TexTSC: Class-Texture Preserving Data Condensation for Time Series Classification

Pouya Hosseinzadeh Utah State University
Peiyu Li Utah State University
Omar Bahri Utah State University
Soukaina Filali Boubrahimi Utah State University
Shah Muhammad Hamdi Utah State University

Abstract

Dataset condensation seeks to generate a small set of synthetic examples that can replace large real datasets for training, but existing methods for time series often rely on unstable training-trajectory matching or capture only limited signal structure. We present TexTSC, a condensation framework that preserves class structure using spectro-temporal second-order statistics instead of trajectory replay. TexTSC models each class's "texture" as the co-activation pattern among intermediate teacher features, aligning Gram matrices of activations in time to capture temporal correlations and in frequency to capture spectral envelopes and harmonics. A short-lag autocorrelation term stabilizes local rhythm, while a lightweight gradient anchor at the final layer ensures discriminative power. TexTSC optimizes synthetic sequences directly, remains model-agnostic, and requires only closed-form statistics, making it simple and stable. Experiments on standard benchmarks show that TexTSC produces compact datasets that retain class-conditional structure and achieve higher classification accuracy than first-order or single-domain baselines.

1 INTRODUCTION

The proliferation of time series data across diverse fields such as traffic forecasting (Li et al., 2017; Liang et al., 2023; Liu et al., 2024a, 2023a,b, 2024c), clinical diagnosis (Harutyunyan et al., 2019; Hayat et al., 2022), financial analysis (Cheng et al., 2022; Kumbure et al., 2022), hydrology (Hosseinzadeh et al., 2023) and space weather forecasting (Hosseinzadeh et al., 2024) has created significant opportunities for deep learning applications. However, the sheer volume and complexity of this data present substantial challenges for training deep neural networks, particularly in terms of computational cost, memory usage, and training time. For tasks like neural architecture search (Rakhshani et al., 2020) and continual learning (Gupta et al., 2021), leveraging the entire dataset is often inefficient and impractical. This has created an urgent need for effective strategies to condense large-scale time series datasets while preserving the critical information necessary for training high-performance models.

To address this challenge, dataset condensation (Wang et al., 2020; Zhao et al., 2020) has emerged as a promising technique. The fundamental goal is to synthesize a small, compact dataset that, when used for training, can replicate the performance of a model trained on the full, original dataset. This approach is distinct from traditional data compression, as its primary aim is to create a dataset that trains models to generalize effectively to unseen data, rather than simply reducing storage size. Existing dataset condensation strategies have shown considerable success, with methods that focus on maximizing test performance (Deng and Russakovsky, 2022; Liu et al., 2022b; Loo et al., 2023; Nguyen et al., 2020, 2021; Wang et al., 2020; Zhou et al., 2022b), matching training gradients (Cazenavette et al., 2022; Cui et al., 2022; Du et al., 2023; Jin et al., 2022, 2021; Kim et al., 2022; Lee et al., 2022b; Li et al., 2022; Zhang et al., 2023; Zhao and Bilen, 2021; Zhao et al., 2020), and aligning feature distributions (Lee et al., 2022a; Wang et al., 2022; Zhao and Bilen, 2023).

However, the majority of current condensation meth-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

ods have been developed and optimized for image (Deng and Russakovsky, 2022; Lee et al., 2022b; Nguyen et al., 2021; Wang et al., 2022; Zhou et al., 2022b) or graph (Jin et al., 2022, 2021; Liu et al., 2022a; Sachdeva et al., 2022) data. Directly applying these techniques to time series data often yields suboptimal results because they fail to account for the unique, defining characteristics of sequential data. Unlike images, the discriminative information in time series is inherently spectro-temporal; features such as local rhythms, periodic patterns, and frequency-domain harmonics are crucial for effective classification. While many works have highlighted the importance of the frequency domain for time series analysis (Woo et al., 2022; Yang and Hong, 2022; Zhang et al., 2022a,b; Zhou et al., 2022a), existing condensation methods often overlook these properties.

In this paper, we introduce TexTSC, a novel dataset condensation framework designed specifically for time series classification. TexTSC operates by preserving the spectro-temporal "texture" of each class, which we define as the second-order statistical patterns of co-activations across intermediate layers of a feature-extractor network. Instead of relying on expensive training trajectory matching, our method focuses on matching these class-conditional statistics in both the time and frequency domains. Specifically, TexTSC matches the Gram matrices of intermediate features to preserve channel correlations in the time domain, while simultaneously matching the Gram matrices of their Fourier transform magnitudes to maintain shared spectral structures. This is complemented by a short-lag autocorrelation constraint to preserve local rhythm and a lightweight gradient anchor on the final classification layer to ensure the synthetic samples remain discriminative.

Through extensive experiments on standard time series classification benchmarks, we demonstrate that TexTSC significantly outperforms baseline methods. The condensed datasets generated by our framework not only retain the essential class-conditional structure of the original data but also lead to improved downstream classification accuracy.

Our contributions can be summarized as follows:

- We propose a novel, spectro-temporal objective for time series dataset condensation that preserves the second-order "texture" of classes by matching Gram matrix statistics of intermediate features in both the time and frequency domains.
- We introduce a simple, stable, and model-agnostic optimization procedure that combines spectro-temporal matching with a short-lag autocorrelation constraint and a lightweight final-layer gradient anchor, avoiding the need for complex training trajectory alignment.
- We validate the effectiveness of TexTSC through comprehensive experiments, showing that it produces compact and informative datasets that significantly narrow the performance gap to full-data training and are robust even at extremely small data budgets.

2 RELATED WORK

2.1 Dataset condensation and coreset selection

Learning from a tiny set of synthetic or selected samples has been explored along two major lines. Dataset condensation methods optimize a small synthetic set to stand in for the original training set by differentiating through the training dynamics of a student network. Early approaches anchored synthetic batches to the gradients of a reference model on real data, while follow-ups matched longer training trajectories or distributional statistics of features to stabilize optimization and improve cross-model transfer (Cazenavette et al., 2022; Kim et al., 2022; Zhao and Bilen, 2021, 2023). In parallel, coreset selection chooses a subset of real examples using clustering, k-center, herding, or submodular criteria. Coresets preserve fidelity but are constrained by duplicates, class imbalance, and redundancy at low budgets; condensation can synthesize informative variations but may be fragile without good inductive biases. TexTSC belongs to the condensation family and introduces spectro-temporal objectives tailored to time series.

2.2 Time-series condensation and distillation

Despite rapid progress in images, condensation for time series is comparatively underexplored. Several works adapt vision objectives to 1D signals, reporting limited transfer when temporal structure and channel correlations are ignored (Zhao and Bilen, 2021; Cazenavette et al., 2022). A recent line targets time-series classification specifically, combining per-layer feature alignment with simple temporal constraints (Liu et al., 2024b). Our method is most closely related to this direction, but differs in two key aspects: (i) we match second-order Gram statistics concurrently in the time domain and in the frequency domain at multiple intermediate taps of a teacher, and (ii) we add a lightweight last-layer gradient anchor that steers synthetic samples toward discriminative cues without dominating the spectro-temporal objectives. This combination improves stability at tiny sample per class

(SPC) budgets and yields condensed sets that generalize across random seeds and student initializations.

2.3 Spectro-temporal representations and perceptual objectives

For natural signals, second-order statistics computed from intermediate features serve as robust texture descriptors. In images, Gram matrices over convolutional activations underpin classical style-transfer objectives. Extensions to audio and 1D signals commonly rely on spectral magnitudes or modulation statistics to capture periodicity and spectral structure. We adopt this principle for time-series classification by matching Gram matrices both in the time domain and on the magnitude of the real-valued Fourier coefficients, complemented by a short-lag autocorrelation term that constrains rhythm and local periodicity. Several studies have underscored the value of the frequency domain in time series analysis by using it to learn more robust representations (Woo et al., 2022; Yang and Hong, 2022; Zhang et al., 2022b; Zhou et al., 2022a). Unlike prior time-series objectives that rely on a single domain or first-order moments, TexTSC uses a multi-tap, multi-domain second-order alignment that preserves class-specific morphology while remaining numerically stable under mixed precision.

2.4 Gradient and training-dynamics matching

Gradient-based anchors have proven effective for condensation: aligning synthetic and real gradients at shared network parameters yields class-discriminative signals even with very small synthetic batches (Zhao and Bilen, 2021; Zhao et al., 2020). Trajectory-matching methods unroll multiple steps of the student optimizer, but suffer from memory and stability issues (Cazenavette et al., 2022; Du et al., 2023). TexTSC adopts a one-step gradient anchor restricted to the teacher's final linear layer, which is cheap, easy to differentiate, and empirically sufficient to discourage degenerate textures. This design preserves the dominance of spectro-temporal objectives while injecting a minimal discriminative bias.

2.5 Augmentation and priors for time series

Classical time-series augmentation includes jittering, scaling, time masking, and frequency-domain perturbations (Wen et al., 2020). These operations improve generalization but do not reduce dataset size. Condensation offers complementary benefits: it compresses the dataset while still allowing augmentations during student training. We use light augmentations for both fairness and stability, similar to prior work on condensed training.

2.6 Proxy data for architecture and hyperparameter search

Proxy datasets are widely used to accelerate neural architecture search (NAS) and hyperparameter tuning (Elsken et al., 2019). Condensed sets offer a task-aware proxy distilled from the original training distribution, and recent studies in vision suggest competitive rank fidelity relative to full-data search (Zhao and Bilen, 2021, 2023). TexTSC evaluates this idea in time-series classification: we rank a compact CNN search space on the condensed set and validate the top candidates on real data. The resulting accuracy and rank correlation indicate that spectro-temporal condensation retains enough signal to guide architecture choices at a fraction of the cost.

3 METHOD

In this section, we formally define the problem of time-series dataset condensation and detail the components of our proposed TexTSC framework. Our approach is centered on a novel objective function that preserves class-conditional spectro-temporal textures by matching second-order statistics, guided by a minimal discriminative anchor. The full objective is a weighted sum of several distinct loss terms, which we will define systematically. Our model's diagram is shown in Figure 1. Our source codes are available at: https://github.com/pouyahosseinzadeh/TexTSC.

3.1 Problem Setting and Notation

We consider a labeled time-series dataset D as defined in Equation 1.

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}, \quad x_i \in \mathbb{R}^{C\times L},\ y_i \in \{1, \dots, K\} \quad (1)$$

and seek a compact, class-balanced synthetic set with SPC samples per class as defined in Equation 2

$$\mathcal{S} = \{\, s_{c,j} \in \mathbb{R}^{C\times L} \mid c \in \{1, \dots, K\},\ j = 1, \dots, \mathrm{SPC}\,\} \quad (2)$$

A student classifier trained on $\mathcal{S}$ should approach the accuracy achieved when trained on $\mathcal{D}$. We employ a lightweight teacher T_θ (a three-block 1D CNN), pre-trained on $\mathcal{D}$, to define class-wise statistical targets. For an input x, let the three intermediate "taps" be:

$$h_\ell(x) \in \mathbb{R}^{C_\ell \times T_\ell}, \quad \ell \in \{1, 2, 3\} \quad (3)$$

taken after each residual block. We denote the class-conditioned subset of the real data by $\mathcal{D}_c = \{x_i : y_i = c\}$.

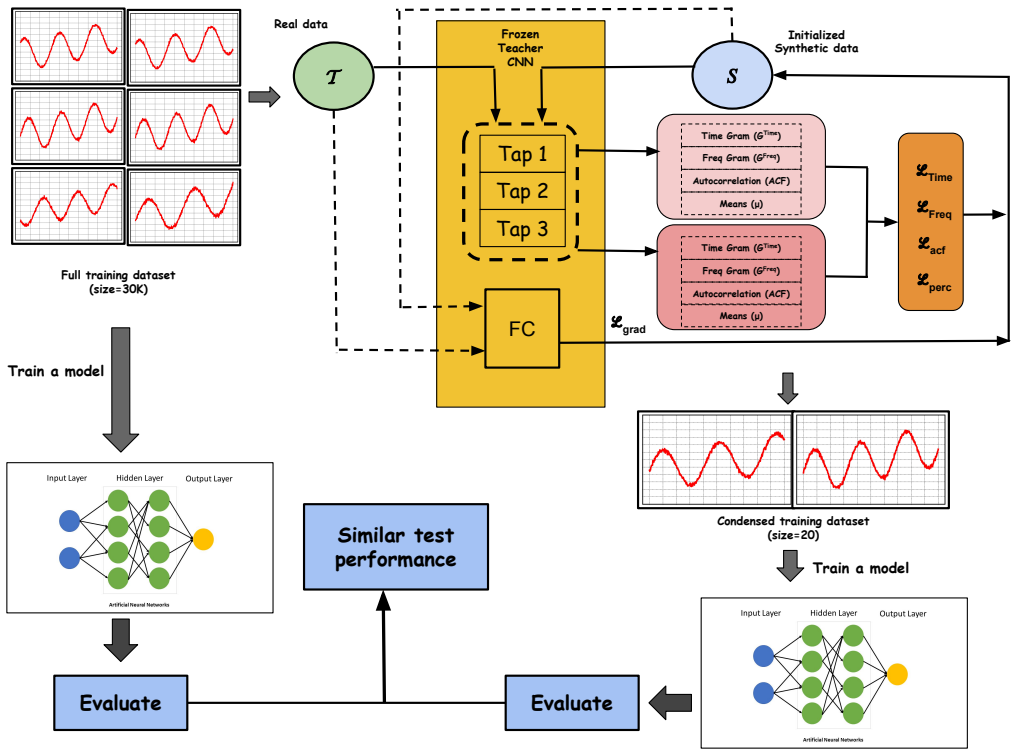

Figure 1: TexTSC data condensation diagram.

3.2 Teacher-Taps Statistics as Condensation Targets

The core of TexTSC is to match a suite of statistics derived from the teacher network's intermediate taps. These targets are designed to capture texture, rhythm, and other perceptual qualities of the time series data for each class.

Time-Domain Gram of Taps. To capture channel cross-correlations over time, we compute the Gram matrix of the vectorized feature taps. For a tap map $h \in \mathbb{R}^{C\times T}$, its time-domain Gram matrix is:

$$G^{\text{time}}(h) = \frac{1}{CT}\,\text{vec}(h)\,\text{vec}(h)^\top \tag{4}$$

The target for each class is the expected Gram matrix, computed by averaging over a balanced batch from the real data $\mathcal{D}_c$ as shown in Equation 5

$$G^{\text{time}}_{\ell,c} = \mathbb{E}_{x\sim\mathcal{D}_c}\big[\,G^{\text{time}}\big(h_\ell(x)\big)\,\big] \tag{5}$$

Frequency-Domain Gram of Taps. To preserve spectral characteristics, we compute Gram matrices on the magnitude of the real-valued Fast Fourier Transform (rFFT). Let $\tilde{h} = |\text{rFFT}(h)| \in \mathbb{R}^{C\times\tilde{T}}$. The frequency-domain Gram matrix is:

$$G^{\text{freq}}(\tilde{h}) = \frac{1}{C\tilde{T}}\,\text{vec}(\tilde{h})\,\text{vec}(\tilde{h})^\top \tag{6}$$

The corresponding class target is the expected frequency-domain Gram matrix:

$$G^{\text{freq}}_{\ell,c} = \mathbb{E}_{x\sim\mathcal{D}_c}\big[\,G^{\text{freq}}\big(h_\ell(x)\big)\,\big] \tag{7}$$

Short-Lag Autocorrelation. To maintain local rhythm in the raw waveform, we compute the channel-averaged, variance-normalized autocorrelation for lags $\tau \in \{1,\dots,\tau_{\max}\}$ in Equation 8.

$$\text{ACF}(x)[\tau] = \frac{1}{C}\sum_{k=1}^{C}\frac{\sum_{t=1}^{L-\tau}\big(x_{k,t}-\bar{x}_k\big)\big(x_{k,t+\tau}-\bar{x}_k\big)}{\sum_{t=1}^{L}\big(x_{k,t}-\bar{x}_k\big)^2+\varepsilon} \tag{8}$$

where $\bar{x}_k$ is the mean of channel k. The class-level target is the expected ACF vector:

$$\text{ACF}_c = \mathbb{E}_{x\sim\mathcal{D}_c}\big[\,\text{ACF}(x)\,\big] \tag{9}$$

Perceptual Means of Taps. Finally, we stabilize the process by matching a simple first-order statistic—the mean activation value at each tap:

$$\mu_{\ell,c} = \mathbb{E}_{x\sim\mathcal{D}_c}\Big[\,\text{mean}_{(C,T)}\big(h_\ell(x)\big)\,\Big] \tag{10}$$

3.3 The TexTSC Objective Function

The full objective function synthesizes a dataset $\mathcal{S}$ by minimizing a per-class loss $\mathcal{L}_c$ for each class $c = 1,\dots,K$. Let $S_c = \{s_{c,j}\}_{j=1}^{\text{SPC}}$ be the set of synthetic samples for class c.

Algorithm 1: TexTSC

Input: Training set $\mathcal{D}_{\text{tr}} = \{(x_i, y_i)\}_{i=1}^{N}$, validation set $\mathcal{D}_{\text{val}}$, classes $\mathcal{C}$, SPC m, teacher T, student f, iterations T_{cond}.
Input: Weights $w_{\text{time}}, w_{\text{freq}}, w_{\text{acf}}, w_{\text{perc}}, w_{\text{tv}}, w_{\text{div}}, w_{\text{grad}}$; per-class pool B_{pc}; mini-batches M_s, M_r.
Output: Condensed set $\tilde{\mathcal{D}} = \{(\tilde{x}_{c,j}, c) \mid c \in \mathcal{C},\ j = 1, \ldots, m\}$.

Train teacher
Train T on $\mathcal{D}_{\text{tr}}$ with cross-entropy; early-stop on $\mathcal{D}_{\text{val}}$
Per-class targets (teacher taps)
foreach $c \in \mathcal{C}$ **do**
  Draw $X_c \subset \{x : (x, y) \in \mathcal{D}_{\text{tr}},\ y = c\}$ with $|X_c| \le B_{\text{pc}}$
  Compute taps: $H_c^{(1)}, H_c^{(2)}, H_c^{(3)} \leftarrow \text{taps}(T, X_c)$
  $G_{c,\ell}^{\text{time}} \leftarrow \text{Gram}\big(H_c^{(\ell)}\big)$, $G_{c,\ell}^{\text{freq}} \leftarrow \text{Gram}\big(|\mathcal{F}(H_c^{(\ell)})|\big)$
  $r_c \leftarrow \frac{1}{|X_c|} \sum_{x \in X_c} \text{ACF}(x)$, $\mu_{c,\ell} \leftarrow \text{mean}_{\text{chan,time}}\big(H_c^{(\ell)}\big)$
  Form a small real bank $R_c \subset \{x : (x, y) \in \mathcal{D}_{\text{tr}},\ y = c\}$
Initialize synthetic sequences
foreach $c \in \mathcal{C}$ **do**
  Initialize $\tilde{X}_c = \{\tilde{x}_{c,1}, \ldots, \tilde{x}_{c,m}\}$ by KMeans centers on X_c (or random picks) plus noise
Set $\tilde{\mathcal{D}} \leftarrow \bigcup_{c \in \mathcal{C}} \{(\tilde{x}_{c,j}, c)\}_{j=1}^{m}$ and treat $\tilde{x}_{c,j}$ as learnable parameters
Condensation loop
for $t = 1$ **to** T_{cond} **do**
  Warm up then cosine decay the learning rate
  $\mathcal{L}_{\text{tot}} \leftarrow 0$
  foreach $c \in \mathcal{C}$ **do**
    Sample $\tilde{X}_c^{(s)} \subset \tilde{X}_c$ of size M_s and $X_c^{(r)} \subset R_c$ of size M_r
    $h_s^{(\ell)} \leftarrow \text{taps}(T, \tilde{X}_c^{(s)})$
    `// Spectro temporal/style terms`
    $\mathcal{L}_{\text{time}} \leftarrow \sum_\ell \big\|\text{Gram}(h_s^{(\ell)}) - G_{c,\ell}^{\text{time}}\big\|_F^2$
    $\mathcal{L}_{\text{freq}} \leftarrow \sum_\ell \big\|\text{Gram}(|\mathcal{F}(h_s^{(\ell)})|) - G_{c,\ell}^{\text{freq}}\big\|_F^2$
    $\mathcal{L}_{\text{acf}} \leftarrow \big\|\text{ACF}(\tilde{X}_c^{(s)}) - r_c\big\|_2^2$
    $\mathcal{L}_{\text{perc}} \leftarrow \sum_\ell \big\|\text{mean}(h_s^{(\ell)}) - \mu_{c,\ell}\big\|_2^2$
    `// Light regularizers`
    $\mathcal{L}_{\text{tv}} \leftarrow \text{TV}(\tilde{X}_c^{(s)})$, $\mathcal{L}_{\text{div}} \leftarrow \text{AvgOffDiag}\big(\cos\_\text{sim}(\tilde{X}_c^{(s)})\big)$
    `// Decision alignment (last-layer gradient anchor)`
    $g_s \leftarrow \nabla_{W_{\text{fc}}} \text{CE}\big(T(\tilde{X}_c^{(s)}), c\big)$, $g_r \leftarrow \nabla_{W_{\text{fc}}} \text{CE}\big(T(X_c^{(r)}), c\big)$ (stop-grad)
    $\mathcal{L}_{\text{grad}} \leftarrow \|g_s - g_r\|_2^2$
    `// Aggregate`
    $\mathcal{L}_c \leftarrow w_{\text{time}}\mathcal{L}_{\text{time}} + w_{\text{freq}}\mathcal{L}_{\text{freq}} + w_{\text{acf}}\mathcal{L}_{\text{acf}} + w_{\text{perc}}\mathcal{L}_{\text{perc}} + w_{\text{tv}}\mathcal{L}_{\text{tv}} + w_{\text{div}}\mathcal{L}_{\text{div}} + w_{\text{grad}}\mathcal{L}_{\text{grad}}$
    $\mathcal{L}_{\text{tot}} \leftarrow \mathcal{L}_{\text{tot}} + \mathcal{L}_c$
  Update all $\tilde{x}_{c,j}$ with AdamW on $\mathcal{L}_{\text{tot}}$
return $\tilde{\mathcal{D}}$

Data-Statistics Matching Losses. We define four loss terms corresponding to the statistical targets defined above. The losses are the squared Frobenius or Euclidean distances between the mean statistics of the synthetic set S_c and the pre-computed targets:

$$\mathcal{L}_c^{\text{time}} = \sum_{\ell=1}^{3} \big\|\mathbb{E}_{s \sim S_c}[G^{\text{time}}(h_\ell(s))] - G_{\ell,c}^{\text{time}}\big\|_F^2 \tag{11}$$

$$\mathcal{L}_c^{\text{freq}} = \sum_{\ell=1}^{3} \big\|\mathbb{E}_{s \sim S_c}[G^{\text{freq}}(h_\ell(s))] - G_{\ell,c}^{\text{freq}}\big\|_F^2 \tag{12}$$

$$\mathcal{L}_c^{\text{acf}} = \big\|\mathbb{E}_{s \sim S_c}[\text{ACF}(s)] - \text{ACF}_c\big\|_2^2 \tag{13}$$

$$\mathcal{L}_c^{\text{perc}} = \sum_{\ell=1}^{3} \big\|\mathbb{E}_{s \sim S_c}[\text{mean}_{(C,T)}(h_\ell(s))] - \mu_{\ell,c}\big\|_2^2 \tag{14}$$

Gradient Anchor at the Decision Layer. To ensure the synthetic samples are discriminative, we anchor their gradients to those of real samples. Let W be the teacher's final linear layer. For a real mini-batch $\mathcal{B}_c^{\text{real}} \subset \mathcal{D}_c$ and synthetic mini-batch $\mathcal{B}_c^{\text{syn}} \subset S_c$, we define the gradients in Equations 15 and 16.

$$g_c^{\text{real}} = \nabla_W \frac{1}{|\mathcal{B}_c^{\text{real}}|} \sum_{x \in \mathcal{B}_c^{\text{real}}} \ell\big(T_\theta(x), c\big) \tag{15}$$

$$g_c^{\text{syn}} = \nabla_W \frac{1}{|\mathcal{B}_c^{\text{syn}}|} \sum_{s \in \mathcal{B}_c^{\text{syn}}} \ell\big(T_\theta(s), c\big) \tag{16}$$

The gradient anchor penalty minimizes the distance between these two vectors:

$$\mathcal{L}_c^{\text{grad}} = \big\|g_c^{\text{syn}} - g_c^{\text{real}}\big\|_2^2 \tag{17}$$

Waveform Regularizers. We apply a total variation loss to discourage high-frequency noise and a diversity loss to prevent sample collapse:

$$\mathcal{L}^{\text{tv}}(S_c) = \frac{1}{\text{SPC} \cdot CL} \sum_{j=1}^{\text{SPC}} \sum_{k=1}^{C} \sum_{t=1}^{L-1} \big|\, s_{c,j}[k, t+1] - s_{c,j}[k, t] \,\big| \tag{18}$$

$$\mathcal{L}^{\text{div}}(S_c) = \frac{1}{\text{SPC}(\text{SPC} - 1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{\text{SPC}} \frac{\langle \text{vec}(s_{c,i}), \text{vec}(s_{c,j}) \rangle}{\|\text{vec}(s_{c,i})\|_2 \, \|\text{vec}(s_{c,j})\|_2} \tag{19}$$

Total Objective. The final class-wise objective aggregates all terms with non-negative weights λ:

$$\begin{aligned} \mathcal{L}_c = {} & \lambda_{\text{time}}\mathcal{L}_c^{\text{time}} + \lambda_{\text{freq}}\mathcal{L}_c^{\text{freq}} + \lambda_{\text{acf}}\mathcal{L}_c^{\text{acf}} \\ & + \lambda_{\text{perc}}\mathcal{L}_c^{\text{perc}} + \lambda_{\text{grad}}\mathcal{L}_c^{\text{grad}} \\ & + \lambda_{\text{tv}}\mathcal{L}^{\text{tv}}(S_c) + \lambda_{\text{div}}\mathcal{L}^{\text{div}}(S_c) \end{aligned} \tag{20}$$

3.4 Initialization, Optimization, and Training

Synthetic Initialization. We initialize the synthetic samples $\{s_{c,j}\}$ either from K-means cluster centers of the real data or by sampling class exemplars and adding small Gaussian noise, which provides a strong starting point for optimization.

Optimization. We optimize the synthetic data tensor $\mathcal{S}$ directly with the AdamW optimizer, using a short warm-up followed by a cosine decay learning rate schedule. All statistical targets are pre-computed once and frozen, and the teacher network T_θ also remains frozen throughout condensation.

Student Training Protocol. After condensation, a new student network is trained from a random initialization on the final synthetic set $\mathcal{S}$. The student is trained using a standard cross-entropy loss, with the identical recipe used for full-data and all baseline runs to ensure a fair comparison.

4 EXPERIMENTS

4.1 Datasets and metric

We evaluate on five public time–series classification benchmarks spanning human activity, household energy, synthetic bioacoustics, machinery fault diagnosis, and sleep staging (Dau et al., 2019). Table 1 summarizes classes (C), sequence length (T), channels (D), and split sizes for HAR, Electric, Insect, FD, and Sleep, respectively. We report accuracy averaged over multiple random seeds with the corresponding standard deviation, and we follow the official train/test splits released with each dataset. The condensed set size is controlled by the samples-per-class (SPC) budget and the resulting condensed ratio (size(S)/size(T)).

Table 1: Dataset metadata.

Dataset	Domain	C	T	D	Train	Test
HAR	Human Activity	6	128	9	7,352	2,947
Electric	Household Power	7	96	1	8,926	7,711
Insect	Synthetic	10	600	1	25,000	25,000
FD	Fault Diagnosis	3	5,120	1	8,184	2,728
Sleep	Sleep Stages	5	3,000	1	25,612	8,910

4.2 Baselines

We compare TexTSC against strong coreset selection baselines (Random, KMeans, Herding) and state-of-the-art dataset condensation methods adapted from vision to time series (DD (Wang et al., 2020), DSA (Zhao and Bilen, 2021), DM (Zhao and Bilen, 2023), MTT (Cazenavette et al., 2022), IDC (Kim et al., 2022), HaBa (Liu et al., 2022c)), plus CondTSC (Liu et al., 2024b) and the Full trained on all real samples. Baselines are reimplemented from authors' code with their recommended settings.

4.3 Evaluation protocol

Unless otherwise stated, we train the same student classifier across all methods for a fair comparison (a 3-block 1D ResNet), using AdamW with learning rate 10^{-3}, batch size 128, light time–series augmentations on the training batches, and 80 epochs. We perform three runs with different random seeds and report mean±sd. Validation is taken from the official split when provided or via a stratified hold-out from the training split. SPC schedules follow the dataset-specific grids above to keep condensed ratios comparable across methods. This matches the cross-dataset practice in prior work and the table layouts in our draft.

4.4 Overall performance

Table 2 reports accuracy at each SPC/ratio. Across all five datasets and budgets, TexTSC consistently attains the best or second-best accuracy, often closing a significant portion of the gap to the full-data upper bound even at 1% of the data. On HAR and Electric, TexTSC delivers strong gains over coreset baselines at tiny budgets (SPC= 1) and continues to improve as SPC increases. On larger-length datasets (FD, Sleep) and the large-class Insect benchmark, TexTSC maintains robust performance despite the longer temporal context, outperforming prior condensation objectives that match only first-order or single-domain statistics. These trends mirror the cross-budget behavior emphasized in the CondTSC evaluation while improving absolute accuracy in our setting.

4.5 Scaling with condensed size

Figure 2 reports accuracy as the number of samples per class (SPC) increases on HAR and Electric. TexTSC exhibits a clear scaling trend: performance improves steadily with larger condensed sets, closing much of the gap toward the full-data upper bound. At small budgets (SPC= 1), TexTSC already outperforms coreset methods by a wide margin, highlighting the benefits of spectro-temporal second-order matching when data is extremely limited. As the synthetic budget grows (SPC= 5 and SPC= 10), TexTSC continues to improve and consistently dominates competing condensation methods.

Table 2: Overall accuracy(%) performance. The best result is highlighted in bold in each row, and the second-best result is underlined.

Dataset	SPC	ratio (%)	Random	K-means	Herding	DD	DSA	DM	MTT	IDC	HaBa	CondTSC	TexTSC	Full
HAR	1	0.1	50.92 ± 6.53	52.16±3.12	51.48±5.15	21.25±9.15	**69.20±5.68**	16.29±10.49	21.77±14.48	40.72±8.04	64.30±1.95	21.83±6.72	67.73±2.83	92.37±0.29
	5	0.5	57.39±6.11	64.02±2.37	57.78±1.86	27.21±11.72	76.95±4.06	23.45±3.83	27.82±11.48	61.26±5.04	63.68±5.85	59.70±13.56	**78.53±1.00**	
	10	1	59.52±4.18	58.42±10.58	53.79±7.40	14.10±11.17	76.24±3.09	15.69±2.78	14.75±11.71	68.88±4.79	65.23±4.02	65.68±6.24	**83.29±2.13**	
Electric	1	0.1	20.30±6.67	17.26±2.90	17.23±2.71	16.43±4.85	18.77±4.70	26.34±5.77	20.43±4.70	24.93±11.36	25.33±8.69	29.75±1.06	**41.27±2.90**	71.37±1.40
	5	0.5	37.63±6.70	30.86±6.08	28.33±9.95	13.37±3.80	37.26±8.01	13.05±4.81	11.54±0.45	43.20±7.61	18.77±0.57	39.23±2.96	**45.83±4.59**	
	10	1	39.61±10.02	29.35±5.31	28.81±4.36	19.82±7.35	33.57±9.62	21.94±3.88	20.45±5.41	44.23±0.97	24.28±4.59	41.76±1.37	**52.84±3.97**	
FD	1	0.036	52.33±18.12	46.69±18.88	46.84±20.34	27.69±26.27	48.25±10.00	24.47±15.87	41.29±19.07	25.64±19.82	46.96±19.29	53.93±0.79	**73.40±7.70**	98.44±0.93
	10	0.36	50.24±2.11	54.86±0.42	46.09±19.78	31.23±15.14	45.87±9.67	27.77±17.04	41.40±3.19	38.20±21.43	47.49±2.86	55.99±2.04	**78.45±2.91**	
	20	0.72	62.44±2.20	60.56±8.50	45.43±19.02	28.63±18.62	49.72±6.03	38.67±1.71	26.19±18.69	49.25±21.24	52.98±10.93	59.89±5.74	**82.56±9.54**	
Insect	1	0.05	17.13±1.52	8.98±0.32	9.01±0.11	8.89±1.37	18.12±3.21	8.86±1.87	8.97±1.29	12.23±1.71	17.17±2.98	27.95±2.98	**46.92±4.43**	80.29±0.34
	10	0.5	25.22±6.76	36.01±8.66	21.39±2.54	10.01±0.09	25.26±2.81	11.95±2.64	10.10±0.36	19.91±2.76	22.16±1.94	42.29±3.17	**48.35±5.62**	
	20	1	46.40±2.84	44.77±3.85	30.91±3.03	10.34±0.93	31.10±1.30	8.43±1.42	10.11±0.76	24.27±3.93	27.47±1.64	43.84±0.88	**53.82±6.11**	
Sleep	1	0.02	39.08±5.77	22.60±3.08	22.50±0.06	25.54±16.27	**39.06±4.64**	12.46±7.31	33.22±8.43	28.97±11.21	38.20±3.49	20.78±3.92	33.94±10.59	83.40±0.40
	10	0.2	41.67±16.71	18.18±3.29	24.24±3.82	8.85±6.70	32.97±12.12	19.54±7.90	9.36±5.87	13.70±4.03	39.05±4.91	13.34±3.95	**68.41±2.76**	
	20	0.4	38.11±13.12	34.26±11.29	25.06±7.28	19.72±11.29	31.21±4.92	15.30±4.14	26.58±17.07	21.98±6.22	39.61±10.02	11.34±1.47	**61.61±4.16**	

On HAR, TexTSC reaches over 80% test accuracy with only SPC= 5, surpassing all prior objectives and approaching the full-data ceiling of 92.4%. By SPC= 10, the performance climbs further, nearly saturating the upper bound while requiring less than 1% of the original data. On Electric, where coreset methods plateau, TexTSC shows steady improvements with each increment in SPC and achieves the highest condensed-data accuracy across all budgets.

These scaling patterns demonstrate that TexTSC not only secures strong performance in the extreme low-data regime but also remains effective as the synthetic set grows, ensuring a favorable accuracy–cost trade-off. The results mirror earlier observations in CondTSC, where increasing condensed size yields monotonic accuracy gains, but TexTSC improves the absolute level of performance at every budget.

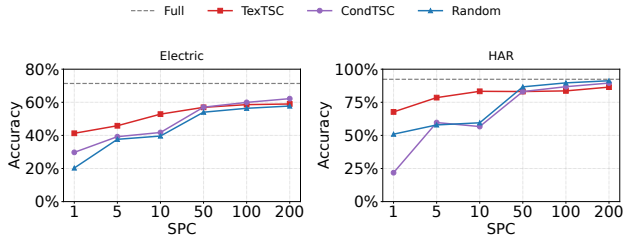

Figure 2: Accuracy of classification on the Electric and HAR datasets with varying condensed data size.

4.6 Learning dynamics on condensed data

Figure 3 plots validation accuracy on HAR at SPC= 1, averaged across random seeds with a shaded ±1 sd band and lightly smoothed (window = 5) to reduce jitter. Accuracy rises quickly during the first 30–40 epochs, peaks around epochs 60–75, and then shows mild late-epoch overfitting. Following standard practice, we therefore report results at the best validation epoch for each run, rather than at convergence. The dynamics are consistent across seeds, indicating that TexTSC produces condensed sets with low variance and stable generalization behavior.

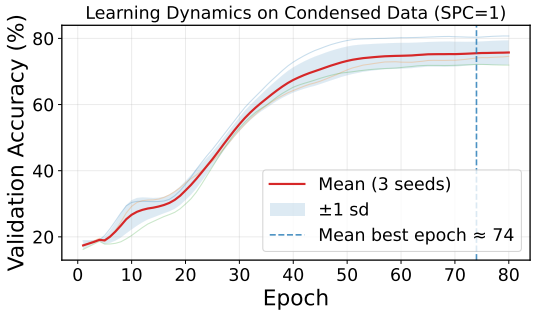

Figure 3: Validation accuracy on HAR at SPC=1.

4.7 Neural Architecture Search as a downstream task

We assess whether condensed sets from TexTSC can accelerate NAS. On the HAR dataset, we perform NAS within our main classifier family (324 variants differing in kernel size, pooling, normalization, and activation). Using only SPC= 10 condensed samples per class, TexTSC identifies architectures that achieve 83.7% test accuracy when retrained on the synthetic set, while the entire search completes in just 14.5 minutes. For comparison, full-data NAS has been reported to require over 11 hours on HAR. This represents a >40× speed-up at a fraction of the compute. Although the rank-correlation between proxy and true rankings

Table 3: Ablation study results (SPC = 1). We report accuracy (%) with mean ± standard deviation.

Dataset	Base	Base+A	Base+A+M	Base+A+M+P
HAR	29.34±4.62	38.16±9.10	58.84±9.10	**67.73±2.83**
Electric	9.89±6.12	1.35±4.41	23.79±6.14	**41.27±2.90**
FD	48.28±3.92	54.51±0.05	55.60±4.35	**73.40±7.70**
Insect	10.16±0.22	15.10±0.86	18.22±2.94	**46.92±2.98**
Sleep	25.13±6.56	31.61±4.31	31.67±4.67	**33.94±10.59**

is modest ($\rho = 0.25$, $\tau_b = 0.17$), the condensed set still provides a useful signal for pruning the search space and consistently yields high-performing architectures.

4.8 Ablation study

Table 3 evaluates TexTSC's components at SPC= 1 across all datasets. In the Base setting, only the last-layer gradient anchor with light TV/diversity is applied, which ensures that synthetic samples stay decision-aligned with real data and avoids collapse. Extending this with Base+A, where time- and frequency-domain Gram matrices are added, injects spectro-temporal texture by matching class-specific co-activation patterns and spectral envelopes, leading to consistent accuracy gains. The next variant, Base+A+M, further incorporates a short-lag autocorrelation term that preserves local periodicity and rhythm, so sequences not only appear texturally correct but also exhibit realistic pulses. Finally, the Full objective, denoted as Base +A+M+P, introduces perceptual means that align coarse feature statistics across taps, stabilizing the texture match, reducing drift, and improving classifier-friendliness. Across all datasets, accuracy improves monotonically from Base through the Full setting, confirming that each component plays a complementary role and that the complete objective consistently delivers the strongest performance.

4.9 Cross-architecture transfer

Table 4 evaluates how well condensed sets generated with one teacher architecture transfer to different student models on HAR. TexTSC achieves reasonable transferability: while accuracy is usually highest when teacher and student match, performance does not collapse when applied across mismatched pairs. This indicates that the spectro-temporal statistics captured by TexTSC are not tied to a single architecture and retain some generality, though the benefit is more modest than in within-architecture settings.

Table 4: Cross-architecture accuracy (%) on HAR with *spc*=5. Rows indicate the network used to produce the condensed set (train network); columns indicate the evaluation network trained on that set.

	Evaluation Network			
Train Network	MLP	CNNIN	ResNet18	Transformer
MLP	**80.29±2.39**	24.34±6.65	50.56±3.96	72.14±2.12
CNNIN	50.98±4.01	**63.88±4.21**	58.31±2.40	57.86±6.35
ResNet18	**50.53±1.99**	22.31±4.33	49.82±7.25	47.72±3.04
Transformer	40.07±8.70	21.37±4.14	**44.36±3.73**	43.07±8.75

5 CONCLUSION

We introduced TexTSC, a dataset condensation framework for time series classification that matches spectro-temporal second-order statistics with a lightweight discriminative anchor. By combining time- and frequency-domain Gram alignment with short-lag autocorrelation and perceptual means, TexTSC generates compact synthetic sets that preserve class-level structure. Experiments across five benchmarks showed that TexTSC consistently outperforms first-order and single-domain baselines, maintains robustness at extremely small budgets, and scales favorably with larger synthetic sets. Additional studies on ablations, cross-architecture transfer, and neural architecture search demonstrate that the condensed data retain useful discriminative and structural cues, though the benefits vary by setting. Overall, TexTSC provides a simple and stable approach for condensing sequential data and offers a step toward practical dataset condensation for real-world time series tasks.

Acknowledgements

This project has been supported in part by funding from the Division of Atmospheric and Geospace Sciences within the Directorate for Geosciences, under NSF awards #2204363, #2240022, #2530946, and #2301397. We would also like to thank Dr. Eamonn Keogh and his team at UCR for their efforts in collecting the data.

References

Cazenavette, G., Wang, T., Torralba, A., Efros, A. A., and Zhu, J.-Y. (2022). Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759.

Cheng, D., Yang, F., Xiang, S., and Liu, J. (2022). Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition*, 121:108218.

Cui, J., Wang, R., Si, S., and Hsieh, C.-J. (2022). Scaling up dataset distillation to imagenet-1k with constant memory. *arXiv preprint arXiv:2211.10586*.

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., and Keogh, E. (2019). The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305.

Deng, Z. and Russakovsky, O. (2022). Remember the past: Distilling datasets into addressable memories for neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pages 34391–34404.

Du, J., Jiang, Y., Tan, V. Y., Zhou, J. T., and Li, H. (2023). Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3758.

Elsken, T., Metzen, J. H., and Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21.

Gupta, V., Narwariya, J., Malhotra, P., Vig, L., and Shroff, G. (2021). Continual learning for multivariate time series tasks with variable input dimensions. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 161–170. IEEE.

Harutyunyan, H., Khachatrian, H., Kale, D. C., Ver Steeg, G., and Galstyan, A. (2019). Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1).

Hayat, N., Geras, K. J., and Shamout, F. E. (2022). Medfuse: Multi-modal fusion with clinical time-series data and chest x-ray images. In *Machine Learning for Healthcare Conference*, pages 479–503. PMLR.

Hosseinzadeh, P., Filali Boubrahimi, S., and Hamdi, S. M. (2024). Improving solar energetic particle event prediction through multivariate time series data augmentation. *The Astrophysical Journal Supplement Series*, 270(2):31.

Hosseinzadeh, P., Nassar, A., Boubrahimi, S. F., and Hamdi, S. M. (2023). Ml-based streamflow prediction in the upper colorado river basin using climate variables time series data. *Hydrology*, 10(2):29.

Jin, W., Tang, X., Jiang, H., Li, Z., Zhang, D., Tang, J., and Yin, B. (2022). Condensing graphs via one-step gradient matching. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 720–730.

Jin, W., Zhao, L., Zhang, S., Liu, Y., Tang, J., and Shah, N. (2021). Graph condensation for graph neural networks. *arXiv preprint arXiv:2110.07580*.

Kim, J.-H., Kim, J., Oh, S. J., Yun, S., Song, H., Jeong, J.-h., Ha, J.-W., and Song, H. O. (2022). Dataset condensation via efficient synthetic-data parameterization. In *International Conference on Machine Learning*, pages 11102–11118. PMLR.

Kumbure, M. M., Lohrmann, C., Luukka, P., and Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 197:116659.

Lee, H. B., Lee, D. B., and Hwang, S. J. (2022a). Dataset condensation with latent space knowledge factorization and sharing. *arXiv preprint arXiv:2208.10494*.

Lee, S., Chun, S., Jung, S., Yun, S., and Yoon, S. (2022b). Dataset condensation with contrastive signals. In *International Conference on Machine Learning*, pages 12352–12364. PMLR.

Li, G., Togo, R., Ogawa, T., and Haseyama, M. (2022). Dataset distillation using parameter pruning. *arXiv preprint arXiv:2209.14609*.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

Liang, C., Huang, Z., Liu, Y., Liu, Z., Zheng, G., Shi, H., Wu, K., Du, Y., Li, F., and Li, Z. (2023). Cblab: Supporting the training of large-scale traffic control policies with scalable traffic simulation. *arXiv preprint arXiv:2210.00896*.

Liu, M., Li, S., Chen, X., and Song, L. (2022a). Graph condensation via receptive field distribution matching. *arXiv preprint arXiv:2206.13697*.

Liu, S., Wang, K., Yang, X., Ye, J., and Wang, X. (2022b). Dataset distillation via factorization. In *Advances in Neural Information Processing Systems*, volume 35, pages 1100–1113.

Liu, S., Wang, K., Yang, X., Ye, J., and Wang, X. (2022c). Dataset distillation via factorization. In *Advances in Neural Information Processing Systems*, volume 35, pages 1100–1113.

Liu, Z., Ding, J., and Zheng, G. (2024a). Frequency enhanced pre-training for cross-city few-shot traffic forecasting. *arXiv preprint arXiv:2406.02614*.

Liu, Z., Hao, K., Zheng, G., and Yu, Y. (2024b). Dataset condensation for time series classification via dual domain matching. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. arXiv:2403.07245.

Liu, Z., Liang, C., Zheng, G., and Wei, H. (2023a). Fdti: Fine-grained deep traffic inference with roadnet-enriched graph. *arXiv preprint arXiv:2306.10945*.

Liu, Z., Zheng, G., and Yu, Y. (2023b). Cross-city few-shot traffic forecasting via traffic pattern bank. *arXiv preprint arXiv:2308.09727*.

Liu, Z., Zheng, G., and Yu, Y. (2024c). Multi-scale traffic pattern bank for cross-city few-shot traffic forecasting. *arXiv preprint arXiv:2402.00397*.

Loo, N., Hasani, R., Lechner, M., and Rus, D. (2023). Dataset distillation with convexified implicit gradients. *arXiv preprint arXiv:2302.06755*.

Nguyen, T., Chen, Z., and Lee, J. (2020). Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*.

Nguyen, T., Novak, R., Xiao, L., and Lee, J. (2021). Dataset distillation with infinitely wide convolutional networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 5186–5198.

Rakhshani, H., Fawaz, H. I., Idoumghar, L., Forestier, G., Lepagnot, J., Weber, J., Brévilliers, M., and Muller, P.-A. (2020). Neural architecture search for time series classification. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Sachdeva, N., Dhaliwal, M. P., Wu, C.-J., and McAuley, J. (2022). Infinite recommendation networks: A data-centric approach. *arXiv preprint arXiv:2206.02626*.

Wang, K., Zhao, B., Peng, X., Zhu, Z., Yang, S., Wang, S., Huang, G., Bilen, H., Wang, X., and You, Y. (2022). Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12196–12205.

Wang, T., Zhu, J.-Y., Torralba, A., and Efros, A. A. (2020). Dataset distillation. *arXiv preprint arXiv:1811.10959*.

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., and Xu, H. (2020). Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*.

Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. (2022). Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. *arXiv preprint arXiv:2202.01575*.

Yang, L. and Hong, S. (2022). Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International Conference on Machine Learning*, pages 25038–25054. PMLR.

Zhang, L., Zhang, J., Lei, B., Mukherjee, S., Pan, X., Zhao, B., Ding, C., Li, Y., and Xu, D. (2023). Accelerating dataset distillation via model augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11950–11959.

Zhang, W., Yang, L., Geng, S., and Hong, S. (2022a). Cross reconstruction transformer for self-supervised time series representation learning. *arXiv preprint arXiv:2205.09928*.

Zhang, X., Zhao, Z., Tsiligkaridis, T., and Zitnik, M. (2022b). Self-supervised contrastive pre-training for time series via time-frequency consistency. In *Advances in Neural Information Processing Systems*, volume 35, pages 3988–4003.

Zhao, B. and Bilen, H. (2021). Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR.

Zhao, B. and Bilen, H. (2023). Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523.

Zhao, B., Mopuri, K. R., and Bilen, H. (2020). Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., and Jin, R. (2022a). Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pages 27268–27286. PMLR.

Zhou, Y., Nezhadarya, E., and Ba, J. (2022b). Dataset distillation using neural feature regression. In *Advances in Neural Information Processing Systems*, volume 35, pages 9813–9827.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes, see Section 3]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [No]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes, See Section 3]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes, see Section 3]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes, see Section 4]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes, see Section 4]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes, see Section 2 and Section 4]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]