.91%	32.73 ↓9.72%	30.89 ↓3.55%
	Forgetting (Toneva et al., 2018)	14.92 ↓35.88%	12.19 ↓11.02%	11.46 ↓4.37%	11.29 ↓2.82%	32.84 ↓10.09%	31.36 ↓5.13%	30.63 ↓2.68%	30.69 ↓2.88%
	GraNd (Paul et al., 2021)	13.93 ↓26.87%	13.09 ↓19.22%	12.33 ↓12.30%	11.50 ↓4.74%	35.26 ↓18.20%	31.71 ↓6.30%	30.90 ↓3.59%	30.20 ↓1.24%
	Cal (Margatina et al., 2021)	14.79 ↓34.70%	12.40 ↓12.93%	12.64 ↓15.12%	12.40 ↓12.93%	33.77 ↓13.21%	32.25 ↓8.11%	31.45 ↓5.43%	30.81 ↓3.29%
	Glister (Killamsetty et al., 2021)	14.48 ↓31.88%	13.23 ↓20.49%	12.51 ↓13.93%	11.74 ↓6.92%	33.70 ↓12.97%	33.65 ↓12.81%	32.78 ↓9.89%	31.26 ↓4.79%
	GraphCut (Iyer et al., 2021)	15.07 ↓37.25%	12.26 ↓11.66%	11.84 ↓7.83%	11.27 ↓2.64%	35.73 ↓19.78%	31.49 ↓5.56%	30.76 ↓3.12%	31.23 ↓4.69%
	FaLo (Iyer et al., 2021)	15.95 ↓45.26%	12.56 ↓14.39%	12.13 ↓10.47%	11.40 ↓3.83%	35.07 ↓17.57%	31.72 ↓6.34%	31.23 ↓4.69%	31.01 ↓3.96%
Dynamic	Soft Random	15.00 ↓36.61%	12.41 ↓13.02%	12.01 ↓9.38%	11.34 ↓3.28%	33.55 ↓12.47%	31.34 ↓5.06%	30.48 ↓2.18%	30.35 ↓1.74%
	ϵ-greedy (Raju et al., 2021)	13.82 ↓25.87%	12.21 ↓11.20%	11.84 ↓7.83%	11.24 ↓2.37%	33.25 ↓11.46%	30.98 ↓3.86%	30.46 ↓2.11%	30.30 ↓1.58%
	UCB (Raju et al., 2021)	13.80 ↓25.68%	12.41 ↓13.02%	11.76 ↓7.10%	11.28 ↓2.73%	33.24 ↓11.43%	31.08 ↓4.19%	30.43 ↓2.01%	30.29 ↓1.54%
	InfoBatch (Qin et al., 2024)	13.66 ↓24.41%	12.12 ↓10.38%	11.83 ↓7.74%	11.33 ↓3.19%	32.02 ↓7.34%	30.78 ↓3.18%	30.24 ↓1.37%	29.83 ↓0.00%
	ST-Prune (Our)	**12.13** ↓13.21	**11.75** ↓7.01	**11.11** ↓5.56	**11.19** ↓2.82	**29.56** ↑0.91	**29.42** ↑1.37	**29.15** ↑2.28	**29.05** ↑2.61
	Whole Dataset	$10.98_{\pm 0.23}$				$29.83_{\pm 0.09}$			

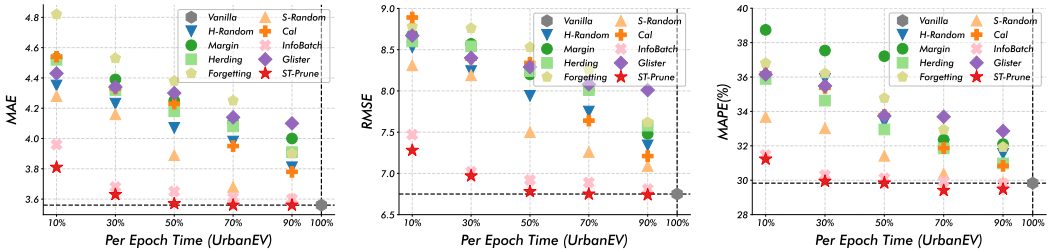


Figure 4: The trade-off between per-epoch time and performance in *UrbanEV*. Specifically, we report the test performance when methods achieve per epoch times of {10%, 30%, 50%, 70%, 90%} of the full dataset training time. "Vanilla" denotes the full dataset training. Time: Total wall-clock time.

We attribute this to *UrbanEV*'s low redundancy and high signal-to-noise ratio characteristics, while ST-Prune further improves performance by filtering out noisy samples during the pruning process.

5.3 Efficiency Study (RQ2)

Figure 4 illustrates the efficiency comparison. We observe the following: ❹ *ST-Prune strikes a superior balance between training time and performance,* significantly outperforming other baselines. Notably, ST-Prune achieves nearly 2× acceleration (≈ 50% reduction time) with negligible performance loss across different metrics. ❹ Even under an aggressive 10× speedup, ST-Prune incurs only marginal degradation, maintaining acceptable performance where others deteriorate rapidly.

5.4 Scalability Study (RQ3)

To assess the scalability of our method, we conduct evaluations from both data and model perspectives. Specifically, we validate performance using the representative large-scale ST dataset *LargeST* Liu et al. (2024b). Additionally, we employ the ST foundation model *OpenCity* Li et al. (2025b) across Mini, Base, and Plus scales.

Large-Scale Datasets. Table 2 presents comparative performance on the LargeST benchmark. We highlight the following observations: ❶ *Competitive efficiency-performance trade-off.* At a 10% data retention rate, heuristic baselines (*i.e.*, *Soft Random*) achieve the fastest dynamic pruning speeds but suffer significant performance degradation. In contrast, ST-Prune achieves relative performance gains of $11.09\% \sim 38.80\%$ over heuristics and even slightly outperforms full-data training, despite the additional computational overhead incurred by our customized policy design. ❷ *Robustness across spatio-temporal scales.* Unlike heuristic baselines (*i.e.*, *Soft Random*) that exhibit marked performance drops on smaller datasets (*e.g.*, SD), our ST-Prune maintains consistent performance advantages across varying scales. ❸ *Superior efficiency and scalability.* Even in extreme 1% retention scenarios, our model retains competitive performance while drastically reducing computational costs. By cutting time from days to hours, it demonstrates significant potential for large-scale deployment.

ST-Foundation Models. Following Li et al. (2025b) settings, we further validate model scalability by integrating ST-Prune with the *OpenCity* foundation model series. Figure 5 illustrates the

Table 2: Comparison of ST-Prune with the fastest dynamic method *Soft Random* on *LargeST*.

Dataset	SD (#716)				GBA (#2352)				GLA (#3834)			
	MAE	RMSE	MAPE (%)	TIME	MAE	RMSE	MAPE (%)	TIME	MAE	RMSE	MAPE (%)	TIME
Whole Dataset	24.13 ± 1.14	38.03 ± 1.32	15.77 ± 0.87	2.07h	27.34 ± 1.04	42.11 ± 1.19	24.11 ± 0.48	13.05h	26.26 ± 0.84	41.12 ± 1.47	16.06 ± 0.63	30.53h
Soft Random (10%)	30.41	47.96	20.43	0.52h	33.05	48.71	39.30	2.97h	29.74	45.98	20.97	7.25h
ST-Prune (10%)	24.01	37.73	15.51	0.61h	27.32	41.94	24.05	4.92h	26.22	40.88	15.95	13.38h
ST-Prune (1%)	24.76	38.19	16.51	0.31h	28.16	42.77	23.64	2.84h	26.95	41.57	17.62	7.12h
Improvement	↑ 21.05%	↑ 21.33%	↑ 24.08%	×6.68	↑ 17.33%	↑ 13.90%	↑ 38.80%	×4.60	↑ 11.84%	↑ 11.09%	↑ 23.94%	×4.29

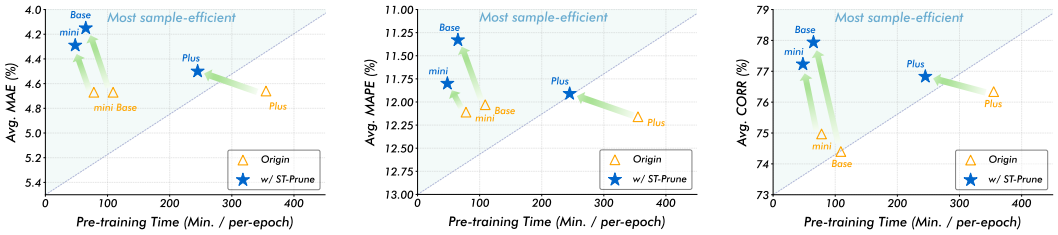


Figure 5: Performance and efficiency trade-offs with ST-Prune for ST-foundation model *OpenCity*.

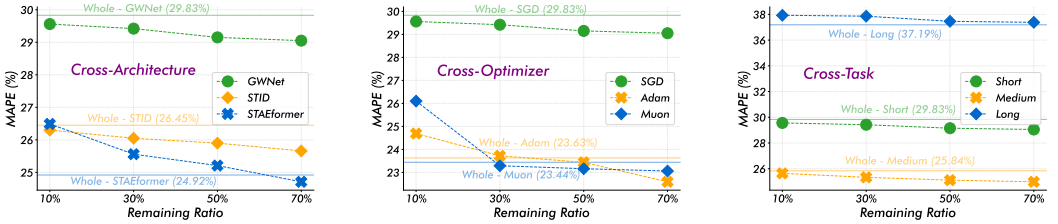


Figure 6: Performance variations across architectures, optimizers, and tasks under different data rates.

trade-off between per-epoch pre-training time and downstream performance, revealing the following: ❶ *Simultaneous gains in efficiency and capability.* ST-Prune achieves a "win-win" outcome, with green arrow consistently pointing towards the "most sample-efficient" region. Across all model scales, ST-Prune not only significantly reduces the pre-training time (shifting left) but also generally improves the prediction accuracy (shifting upwards). This suggests that our approach effectively purifies the pre-training corpus by eliminating noisy or redundant samples that hinder convergence. ❷ *Democratizing large-scale model training.* For the computation-intensive Plus scale, ST-Prune significantly curtails per-epoch training time (*e.g.*, from ∼350 to 250 mins) while outperforming the original baseline. Notably, ST-Prune reduces the training cost of the Base scale to a level smaller than the Mini scale, thus effectively training larger and more powerful models under constrained computational resources.

5.5 Universality Study (RQ4)

To evaluate the universality of our method, we conduct evaluations on *UrbanEV* across different backbones, optimizers, and tasks. Specifically, we validate performance using representative ST architectures including GWNet, STID, and STAEformer; distinct optimizers such as SGD, Adam Kingma & Ba (2015), and Muon Jordan et al. (2024); and prediction horizons covering short-term (*i.e.*, 12→12), medium-term (*i.e.*, 24→24), and long-term (*i.e.*, 96→96) tasks. From each subgraph in Figure 6, we observed that:

Cross-Architecture Evaluation. ❶ *Universal performance gains.* Notably, across all backbones, models trained with ST-Prune (dashed lines) consistently outperform their full-dataset counterparts (solid lines), achieving lower MAPE at most retention levels. ❷ *Architectural robustness.* Our method significantly boosts lightweight models (STID / GWNet) in low-retention regimes, and even pushes the performance of the over-parameterized model STAEformer as the data retention ratio grows.

Cross-Optimizer Evaluation. ❶ *Optimization Robustness.* ST-Prune is optimizer-agnostic, outperforming full-dataset baselines even with SGD at low retention rates. ❷ *Boosting SoTA Optimizers.* For advanced optimizers like Adam and Muon, which already yield low error rates ($< 24\%$), ST-Prune further pushes performance boundaries. At $> 30\%$ retention, it enables both to surpass their full-dataset performance, demonstrating its efficacy across diverse optimization landscapes.

Cross-Task Evaluation. ❶ *Short-to-Medium Robustness.* ST-Prune consistently outperforms full-dataset baselines across all retention ratios, effectively capturing immediate temporal depen-

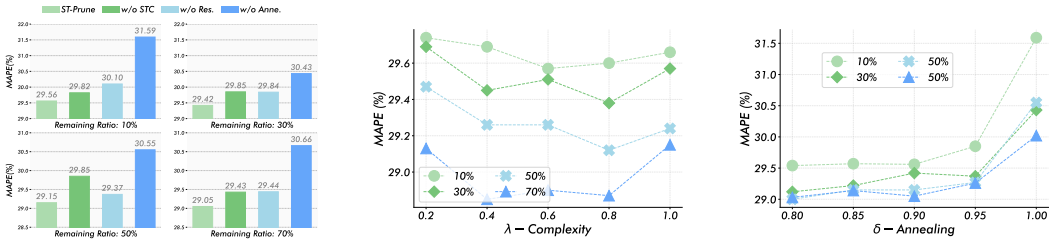


Figure 7: Ablation and parameter sensitivity analysis on *UrbanEV* dataset under different data rates.

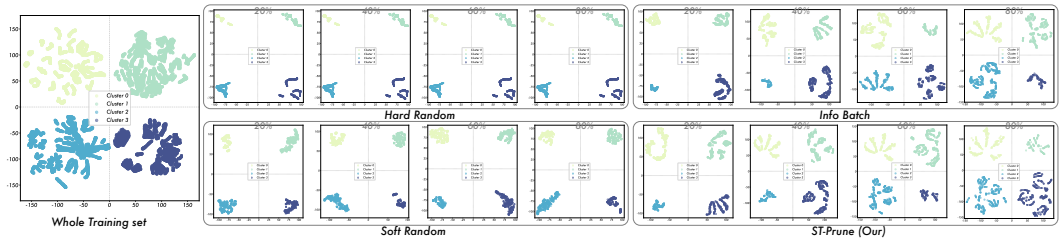


Figure 8: Visualization of t-SNE fixed-corner clustering of the *PEMS08* distribution with the full training set and a retention rate of 20%. The right panel shows the evolution of selected subsets through different pruning strategies at different training progresses (from 20% to 80%).

dencies within denoised subsets. ❷ *Long-Term Resilience.* Despite the complexity of long-horizon dependencies, `ST-Prune` remains robust, matching full-dataset performance at just 50% retention.

5.6 Mechanism & Robustness Study (RQ5)

Ablation Study. To validate the effectiveness of each component in `ST-Prune` , we compare against four variants: removing the spatio-temporal complexity score (*w/o STC*), the score rescaling strategy (*w/o Res.*), and the annealing scheduler (*w/o Anne.*). As shown in Figure 7 Left, we observe that: ❶ *Every component matters.* Removing any part leads to performance degradation, demonstrating that all modules play indispensable roles. ❷ *Necessity of scheduling with annealing.* The performance degradation of *w/o Anne.* is the most significant, especially under low retention rates, which illustrates its positive role in variance correction for sparse scenarios. ❸ *Importance of Scoring Components.* Removing *STC* leads to noticeable error increases, indicating that both the intrinsic difficulty of samples and the historical consistency are critical for robust sample valuation.

Parameter Sensitivity. We further investigate the sensitivity of `ST-Prune` to key hyperparameters: the complexity weight and the annealing rate . ❶ *Impact of Complexity Weight* . Figure 7 (Middle) exhibits a convex trend for , where extreme values (too small or too large) hamper performance. The optimal range lies between 0.4 and 0.6, suggesting that balancing the raw loss with spatio-temporal structural complexity is essential for identifying informative samples. ❷ *Impact of Annealing Rate* . Figure 7 (Right) reveals that performance is stable when $\delta \in [0.8, 0.95]$ but deteriorates rapidly as δ approaches 1, *i.e. w/o Anne.*, consistent with previous observations in the ablation study.

Interpretability Study. To intuitively understand how `ST-Prune` selects spatio-temporal sample compared to baselines, we visualize the latent distribution of the selected subsets using t-SNE in Figure 8. We can observe that: ❶ *Dynamic alignment with training dynamics.* While heuristic baselines (*e.g.*, *Hard/Soft Random*) produce static, sparse distributions via uniform sampling, `ST-Prune` exhibits a clear evolutionary trajectory, balancing the preservation of representative centroids with the exploration of global diversity. ❷ *Restoration of intrinsic data topology.* Surpassing *InfoBatch*, `ST-Prune` rapidly reconstructs the approximate manifold topology of the original distribution during training, thereby ensuring robust generalization performance.

6 Conclusion and Future Work

In this paper, we investigate the redundancy inherent in spatio-temporal training datasets and explore effective approaches for efficient dynamic data pruning. We propose `ST-Prune` that uses complexity-informed scoring with stationarity-aware gradient rescaling to prioritize informative samples during training. Extensive experiments confirm its effectiveness. In future work, we aim to explore how to extend this method to continual forecasting where the spatial topology evolves over training time.

AI USE STATEMENT

We declare that all aspects of the conceptual development, core methodology and content, experimental methods, and analysis of this paper were independently completed by the authors without the assistance of any AI tools.

ETHICS STATEMENT

This paper presents an algorithm designed to facilitate efficient training of spatio-temporal models. Our work has many potential societal impacts, but we believe none require special mention here.

REPRODUCIBILITY STATEMENT

We used publicly available datasets for training and evaluation, and plan to open-source the code. The numerical tables in the manuscript were manually double-checked and cross-verified against the records. The authors take responsibility for the text, claims, and research findings of this study.

REFERENCES

Amit Agarwal, Bhumika Gupta, Gaurav Bhatt, and Ankush Mittal. Construction of a semi-automated model for faq retrieval via short message service. In *Proceedings of FIRE*, 2015.

Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *ECCV*, pp. 137–153. Springer, 2020.

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

Allan M Avila and Igor Mezić. Data-driven analysis and forecasting of highway traffic dynamics. *Nature communications*, 11(1):2090, 2020.

Lei Bai, Lina Yao, Can Li, Xianjie Wang, and Chunrong Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In *NeurIPS*, 2020.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4750–4759, 2022.

Ke Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.

Min Chen, Guansong Pang, Wenjun Wang, and Cheng Yan. Information bottleneck-guided mlps for robust spatial-temporal forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

Wei Chen and Yuxuan Liang. Expand and compress: Exploring tuning principles for continual spatio-temporal graph forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Wei Chen and Yuxuan Liang. Learning with calibration: Exploring test-time computing of spatio-temporal forecasting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Andrea Cini, Ivan Marisca, Daniele Zambon, and Cesare Alippi. Taming local effects in graph-based spatiotemporal forecasting. *Advances in Neural Information Processing Systems*, 36:55375–55393, 2023.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *ICLR*, 2019.

Yuchen Fang, Yuxuan Liang, Bo Hui, Zezhi Shao, Liwei Deng, Xu Liu, Xinke Jiang, and Kai Zheng. Efficient large-scale traffic forecasting with transformers: A spatial data management perspective. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 307–317, 2025.

Yisong Fu, Fei Wang, Zezhi Shao, Boyu Diao, Lin Wu, Zhulin An, Chengqing Yu, Yujie Li, and Yongjun Xu. On the integration of spatial-temporal knowledge: A lightweight approach to atmospheric time series forecasting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pp. 181–195. Springer, 2022.

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pp. 291–300, 2004.

Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic Learning Theory*, pp. 722–754. PMLR, 2021.

Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 8078–8086, 2023.

Guangyin Jin, Yuxuan Liang, Yuchen Fang, Zezhi Shao, Jincai Huang, Junbo Zhang, and Yu Zheng. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(10):5388–5408, 2023.

Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I Webb, Irwin King, and Shirui Pan. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pp. 2525–2534. PMLR, 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.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Shiye Lei and Dacheng Tao. A comprehensive survey of dataset distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):17–32, 2023.

Han Li, Haohao Qu, Xiaojun Tan, Linlin You, Rui Zhu, and Wenqi Fan. Urbanev: An open benchmark dataset for urban electric vehicle charging demand prediction. *Scientific Data*, pp. 523, 2025a. ISSN 2052-4463. doi: 10.1038/s41597-025-04874-4.

Xinglin Li, Kun Wang, Hanhui Deng, Yuxuan Liang, and Di Wu. Attend who is weak: Enhancing graph condensation via cross-free adversarial training. *arXiv preprint arXiv:2311.15772*, 2023.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *ICLR*, 2018a.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018b.

Zhonghang Li, Long Xia, Lei Shi, Yong Xu, Dawei Yin, and Chao Huang. Open spatio-temporal foundation models for traffic prediction. *ACM Transactions on Intelligent Systems and Technology*, 2025b.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 4125–4129, 2023.

Xu Liu, Yuxuan Liang, Chao Huang, Hengchang Hu, Yushi Cao, Bryan Hooi, and Roger Zimmermann. Reinventing node-centric traffic forecasting for improved accuracy and efficiency. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 21–38. Springer, 2024a.

Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. Largest: A benchmark dataset for large-scale traffic forecasting. *Advances in Neural Information Processing Systems*, 36, 2024b.

Jiaming Ma, Binwu Wang, Guanjun Wang, Kuo Yang, Zhengyang Zhou, Pengkun Wang, Xu Wang, and Yang Wang. Less but more: Linear adaptive graph learning empowering spatiotemporal forecasting. In *NeurIPS*, 2025.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*, 2021.

Brian B Moser, Arundhati S Shanbhag, Stanislav Frolov, Federico Raue, Joachim Folz, and Andreas Dengel. A coreset selection of coreset selection literature: Introduction and recent advances. *arXiv preprint arXiv:2505.17799*, 2025.

Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34:5186–5198, 2021.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. In *International Conference on Machine Learning*, pp. 25904–25938. PMLR, 2023.

Seth Ockerman, Amal Gueroudji, Tanwi Mallick, Yixuan He, Line Pouchard, Robert Ross, and Shivaram Venkataraman. Pgt-i: Scaling spatiotemporal gnns with memory-efficient distributed training. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 217–236, 2025.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34: 20596–20607, 2021.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, Yang You, et al. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In *The Twelfth International Conference on Learning Representations*, 2024.

Ravi S Raju, Kyle Daruwalla, and Mikko Lipasti. Accelerating deep learning with dynamic data pruning. *arXiv preprint arXiv:2111.12621*, 2021.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 4454–4458, 2022.

Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Guangyin Jin, Xin Cao, Gao Cong, et al. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *arXiv preprint arXiv:2310.06119*, 2023.

Zezhi Shao, Yujie Li, Fei Wang, Chengqing Yu, Yisong Fu, Tangwen Qian, Bin Xu, Boyu Diao, Yongjun Xu, and Xueqi Cheng. Blast: Balanced sampling time series corpus for universal forecasting models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2502–2513, 2025.

Jae-hun Shim, Kyeongbo Kong, and Suk-Ju Kang. Core-set sampling for efficient neural architecture search. *arXiv preprint arXiv:2107.06869*, 2021.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 914–921, 2020.

Jiabin Tang, Wei Wei, Lianghao Xia, and Chao Huang. Easyst: A simple framework for spatio-temporal prediction. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2220–2229, 2024.

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. *arXiv preprint arXiv:1812.05159*, 2018.

Binwu Wang, Pengkun Wang, Zhengyang Zhou, Zhe Zhao, Wei Xu, and Yang Wang. Make bricks with a little straw: Large-scale spatio-temporal graph learning with restricted gpu-memory capacity. In *IJCAI*, pp. 2388–2396. Jeju, South Korea, 2024.

Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12196–12205, 2022.

Kun Wang, Yuxuan Liang, Xinglin Li, Guohao Li, Bernard Ghanem, Roger Zimmermann, Huahui Yi, Yudong Zhang, Yang Wang, et al. Brave the wind and the waves: Discovering robust and generalizable graph lottery tickets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Shaobo Wang, Jiaming Wang, Jiajun Zhang, Cong Wang, Yue Min, Zichen Wen, Fei Huang, Huiqiang Jiang, Junyang Lin, Dayiheng Liu, et al. Winning the pruning gamble: A unified approach to joint sample and token pruning for efficient supervised fine-tuning. *arXiv preprint arXiv:2509.23873*, 2025.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Max Welling. Herding dynamical weights to learn. In *ICMLg*, pp. 1121–1128, 2009.

Wenchao Weng, Hanyu Jiang, Mei Wu, Xiao Han, Haidong Gao, Guojiang Shen, and Xiangjie Kong. Let's group: a plug-and-play subgraph learning method for memory-efficient spatio-temporal graph modeling. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3471–3479, 2025.

Hao Wu, Haomin Wen, Guibin Zhang, Yutong Xia, Yuxuan Liang, Yu Zheng, Qingsong Wen, and Kun Wang. Dynst: Dynamic sparse training for resource-constrained spatio-temporal forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2682–2692, 2025.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

Xiaosong Yang, Thomas L Delworth, Liwei Jia, Nathaniel C Johnson, Feiyu Lu, and Colleen McHugh. Skillful seasonal prediction of wind energy resources in the contiguous united states. *Communications Earth & Environment*, 5(1):313, 2024.

Du Yin, Hao Xue, Arian Prabowo, Shuang Ao, and Flora Salim. Xxltraffic: Expanding and extremely long traffic forecasting beyond test adaptation. In *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*, pp. 511–521, 2025.

Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *IEEE transactions on pattern analysis and machine intelligence*, 46(1):150–170, 2023.

Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. Data-centric artificial intelligence: A survey. *ACM Computing Surveys*, 57(5):1–42, 2025.

Guibin Zhang, Haonan Dong, Zhixun Li, Dingshuo Chen, Kai Wang, Tianlong Chen, Yuxuan Liang, Dawei Cheng, Kun Wang, et al. Gder: Safeguarding efficiency, balancing, and robustness via prototypical graph pruning. *Advances in Neural Information Processing Systems*, 37:50285–50312, 2024a.

Guibin Zhang, Kun Wang, Wei Huang, Yanwei Yue, Yang Wang, Roger Zimmermann, Aojun Zhou, Dawei Cheng, Jin Zeng, and Yuxuan Liang. Graph lottery ticket automated. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=nmBjBZoySX.

Yuchen Zhang, Tianle Zhang, Kai Wang, Ziyao Guo, Yuxuan Liang, Xavier Bresson, Wei Jin, and Yang You. Navigating complexity: Toward lossless graph condensation via expanding window matching. In *Forty-first International Conference on Machine Learning*, 2024c.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6514–6523, 2023.

Yiji Zhao, Zihao Zhong, Ao Wang, Haomin Wen, Ming Jin, Yuxuan Liang, Huaiyu Wan, and Hao Wu. Fast: Efficient and effective long-horizon forecasting for large-scale spatial-temporal graphs via mixture-of-experts. In *Proceedings of the 32th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026.

SUPPLEMENTARY MATERIAL

ST-PRUNE : EXPLORING DYNAMIC SAMPLE PRUNING OF SPATIO-TEMPORAL TRAINING

TABLE OF CONTENTS

A Experimental Details **15**

A.1 Dataset and Protocol Details 15

A.2 Baseline and Parameter Details 16

B More Results **19**

C More Discussion **19**

C.1 Impact Statement 19

C.2 Current Limitations 19

C.3 Future Work 20

A EXPERIMENTAL DETAILS

A.1 DATASET AND PROTOCOL DETAILS

A.1.1 DATASETS DETAILS

Our experiments are carried out on five real-world datasets from diffrent domain. The statistics of these spatio-temporal datasets are shown in Table 3.

Table 3: Summary of datasets used for our experiments. Degree: the average degree of each node. Data Points: multiplication of nodes and frames. M: million (10^6).

Source	Dataset	Nodes	Time Range	Frames	Sampling Rate	Data Points
(Song et al., 2020)	PEMS08	170	07/01/2016 – 08/31/2016	17,856	5 minutes	3.04M
(Li et al., 2025a)	UrbanEV	275	09/01/2022 – 02/28/2023	4344	1 hour	1.19M
LargeST (Liu et al., 2024b)	GLA	3,834	01/01/2019 – 12/31/2019	35,040	15 minutes	134.3M
	GBA	2,352	01/01/2019 – 12/31/2019	35,040	15 minutes	88.7M
	SD	716	01/01/2019 – 12/31/2019	35,040	15 minutes	25.1M

We adopt the standard approach widely used in prior works (Li et al., 2018a; Bai et al., 2020) to determine the graph structure prior. Specifically, we build the adjacency matrix A using a thresholded Gaussian kernel, formulated as:

$$A_{[ij]} = \begin{cases} \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right), & \text{if } \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \geq r \text{ and } i \neq j, \\ 0, & \text{otherwise,} \end{cases}$$

where d_{ij} denotes the geographical distance between sensors i and j, σ is chosen as the standard deviation of all pairwise distances, and r is a predefined similarity threshold used to sparsify the graph. For the specific parameter settings, we follow the paper corresponding to the dataset.

A.1.2 PROTOCOL DETAILS

We use different metrics such as MAE, RMSE, MAPE, and CORR. Formally, these metrics are formulated as following:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \qquad \text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|, \qquad \text{Corr} = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}},$$

where n represents the indices of all observed samples, y_i denotes the i-th actual sample, and $\hat{y}_i$ is the corresponding prediction.

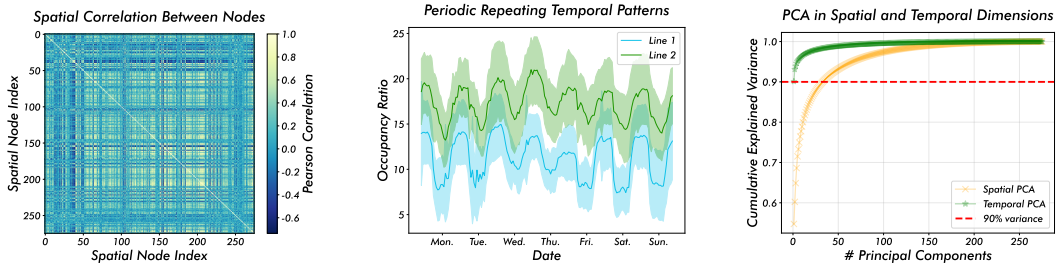


Figure 9: The ST data redundancy characteristics and statistical properties of URBANEV (Li et al., 2025a).

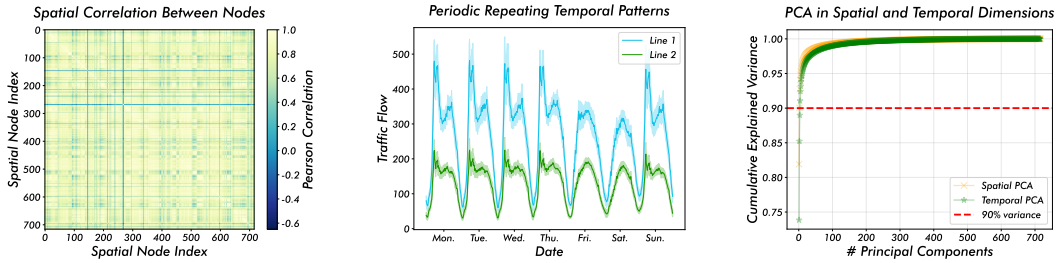


Figure 10: The ST data redundancy characteristics and statistical properties of SD (Liu et al., 2024b).

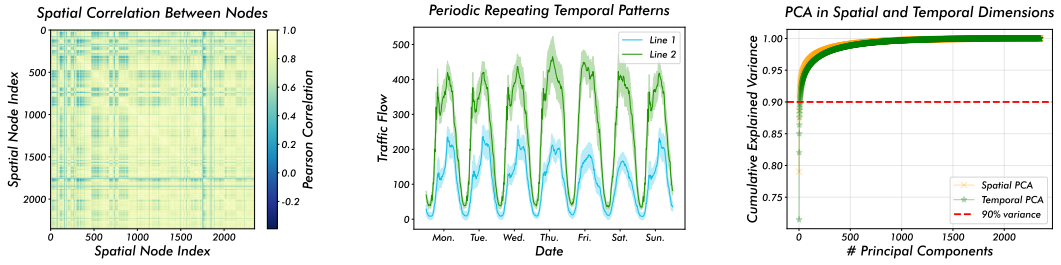


Figure 11: The ST data redundancy characteristics and statistical properties of GBA (Liu et al., 2024b).

A.2 BASELINE AND PARAMETER DETAILS

A.2.1 BASELINE DETAILS

In this subsection, we describe in detail the advanced dataset pruning and selection methods and different spatiotemporal backbone models that we use in our default evaluation.

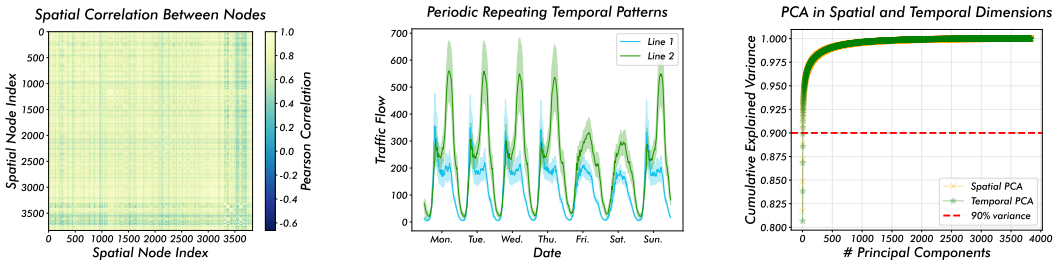


Figure 12: The ST data redundancy characteristics and statistical properties of GLA (Liu et al., 2024b).

Data Pruning and Selection Methods. To ensure fair use of these baselines, we use a popular coreset selection benchmark libraries: `https://github.com/PatrickZH/DeepCore` and adapt the respective methods to ST forecasting tasks.

- *Static Dataset Pruning*:
  - † *Random-based*:
    - ‡ **Hard Random**: Random selection before training.
  - † *Geometry-based*:
    - ‡ ***CD*** **(Agarwal et al., 2015)**: Adapted the diversity metric from image feature space to spatio-temporal feature space, using spatio-temporal embedding vectors to calculate inter-sample distances for selecting representative patterns.
    - ‡ ***Herding*** **(Welling, 2009)**: Modified prototype selection from classification labels to regression targets, finding core samples in spatio-temporal feature space that represent different trends.
    - ‡ ***K-Means*** **(Sener & Savarese, 2018)**: Adapted *K-Means* clustering from image features to spatio-temporal sequence features, clustering based on temporal patterns and spatial correlations in samples for center selection.
  - † *Uncertainty-based*:
    - ‡ ***Least Confidence*** **(Coleman et al., 2019)**:Replaced classification probability confidence with regression prediction uncertainty estimation, selecting spatio-temporal samples with maximum prediction variance.
    - ‡ ***Entropy*** **(Coleman et al., 2019)**: Adapted entropy calculation from classification probability distributions to regression prediction distributions, selecting spatio-temporal sequence samples with maximum uncertainty-based information content.
    - ‡ ***Margin*** **(Coleman et al., 2019)**: Changed from classification decision boundary distances to regression prediction interval widths, selecting spatio-temporal samples with maximum prediction confidence intervals.
  - † *Loss-based*:
    - ‡ ***Forgetting*** **(Toneva et al., 2018)**: Replaced classification accuracy forgetting statistics with regression error change statistics, tracking fluctuation patterns of spatio-temporal sequence samples' prediction errors during training.
  - † *Decision boundary-based*:
    - ‡ ***GraNd*** **(Paul et al., 2021)**: Adapted gradient norm calculation from classification loss to regression loss, selecting samples with greatest impact on the model based on spatio-temporal prediction model parameter gradient magnitudes.
  - † *Bi-level optimization-based*:
    - ‡ ***Glister*** **(Killamsetty et al., 2021)**: Replaced classification loss with regression loss in the bi-level optimization framework, selecting most valuable training samples through gradient matching for spatio-temporal prediction tasks.
  - † *Submodular function-based*:
    - ‡ ***GraphCut*** **(Iyer et al., 2021)**: Replaced similarity-based construction with spatio-temporal sequence similarity-based construction, performing partitioning based on spatio-temporal correlations to select representative samples.

‡ ***FaLo*** **(Iyer et al., 2021)**: Adapted the facility location problem from classification to regression tasks, finding facility points in spatio-temporal feature space that cover different patterns as core samples.

- *Dynamic Dataset Pruning*:
  - † *Random-based*:
    - ‡ ***Soft Random***: Random selection during training.
  - † *Uncertainty-based*:
    - ‡ ***ϵ-greedy*** **(Raju et al., 2021)**: Adapted greedy selection strategy from classification models to regression model parameter changes, dynamically selecting samples most beneficial for spatio-temporal prediction model training.
    - ‡ ***UCB*** **(Raju et al., 2021)**: Adapted multi-armed bandit classification reward mechanism to regression task rewards, dynamically balancing exploration and exploitation in sample selection based on prediction error reduction.
  - † *Loss-based*:
    - ‡ ***InfoBatch*** **(Qin et al., 2024)**: Adapted dynamic batch selection from classification loss-based to spatiotemporal regression loss-based selection, dynamically adjusting training batch composition according to the information content of spatio-temporal sequence samples.

Spatio-Temporal Neural Network. To ensure fair use of these models, we use an implementation of the popular spatio-temporal neural network benchmark library: https://github.com/liuxu77/LargeST, adapted to the above data pruning and selection repository.

- *Graph-based*:
  - † ***GWNet*** **(Wu et al., 2019)**: *GWNet* is a graph-based deep learning framework tailored for spatio-temporal forecasting. It leverages graph convolutional networks to model complex spatial correlations among nodes and employs dilated causal convolutions to capture long-range temporal dependencies, enabling effective representation of spatial-temporal dynamics.
- *MLP-based*:
  - † ***STID*** **(Shao et al., 2022)**: *STID* serves as a lightweight MLP-based baseline for spatio-temporal prediction. By explicitly incorporating spatial and temporal identity embeddings, it alleviates the indistinguishability problem in spatio-temporal data. Its parameter-efficient MLP architecture achieves competitive forecasting accuracy with low computational overhead.
- *Transformer-based*:
  - † ***STAEformer*** **(Liu et al., 2023)**: *STAEformer* is a effective transformer-based model that adapts the vanilla transformer architecture to the spatio-temporal domain. It introduces adaptive spatial-temporal embeddings and attention mechanisms to dynamically capture spatial dependencies and temporal evolution, pushing transformer models to high-competitive performance for spatio-temporal forecasting tasks.

Spatio-Temporal Foundation Model. We also applied our method to the recently popular spatio-temporal foundation models to evaluate scalability. We followed and used the open-source code repository and pre-trained datasets of OpenCity (Li et al., 2025b) from: https://github.com/HKUDS/OpenCity.

† ***OpenCity*** **(Li et al., 2025b)**: *OpenCity* is a large-scale foundation model designed for general-purpose spatio-temporal learning in urban environments. It is pre-trained on massive, heterogeneous urban data spanning multiple cities and domains, enabling strong cross-city and cross-task generalization.

A.2.2 Parameter Details

Detailed hyperparameters settings are shown in Table 4. We use the same parameter configurations for our `ST-Prune` , along with the other baseline methods according to the recommendation of previous papers (Guo et al., 2022; Liu et al., 2024b). All experiments are conducted on a Linux server equipped with a 1 × Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz (512GB memory) and 8

× NVIDIA A100 (80GB memory) GPUs. To carry out benchmark testing experiments, all baselines and our method are set to run 100 epochs by default without early stopping mechanism to ensure fairness of efficiency experiments. *The source code, experimental materials can be accessed in the anonymous repository at* https://anonymous.4open.science/r/ST-DSP.

Table 4: Hyperparameters setting.

Training epochs	100
Selection epochs	10
Batch size	256
Data Fraction	0.1 / 0.3 / 0.5 / 0.7
Optimizer	SGD
Learning rate	1e-3
Minimum Learning rate	1e-4
Momentum	0.9
Weight Decay	1e-4
Scheduler	CosineAnnealingLR
Gamma	0.5
Step Size	50
Loss Function	MAE
λ	0.5
δ	0.9

B More Results

We also provide MAE and RMSE comparisons of the performance of our method ST-Prune with the state-of-the-art dataset pruning and selection methods when $\{10\%, 30\%, 50\%, 70\%\}$ of the full set remains, as shown in Tables 5 and 6.

C More Discussion

C.1 Impact Statement

This paper presents work whose goal is to advance the field of machine learning for physical world modeling. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

C.2 Current Limitations

While ST-Prune demonstrates favorable accuracy–efficiency trade-offs, several limitations remain.

❶ *Proxy-based sample valuation.* ST-Prune uses prediction-error heterogeneity and short-window dynamic intensity as efficient proxies for sample informativeness. These quantities do not perfectly distinguish learnable dynamics from noise, and therefore do not guarantee that every rare event or local pattern is preserved. Similarly, our stationarity-aware rescaling is an empirical correction for distribution shift rather than a general guarantee of unbiased optimization.

❷ *Workload-dependent efficiency.* Dynamic scoring, sample bookkeeping, and full-data annealing introduce additional costs beyond processing the retained subset. Although these costs are offset by substantial pruning in our experiments, the net benefit can diminish for lightweight models or small datasets. Thus, the nominal retention ratio should not be interpreted directly as the end-to-end computation ratio.

❸ *Scope of validation.* Our experiments cover multiple traffic and charging-demand datasets, architectures, optimizers, prediction horizons, and the OpenCity foundation-model family. Nevertheless,

Table 5: Performance comparison to state-of-the-art dataset pruning methods when remaining $\{10\%, 30\%, 50\%, 70\%\}$ of the full set. All methods are trained using **GWNet**, and the reported metric MAE represent the average of **five runs**.

	Dataset	PEMS08 (MAE ↓)				URBANEV (MAE ↓)			
	Remaining Ratio %	10	30	50	70	10	30	50	70
Static	Hard Random	19.92$_{\downarrow 16.90\%}$	18.75$_{\downarrow 10.04\%}$	18.24$_{\downarrow 7.04\%}$	17.92$_{\downarrow 5.16\%}$	4.23$_{\downarrow 18.82\%}$	3.96$_{\downarrow 11.24\%}$	3.82$_{\downarrow 7.30\%}$	3.76$_{\downarrow 5.62\%}$
	CD (Agarwal et al., 2020)	19.99$_{\downarrow 17.31\%}$	18.50$_{\downarrow 8.57\%}$	18.12$_{\downarrow 6.34\%}$	17.86$_{\downarrow 4.81\%}$	4.34$_{\downarrow 21.91\%}$	3.95$_{\downarrow 10.96\%}$	3.81$_{\downarrow 7.02\%}$	3.79$_{\downarrow 6.46\%}$
	Herding (Welling, 2009)	20.14$_{\downarrow 18.19\%}$	18.51$_{\downarrow 8.63\%}$	18.16$_{\downarrow 6.57\%}$	17.84$_{\downarrow 4.69\%}$	4.37$_{\downarrow 22.75\%}$	3.92$_{\downarrow 10.11\%}$	3.88$_{\downarrow 8.99\%}$	3.74$_{\downarrow 5.06\%}$
	K-Means (Sener & Savarese, 2018)	20.21$_{\downarrow 18.60\%}$	18.59$_{\downarrow 9.10\%}$	18.28$_{\downarrow 7.28\%}$	17.81$_{\downarrow 4.52\%}$	4.33$_{\downarrow 21.63\%}$	4.00$_{\downarrow 12.36\%}$	3.81$_{\downarrow 7.02\%}$	3.77$_{\downarrow 5.90\%}$
	Least Confidence (Coleman et al., 2019)	20.61$_{\downarrow 20.95\%}$	19.48$_{\downarrow 14.32\%}$	18.14$_{\downarrow 6.46\%}$	18.11$_{\downarrow 6.28\%}$	4.22$_{\downarrow 18.54\%}$	4.47$_{\downarrow 25.56\%}$	3.82$_{\downarrow 7.30\%}$	3.71$_{\downarrow 4.21\%}$
	Entropy (Coleman et al., 2019)	20.65$_{\downarrow 21.19\%}$	18.49$_{\downarrow 8.51\%}$	18.10$_{\downarrow 6.22\%}$	17.88$_{\downarrow 4.93\%}$	4.32$_{\downarrow 21.35\%}$	3.96$_{\downarrow 11.24\%}$	3.81$_{\downarrow 7.02\%}$	3.76$_{\downarrow 5.62\%}$
	Margin (Coleman et al., 2019)	20.42$_{\downarrow 19.84\%}$	18.62$_{\downarrow 9.27\%}$	18.13$_{\downarrow 6.40\%}$	17.97$_{\downarrow 5.46\%}$	4.25$_{\downarrow 19.38\%}$	4.02$_{\downarrow 12.92\%}$	4.00$_{\downarrow 12.36\%}$	3.72$_{\downarrow 4.49\%}$
	Forgetting (Toneva et al., 2018)	19.77$_{\downarrow 16.02\%}$	18.84$_{\downarrow 10.56\%}$	18.16$_{\downarrow 6.57\%}$	17.81$_{\downarrow 4.52\%}$	4.24$_{\downarrow 19.10\%}$	3.97$_{\downarrow 11.52\%}$	3.80$_{\downarrow 6.74\%}$	3.74$_{\downarrow 5.06\%}$
	GraNd (Paul et al., 2021)	20.07$_{\downarrow 17.78\%}$	18.48$_{\downarrow 8.45\%}$	18.30$_{\downarrow 7.39\%}$	17.85$_{\downarrow 4.75\%}$	4.24$_{\downarrow 19.10\%}$	3.99$_{\downarrow 12.08\%}$	3.85$_{\downarrow 8.15\%}$	3.78$_{\downarrow 6.18\%}$
	Cal (Margatina et al., 2021)	19.71$_{\downarrow 15.67\%}$	18.66$_{\downarrow 9.51\%}$	18.77$_{\downarrow 10.15\%}$	18.14$_{\downarrow 6.46\%}$	4.22$_{\downarrow 18.54\%}$	3.99$_{\downarrow 12.08\%}$	3.83$_{\downarrow 7.58\%}$	3.75$_{\downarrow 5.34\%}$
	Glister (Killamsetty et al., 2021)	20.71$_{\downarrow 21.54\%}$	19.10$_{\downarrow 12.09\%}$	18.87$_{\downarrow 10.74\%}$	18.40$_{\downarrow 7.98\%}$	4.29$_{\downarrow 20.51\%}$	4.13$_{\downarrow 16.01\%}$	3.90$_{\downarrow 9.55\%}$	3.78$_{\downarrow 6.18\%}$
	GraphCut (Iyer et al., 2021)	20.21$_{\downarrow 18.60\%}$	18.55$_{\downarrow 8.86\%}$	18.01$_{\downarrow 5.69\%}$	17.78$_{\downarrow 4.34\%}$	4.29$_{\downarrow 20.51\%}$	3.95$_{\downarrow 10.96\%}$	3.84$_{\downarrow 7.87\%}$	3.75$_{\downarrow 5.34\%}$
	FaLo (Iyer et al., 2021)	19.97$_{\downarrow 17.19\%}$	18.57$_{\downarrow 8.98\%}$	18.19$_{\downarrow 6.75\%}$	17.95$_{\downarrow 5.34\%}$	4.36$_{\downarrow 22.47\%}$	3.96$_{\downarrow 11.24\%}$	3.85$_{\downarrow 8.15\%}$	3.79$_{\downarrow 6.46\%}$
Dynamic	Soft Random	21.57$_{\downarrow 26.58\%}$	18.86$_{\downarrow 10.68\%}$	18.49$_{\downarrow 8.51\%}$	17.57$_{\downarrow 3.11\%}$	4.25$_{\downarrow 19.38\%}$	3.86$_{\downarrow 8.43\%}$	3.74$_{\downarrow 5.06\%}$	3.69$_{\downarrow 3.65\%}$
	ϵ-greedy (Raju et al., 2021)	20.25$_{\downarrow 18.84\%}$	18.82$_{\downarrow 10.45\%}$	18.43$_{\downarrow 8.16\%}$	17.97$_{\downarrow 5.46\%}$	4.15$_{\downarrow 16.57\%}$	3.84$_{\downarrow 7.87\%}$	3.85$_{\downarrow 8.15\%}$	3.77$_{\downarrow 5.90\%}$
	UCB (Raju et al., 2021)	20.24$_{\downarrow 18.78\%}$	18.95$_{\downarrow 11.21\%}$	18.34$_{\downarrow 7.63\%}$	18.05$_{\downarrow 5.93\%}$	4.14$_{\downarrow 16.29\%}$	3.83$_{\downarrow 7.58\%}$	3.82$_{\downarrow 7.30\%}$	3.71$_{\downarrow 4.21\%}$
	InfoBatch (Qin et al., 2024)	19.03$_{\downarrow 11.68\%}$	18.55$_{\downarrow 8.86\%}$	18.07$_{\downarrow 6.04\%}$	17.92$_{\downarrow 5.16\%}$	4.03$_{\downarrow 13.20\%}$	3.99$_{\downarrow 12.08\%}$	3.67$_{\downarrow 3.08\%}$	3.66$_{\downarrow 2.81\%}$
	ST-Prune (Our)	**18.32**$_{\downarrow 7.51}$	**17.91**$_{\downarrow 5.11}$	**17.63**$_{\downarrow 3.46}$	**17.63**$_{\downarrow 3.46}$	**3.63**$_{\downarrow 1.97}$	**3.62**$_{\downarrow 1.69}$	**3.58**$_{\downarrow 0.56}$	**3.56**$_{\uparrow 0.00}$
	Whole Dataset	17.04$_{\pm 0.23}$				3.56$_{\pm 0.09}$			

Table 6: Performance comparison to state-of-the-art dataset pruning methods when remaining $\{10\%, 30\%, 50\%, 70\%\}$ of the full set. All methods are trained using **GWNet**, and the reported metric RMSE represent the average of **five runs**.

	Dataset	PEMS08 (RMSE ↓)				URBANEV (RMSE ↓)			
	Remaining Ratio %	10	30	50	70	10	30	50	70
Static	Hard Random	31.29$_{\downarrow 16.28\%}$	29.24$_{\downarrow 8.66\%}$	28.72$_{\downarrow 6.73\%}$	28.29$_{\downarrow 5.13\%}$	8.24$_{\downarrow 22.07\%}$	7.73$_{\downarrow 14.52\%}$	7.35$_{\downarrow 8.89\%}$	7.24$_{\downarrow 7.26\%}$
	CD (Agarwal et al., 2020)	31.31$_{\downarrow 16.35\%}$	29.04$_{\downarrow 7.92\%}$	28.61$_{\downarrow 6.32\%}$	28.10$_{\downarrow 4.42\%}$	8.42$_{\downarrow 24.74\%}$	7.69$_{\downarrow 13.93\%}$	7.37$_{\downarrow 9.19\%}$	7.28$_{\downarrow 7.85\%}$
	Herding (Welling, 2009)	31.15$_{\downarrow 15.76\%}$	29.16$_{\downarrow 8.36\%}$	28.48$_{\downarrow 5.83\%}$	28.04$_{\downarrow 4.20\%}$	8.60$_{\downarrow 27.41\%}$	7.60$_{\downarrow 12.59\%}$	7.52$_{\downarrow 11.41\%}$	7.14$_{\downarrow 5.78\%}$
	K-Means (Sener & Savarese, 2018)	31.50$_{\downarrow 17.06\%}$	29.23$_{\downarrow 8.62\%}$	28.74$_{\downarrow 6.80\%}$	28.05$_{\downarrow 4.24\%}$	8.41$_{\downarrow 24.59\%}$	7.78$_{\downarrow 15.26\%}$	7.38$_{\downarrow 9.33\%}$	7.21$_{\downarrow 6.81\%}$
	Least Confidence (Coleman et al., 2019)	32.36$_{\downarrow 20.25\%}$	30.11$_{\downarrow 11.89\%}$	28.54$_{\downarrow 6.06\%}$	28.43$_{\downarrow 5.65\%}$	8.29$_{\downarrow 22.81\%}$	8.65$_{\downarrow 28.15\%}$	7.36$_{\downarrow 9.04\%}$	7.07$_{\downarrow 4.74\%}$
	Entropy (Coleman et al., 2019)	32.46$_{\downarrow 20.62\%}$	29.01$_{\downarrow 7.80\%}$	28.48$_{\downarrow 5.83\%}$	28.18$_{\downarrow 4.72\%}$	8.39$_{\downarrow 24.30\%}$	7.68$_{\downarrow 13.78\%}$	7.37$_{\downarrow 9.19\%}$	7.28$_{\downarrow 7.85\%}$
	Margin (Coleman et al., 2019)	32.18$_{\downarrow 19.58\%}$	29.20$_{\downarrow 8.51\%}$	28.55$_{\downarrow 6.09\%}$	28.32$_{\downarrow 5.24\%}$	8.20$_{\downarrow 21.48\%}$	7.83$_{\downarrow 16.00\%}$	7.56$_{\downarrow 12.00\%}$	7.10$_{\downarrow 5.19\%}$
	Forgetting (Toneva et al., 2018)	30.91$_{\downarrow 14.86\%}$	29.66$_{\downarrow 10.22\%}$	28.66$_{\downarrow 6.50\%}$	28.08$_{\downarrow 4.35\%}$	8.22$_{\downarrow 21.78\%}$	7.74$_{\downarrow 14.67\%}$	7.38$_{\downarrow 9.33\%}$	7.21$_{\downarrow 6.81\%}$
	GraNd (Paul et al., 2021)	31.08$_{\downarrow 15.50\%}$	29.01$_{\downarrow 7.80\%}$	28.70$_{\downarrow 6.65\%}$	28.22$_{\downarrow 4.87\%}$	8.23$_{\downarrow 21.93\%}$	7.76$_{\downarrow 14.96\%}$	7.42$_{\downarrow 9.93\%}$	7.31$_{\downarrow 8.30\%}$
	Cal (Margatina et al., 2021)	30.57$_{\downarrow 13.60\%}$	29.34$_{\downarrow 9.03\%}$	29.20$_{\downarrow 8.51\%}$	28.48$_{\downarrow 5.83\%}$	8.29$_{\downarrow 22.81\%}$	7.74$_{\downarrow 14.67\%}$	7.36$_{\downarrow 9.04\%}$	7.20$_{\downarrow 6.67\%}$
	Glister (Killamsetty et al., 2021)	32.32$_{\downarrow 20.10\%}$	30.20$_{\downarrow 12.23\%}$	29.83$_{\downarrow 10.85\%}$	29.30$_{\downarrow 8.88\%}$	8.23$_{\downarrow 21.93\%}$	8.06$_{\downarrow 19.41\%}$	7.60$_{\downarrow 12.59\%}$	7.34$_{\downarrow 8.74\%}$
	GraphCut (Iyer et al., 2021)	31.35$_{\downarrow 16.50\%}$	29.19$_{\downarrow 8.47\%}$	28.37$_{\downarrow 5.43\%}$	28.11$_{\downarrow 4.46\%}$	8.47$_{\downarrow 25.48\%}$	7.67$_{\downarrow 13.63\%}$	7.44$_{\downarrow 10.22\%}$	7.15$_{\downarrow 5.93\%}$
	FaLo (Iyer et al., 2021)	31.04$_{\downarrow 15.35\%}$	29.22$_{\downarrow 8.58\%}$	28.59$_{\downarrow 6.24\%}$	28.32$_{\downarrow 5.24\%}$	8.40$_{\downarrow 24.44\%}$	7.63$_{\downarrow 13.04\%}$	7.46$_{\downarrow 10.52\%}$	7.26$_{\downarrow 7.56\%}$
Dynamic	Soft Random	32.71$_{\downarrow 21.55\%}$	29.44$_{\downarrow 9.40\%}$	28.82$_{\downarrow 7.10\%}$	27.86$_{\downarrow 3.53\%}$	8.29$_{\downarrow 22.81\%}$	7.45$_{\downarrow 10.37\%}$	7.19$_{\downarrow 6.52\%}$	7.24$_{\downarrow 7.26\%}$
	ϵ-greedy (Raju et al., 2021)	29.94$_{\downarrow 11.26\%}$	29.64$_{\downarrow 10.14\%}$	28.74$_{\downarrow 6.80\%}$	27.96$_{\downarrow 3.90\%}$	8.18$_{\downarrow 21.19\%}$	7.47$_{\downarrow 10.67\%}$	7.51$_{\downarrow 11.26\%}$	7.33$_{\downarrow 8.59\%}$
	UCB (Raju et al., 2021)	29.93$_{\downarrow 11.22\%}$	29.87$_{\downarrow 11.00\%}$	28.69$_{\downarrow 6.61\%}$	28.04$_{\downarrow 4.20\%}$	8.15$_{\downarrow 20.74\%}$	7.46$_{\downarrow 10.52\%}$	7.47$_{\downarrow 10.67\%}$	7.30$_{\downarrow 8.15\%}$
	InfoBatch (Qin et al., 2024)	30.16$_{\downarrow 12.08\%}$	29.48$_{\downarrow 9.55\%}$	28.78$_{\downarrow 6.94\%}$	28.48$_{\downarrow 5.83\%}$	7.54$_{\downarrow 11.70\%}$	7.46$_{\downarrow 10.52\%}$	7.02$_{\downarrow 4.00\%}$	6.95$_{\downarrow 2.96\%}$
	ST-Prune (Our)	**28.37**$_{\downarrow 5.46}$	**27.89**$_{\downarrow 3.64}$	**27.61**$_{\downarrow 2.60}$	**27.71**$_{\downarrow 2.97}$	**6.91**$_{\downarrow 2.37}$	**6.89**$_{\downarrow 2.07}$	**6.75**$_{\downarrow 0.00}$	**6.75**$_{\uparrow 0.00}$
	Whole Dataset	26.91$_{\pm 0.23}$				6.75$_{\pm 0.09}$			

these results do not establish universal effectiveness across other physical systems or dynamically evolving sensor networks. In particular, changing node coverage may affect the comparability of spatial complexity scores and require additional adaptation.

C.3 FUTURE WORK

❶ *Adaptive sample selection.* A promising direction is to replace handcrafted scoring and rescaling rules with lightweight learnable policies that adapt to training dynamics, while explicitly accounting for their additional computational cost.

❷ *Broader and dynamic settings.* Future work will extend ST-Prune to additional physical forecasting domains and continual or streaming scenarios with distribution shifts and evolving spatial topology. Evaluating rare-event performance and time-to-quality will further clarify when dynamic pruning provides the largest practical benefit.