Parallel-friendly Spatio-Temporal Graph Learning for Photovoltaic Degradation Analysis at Scale

Yangxin Fan[1,3], Raymond Wieser[1,2], Laura Bruckman[1,2], Roger French[1,2,3], Yinghui Wu[1,3]

[1]SDLE Research Center, Case Western Reserve University

[2]Department of Material Science, Case Western Reserve University

[3]Department of Computer and Data Sciences, Case Western Reserve University

Cleveland, Ohio, USA

{yxf451,rxw497,lsh41,rxf131,yxw1650}@case.edu

ABSTRACT

Photovoltaic (PV) power stations have become an integral component to the global sustainable energy landscape. Accurately monitoring and estimating the performance of PV systems is critical to their feasibility for power generation and as a financial asset. One of the most challenging problems is to understand and estimate the long-term Performance Loss Rate (PLR) for large fleets of PV inverters. This paper introduces a novel **S**patio-**T**emporal **G**raph Neural Network empowered, long-term **Trend** analysis system (**ST-GTrend**), to estimate PLR of PV systems at fleet-level. ST-GTrend nontrivially integrates spatio-temporal coherence and graph attention to separate PLR as a long-term "aging" trend from multiple fluctuation terms in the PV input data, with a design that can easily scale to large PV sets with effective, multi-level parallel computation. (1) To cope with diverse degradation patterns in timeseries, ST-GTrend adopts a paralleled graph autoencoder array to extract aging and fluctuation terms simultaneously, and imposes flatness and smoothness regularizations to disentangle between aging and fluctuation. (2) For large PV systems, ST-GTrend enables a multi-level parallelization paradigm to scale the training and inference computation with a provable performance guarantee. ST-GTrend has been deployed in *CRADLE*, a scientific high performance computing infrastructure. We evaluated ST-GTrend with three real-world large-scale PV datasets, spanning a time period of 10 years. Our results show that ST-GTrend reduces MAPE and Euclidean distance-based errors on average by 34.74% and 33.66% of SOTA methods, and scales well to large PV sets. We also showcase that the advantages of ST-GTrend generalize for the need of long-term trend analysis in financial and economic data. Our source code, datasets, and a full version of the paper are made available [1].

1 INTRODUCTION

Photovoltaic (PV) energy represents a promising solution to growing uncertainty over the stability of the world's energy resources and lowering the carbon footprint [19]. The financial benefits of investing in commodity scale solar must be highlighted to encourage the adoption of PV power generation. This is best accomplished through careful accounting of all of the possible costs associated with the production of energy. The "Levelized Cost of Energy" (LCOE) [7] is a measurement of the total cost of the generation of one unit of energy, and is used to quantify the profitability of a technology. LCOE summarizes the lifetime of the system, including initial cost for planning, building, maintenance, and destruction of a PV site. A critical indicator of LCOE, known as *Performance Loss*, quantifies the long-term ability of a PV system to maintain the generation of the same amount of power under the same weather conditions [26]. Performance loss in PV modules on an individual's roof may impact a single family electricity bill, but the performance loss across fleets PV power plants contribute to millions of dollars in lost revenue. The *Performance Loss Rate* (PLR) of a PV system indicates the decline of the power output over time (in units of % per annum (%/a, or %/year)) [17]. PLR represents the physical degradation of PV modules, which is a critical property of any PV system. An accurate understanding of the PLR of PV systems is critical to inform the LCOE calculation towards more accurate financial predictions, and allow for targeted assets maintenance [8].

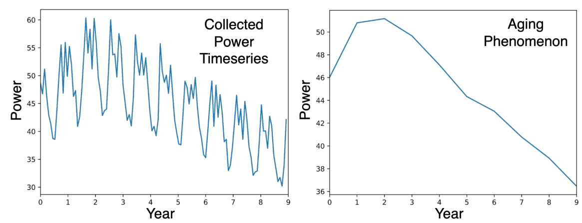


Figure 1: A real PV system with 10-years power timeseries exhibiting non-monotonic degradation pattern. In first two years, the performance actually goes up, and then it degrades as time goes by. Methods that characterize PLR as global linear trend are not accurate for such cases.

Traditional PLR estimation approaches either compute relative PLR [32], by comparing power data output to its counterpart from initial state of the PV system; or absolute PLR [24], which is the coefficient of the slope of the power output as time goes by. They typically model PLR as "monotonic, linearly decreasing quantities", assuming linear degradation of the system components[26]. Nevertheless, we observe that PLR in real PV systems do not always follow a monotonic decreasing or increasing trend. For example, the new installation of PV components may cause a break-in period when the performance of PV systems actually improves [26, 33] (as illustrated in Fig. 1). Indeed, complex module operations and transient effects (*e.g.,* build up of debries on module surfaces) can lead to diversified degradation patterns of power loss and recovery that are both technology and weather dependent[36].

Several data-driven approaches have been investigated to improve PLR estimation beyond traditional approaches. Nevertheless, they remain limited for the need of large PV fleet analysis. (1) State-of-the-art methods such as 6K, PVUSA, XbX, and XbX+UTC [5, 23, 25, 29] can only provide stand-alone, statistical

[1]https://github.com/Yangxin666/ST-GTrend

modeling for a single PV system, yet does not scale to large fleet-level analysis [17]. (2) They often rely on rich domain knowledge and manual model tuning (such as appropriate thresholds, pre-defined filters and physical models) for individual cases. None of these approaches has addressed a streamlined automated pipeline that can "cold-start" over observed PV timeseries data without prior domain knowledge. This hinders large-scale PV PLR analysis.

Spatio-temporal Graph Neural Networks have been applied for non-linear timeseries forecasting, with recent success in PV power predictive analysis [12, 28, 45]. Such models exploit spatio-temporal-topological features to capture PV fleet as network representations, and have been verified to outperform state-of-the-art methods in PV performance analysis. These inspire us to consider new PLR estimation paradigms with the following intuition. (1) Long-term power timeseries should be stationary, despite of the variations due to weather conditions across different seasons and years; and (2) power timeseries can be decomposed into two major components - a long-term "degradation" trend that represents the changes ("up" and "down") in PV performance, and one or multiple fluctuation terms that capture one or multiple levels of seasonalities and their corresponding noise (remainder). We approach PV degradation estimation as an unsupervised regression problem and investigate GNN-based long-term trend analysis to effectively captures long-term degradation patterns for large fleets of PV inverters.

Contributions. We make the following contributions.

(1) We propose **ST-GTrend**, a spatio-temporal graph autoencoder-based method that can effectively capture the degradation patterns in a fleet of PV systems. ST-GTrend adopts paralleled graph autoencoders (one pattern per GAE) to decompose the input into separated trend and a series of different levels of fluctuation patterns and derives degradation from the trend (aging) GAE channel.

(2) We design a unsupervised learning framework that does not need any prior domain-knowledge. We design a novel learning objective to ensure clear disentanglement between aging and fluctuation terms, consisting of three components: (1) Minimizing the reconstruction error between input and sum of aging and multiple fluctuation terms, (2) Ensuring the smoothness of aging term by reducing the noise level, and (3) Imposing the flatness conditions on the fluctuation terms to ensure stationarity of them.

(3) ST-GTrend supports scalable spatiotemporal graph learning and inference for long-term trend analysis. ST-GTrend achieves this through a novel top-down three-level "Think-like" parallelism graph learning algorithm Para-GTrend.

Deployment. ST-GTrend is a deployed system, hosted by CWRU CRADLE High Performance Cluster (HPC) for real-world PV system analysis, using proprietary PV timeseries data received from our PV manufacturer collaborators. ST-GTrend has been used to provide real-time feedback of the performance of real-world PV systems. To our knowledge, ST-GTrend is the first deployed system that uses spatio-temporal graph autoencoders to perform fleet-level degradation analysis, addressing non-linear and non-monotonic long-term trend analysis in a scalable manner.

Related Work. We summarize related work as follows.

Spatiotemporal Graph Neural Networks. Graph Neural Networks [50] has been extensively investigated for graph representation learning. Spatio-Temporal graph neural networks (ST-GNNs) are an extension of GNNs that are designed to model the dynamic node inputs while assuming inter dependency between connected nodes [50]. ST-GNNs capture both spatial and temporal dependencies of graph nodes with *e.g.,* recurrent graph convolutions [2, 42] or attention layers [52, 53]. The former captures spatiotemporal coherence by filtering inputs and hidden states passed to a recurrent unit using graph convolutions, while the latter learns latent dynamic temporal or spatial dependency through attention mechanisms.

While prior study specifies GNN-based models for short-term time-series prediction and analysis (usually with pre-defined historical and short-term predictive windows), not much has been investigated to capitalize ST-GNNs for providing a long-term trend analysis. Existing short-term predictive methods cannot be easily adapted for unsupervised, long-term PLR analysis, especially given non-linearity and non-monotonicity nature of the long-term trends.

Degradation Analysis in Physical Systems. In general, degradation models can be classified into experienced-based approaches, model-based approaches, knowledge-based approaches, and data-driven approaches [20, 43]. Neural Network-based data-driven approaches has become more popular due to their capacity of capturing complex phenomenon without prior knowledge and producing better approximation than traditional regression methods [20]. Previous work [47] proposes degradation profiling method for a single time-series from the complex physical system and solve it as an optimization problem by quadratic programming. Albeit it achieves a clear separation between aging and fluctuation parts on a single timeseries, it fails to capture non-monotonic aging pattern due to the monotonic constraint imposed on aging part. In contrast, ST-GTrend leverages rich spatial correlation from a fleet of physical systems to extract degradation patterns for multiple timeseries, from multiple system components at once. Moreover, architecture design that supports effective scaling out with provable guarantees over large fleet level analysis is not addressed by prior work.

2 PRELIMINARY

PV Network Model. We model a large-fleet PV timeseries dataset $\mathcal{G} = \{G_1, \ldots G_T\}$ as a sequence of undirected graphs ("snapshots"), where for each snapshot $G_t = (V, E, X_t)$ $(t \in [1, T])$, (1) each node in V denotes a PV system or inverter; (2) E refers to the edge set that links (spatially correlated) PV systems; and (3) $X_t \in R^{N \times d}$ denotes the attribute/feature tensor at timestamp t, where (a) N refers to the number of nodes in G_t at time t; and (b) each node carries a d-ary tuple that records d (PV) measurements at timestamp t.

$$A_{i,j} = \begin{cases} 1, i \neq j \text{ and } \exp\left(-\frac{dist_{ij}^2}{\sigma^2}\right) \geq \epsilon \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

The above equation derives the adjacency matrix A of a PV network. By default, ST-GTrend adopts $dist_{ij}$ as Euclidean distance between the locations[2] of PV systems i and j. σ is standard deviation of all pairwise distances. ϵ is a threshold to control the network sparsity: the larger, the sparser the network is.

PLR: A Characterization. PLR degradation pattern is defined as the "performance loss" measured by power output of PV systems

[2] ST-GTrend readily supports other user-defined measures such as spatial-temporal correlation, auto-correlation, or feature similarity as needed.

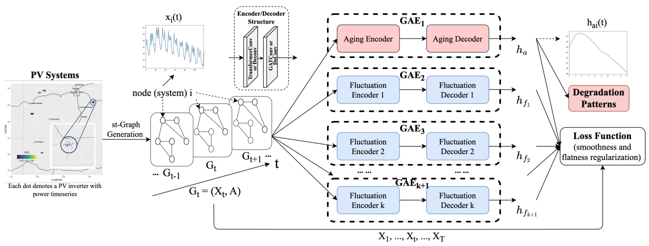


Figure 2: Overview of ST-GTrend Architecture.

under the standard test conditions (STC) over time [27]. STC [34] means fixed test conditions with $25°C$ cell temperature, $1000W/m^2$ plane of module irradiance and AM $1.5G$ spectrum. PV scientists conduct accelerated indoor degradation testing and measure the power output under STC to estimate PLR. Despite the simple computation, indoor testing may not reflect how the PV systems actually degrade in real-world due to the varying environmental factors to which real systems are exposed to. This calls for an automated ML-based approach to estimate PLR of real systems using power output streams. We advocate a modeling of the degradation pattern to be the *aging (trend)* of the timeseries power output over time, and derive the estimated degradation pattern after filtering out the fluctuation terms that capture seasonalities and noises.

Problem Statement. We model PLR estimation as a regression problem. Given the raw PV power timeseries data from a PV fleet, We train a graph neural networks-based model to minimize the error between the real degradation pattern and estimated degradation pattern, by solving an unsupervised regression problem.

More formally, consider a sequence of historically observed PV network graphs $G = \{G_1, \ldots, G_T\}$, where all graphs consist of the same set of nodes V and share the same adjacency matrix A, with varying PV measurements at each node over time, we aim to obtain a St-GNN based model such that it fits the "hidden" PLR pattern by minimizing a loss function that quantifies two types of errors: (a) reconstruction error of the aging and fluctuation terms and (b) additional smoothness and flatness regularization.

3 ST-GTREND FRAMEWORK

We next introduce ST-GTrend architecture and its components.

Model Architecture. We start with the architecture of ST-GTrend. The model ST-GTrend consists of $k+1$ parallelized array of graph autoencoders (GAEs), which *decompose* input PV signals into one *aging term*, and k *fluctuation terms*. (1) The first module (denoted as GAE_1) extracts and learn an *aging* representation h_a, and (2) each module GAE_i ($i \in [2, k+1]$) extracts a distinct fluctuation term.

Justification of design. It is essential for ST-GTrend to learn "disentangled" seasonal-trend representations. (1) Separatedly learned sub-patterns can drastically improve timeseries prediction [15]. (2) Furthermore, instead of leaning a joint representation for multi-seasonal sub-series, ST-GTrend *disentangles* seasonal informational using multiple GAEs: one for a separate fluctuation/seasonal representation. Such design has been shown to be more robust to interventions such as the distribution drift (shifts in distribution of external factors) [49]. (3) Better still, the arrayed design enable a hybrid parallization scheme to scale the training and inference of ST-GTrend model, as verified in Section 5.

We next specify our design of major components in ST-GTrend.

Parallel GAE Array. As illustrated in Fig. 2, Each GAE module consists of an encoder and decoder. Each encoder or decoder consists of a graph transform operator (TransformerConv or TransformerDeConv) followed by a graph attention operator (GATConv or GATDeConv) to form two convolutional or deconvolutional layers. Both operators leverage attention mechanism. In graph transformer operator, the attention mechanism is applied independently to each node and involves self-attention across nodes. On the other hand, graph attention operator computes attention coefficients using a shared self-attention mechanism across all nodes. This design has its convention from established spatio-temporal GNNs *e.g.,* [16]. We adapt such design for basic module in ST-GTrend for aging and fluctuate representations for long-term trend analysis.

Graph Transformer Operators. Graph transformer operators, based on transformer architecture, perform attentive information propagation between nodes by leveraging multi-head attention into graph learning [44]. Multi-headed attention matrices are adopted to replace the original normalized adjacency matrix as transition matrix for message passing. Therefore, nodes can apply self-attention across nodes and update node representations to effectively aggregate features information from their neighbors. We derive output of timeseries at node i x'_i as follows:

$$x'_i = W_1 x_i + \sum_{j \in N(i)} \alpha_{i,j} W_2 x_j \tag{2}$$

Here W_1, W_2, W_3, W_4 are trainable parameters, and D is the hidden size of each attention head. $\alpha_{i,j} = \text{softmax}\left(\frac{(W_3 x_i)^T (W_4 x_j)}{\sqrt{D}}\right)$.

Spatio-Temporal Graph Attention Operators. We adapt the attention mechanism from graph attention networks (GATs) [48]. The operator uses masked self-attention layers and enable different weights to different nodes in a neighborhood by stacking layers, where nodes are able to attend over their neighborhoods' features. This makes ST-GTrend more sensitive to useful information from spatial and temporal "neighbors" in the PV networks. We derive the updated representation of timeseries at node i x'_i as follows:

$$\begin{aligned} x'_i &= \sigma\left(\sum_{j \in N(i)} \alpha_{i,j} W h_j\right); \\ \alpha_{i,j} &= \frac{\exp(\text{LeakyReLU}(a^T [W h_i || W h_j]))}{\sum_{j \in N(i)} \exp(\text{LeakyReLU}(a^T [W h_i || W h_j]))} \end{aligned} \tag{3}$$

Here a is the shared learnable attention parameter vector, W is learnable weight matrix, h_i, h_j are input timeseries for nodes i and j, respectively. σ is an activation function.

Learning Objective. ST-GTrend introduces rich expressiveness with multiple parallelized GAE array. Meanwhile, a clear disentanglement among GAE outputs should be achieved to distinguish aging and fluctuation representations. We next introduce a novel design of the loss function such that it can achieve a separation between aging and fluctuation for long-term trend analysis for PLR estimation. The loss function consists of reconstruction error, flatness regularization, and smoothness regularization.

Reconstruction Error. We aim to decompose X into aging term h_a and k different fluctuation terms $h_{f_1}, h_{f_2}, \ldots, h_{f_k}$ using $k+1$ GAEs. To ensure the quality of decomposition, we want to minimize the

Algorithm 1 : Para-GTrend

Input: A batch of snapshots $\mathcal{G} = \{G_1, \ldots, G_T\}$, (optional) ST-GTrend Model M, number of fluctuation terms k, a set of window sizes W, a set of processors P, a coordinator processor P_0, number of epochs e;
Output: Incrementally trained M' upon batch $\mathcal{G}$.
for $m = 1$ **to** e **do**
  for i in 1 **to** $k+1$ **do in parallel** ▷ Model Parallelism
    if $m = 1$ **then**
      P_0 creates G_i, a copy of G;
    for j in 1 **to** $\lceil \frac{T}{W_i} \rceil$ **do in parallel** ▷ Data Parallelism
      if $m = 1$ **then**
        P_0 ships $G_{i,j} = G_{j+(i-1)\times W_i : j+i\times W_i}$ to $P_{i,j}$;
        $P_{i,j}$ creates $GAE_{i,j}$;
      $GAE_{i,j}$.forward($G_{i,j}$); ▷ Pipeline Parallelism
      Backpropagate and update local gradient of $GAE_{i,j}$;
    if $m = epochs$ **then**
      GAE_i = aggregate($GAE_{i,j}$), $\forall j \in [1, \lceil \frac{T}{W_i} \rceil]$;
P_0 derives M' by assembling all GAE_i, $\forall i \in [1, k+1]$;
return M' from P_0;

Figure 3: Para-GTrend: Three-level Parallel Training.

reconstruction error as follows:

$$\mathcal{L}_{RE} = ||X - h_a - \sum_{q=1}^{k} h_{f_q}||^2 \quad (4)$$

Flatness Regularization. To ensure stationarity of fluctuation, we propose two constraints on h_f: (1) mean constraint: we segment every timeseries to ensure difference of mean between each segment within the series being as small as possible and (2) slope constraint: sum of absolute value of global slope of extracted h_f for each node should be as close to zero as possible.

For mean constraint (MC), we partition h_f of each node into p segments each with length w. The length of w determines the temporal resolution of extract fluctuation term. We then minimize the sum of difference between mean values of each pair of the segments to ensure flatness of the fluctuation over time defined as:

$$\mathcal{L}_{MC} = \sum_{q=1}^{k} \sum_{l=1}^{N} \sum_{i,j=1}^{p} (m_{ql_i} - m_{ql_j})^2 W_{ij} \quad (5)$$

Here (1) m_{ql_i} denotes the mean of the i-th segment of the l-th node's q-th fluctuation term, (2) W is a weight matrix, where each entry W_{ij} denotes a learnable weight to minimize the mean difference between segmented pair m_{l_i} and m_{l_j}. To ensure the long-term mean being minimized, we apply linear growth of the weight based on the distance between m_{l_i} and m_{l_j}. This is based on the following intuition: the farther the two segments are from each other, the more weights are given to minimize their mean difference.

Slope Constraint. The slope constraint (SC) ensures the global flatness of each fluctuation level, and is defined as:

$$\mathcal{L}_{SC} = \sum_{q=1}^{k} \sum_{l=1}^{N} \left| slope(h_{f_{ql}}) \right| \quad (6)$$

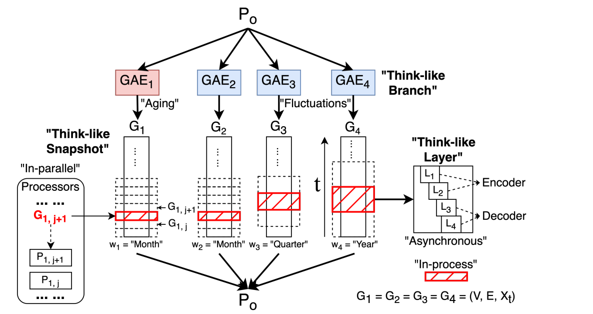


Figure 4: Para-GTrend ($k = 3$ for illustration: one aging channel and three fluctuation channels).

Here the slope is calculated from the least square fit.

Smoothness Regularization. We also want to reduce noises in h_a by minimizing sum of the standard deviation of the first-order differences of aging term of all the nodes. We thus introduce a smoothness regularization (SR) term as:

$$\mathcal{L}_{SR} = \sum_{l=1}^{N} SD(h_{a_l}[t+1] - h_{a_l}[t]) \quad (7)$$

where $t \in [1, T-1]$ with T the total number of timestamps, and SD denotes standard deviation.

Loss Function. Putting these together, we formulate the loss function of ST-GTrend as:

$$\mathcal{L}(X, h_a, h_f) = \mathcal{L}_{RE} + \lambda_1 MC + \lambda_2 \mathcal{L}_{SC} + \lambda_3 \mathcal{L}_{SR} \quad (8)$$

Here $\mathcal{L}_{RE}$, $\mathcal{L}_{MC}$, $\mathcal{L}_{SC}$, and $\mathcal{L}_{SR}$ are defined in the Eq. 3-6. λ_1, λ_2, and λ_3 control the trade-off between reconstruction error and the quality of disentanglement between aging and fluctuations. The minimizes the reconstruction loss while ensuring the flatness of the fluctuation terms and smoothness of the aging term.

Time-cost Analysis. We present an analysis on the inference cost of ST-GTrend. It takes in total $O(Lmf + LPnf^2)$ [18] to derive decomposed results for a timestamp. Here n, m, L, P and f denote number of nodes, number of edges, number of layers of GAE modules, propagation order of message passing (which is bounded by the diameter of PV network), and dimensions of the node features, respectively. As L and P are often small for practical GAE designs, and f are small constants in real PV data, the overall inference cost (including estimating EDP or global PLR) is $O((Lmf + LPnf^2)T + nT)$, which is bounded in $O((m+n)T)$.

4 PARALLELIZATION AND DEPLOYMENT

The design of ST-GTrend is "parallel-friendly" [3]. We next introduce a parallel algorithm, denoted as Para-GTrend, to scale the learning and inference of ST-GTrend to large PV network degradation analysis. Para-GTrend exploits parrallelism with three levels of computation (as illustrated in Fig. 4).

Level I: "Think-like Branch" Model Parallelism. ST-GTrend contains parallelized GAE array with $GAE_1, \ldots, GAE_{k+1}$ branches, as shown in Fig. 2. GAE_1 is the aging branch. $GAE_2, \ldots, GAE_{k+1}$ are the fluctuation branches. This presents opportunities for parallelizing the training by distributing GAE branches among processors. In each training epoch, Para-GTrend computes the forward propagation in parallel without coordination. The output of each branch will be assembled together by an coordinator processor P_0 to calculate global loss in Eqn. 8. Then each branch backpropagates independently and updates their local gradients in parallel.

Level II: "Think-like Snapshot" Data Parallelism. Within each branch, ST-GTrend takes a sequence of "snapshots" $\{G_1, \dots, G_T\}$ as the input. Para-GTrend processes the temporal components of the input with a specified length L in parallel. For each fluctuation branch, L is set to be the window size w for the corresponding fluctuation branch. For example, in Fig. 4, there are three fluctuation branches, each with different window size corresponding to "month", "quarter", and "year". For aging branch, $L = \min_{j \in [2,k+1]} w_j$. Besides, ST-GTrend exploits the mini-batch data parallelism (MBDP) in DGL [54] to achieve even larger speed-up. It splits the information propagation of the network into parallelly computed message flow graphs induced by mutually disjoint node batches.

Level III:"Think-like Layer" Pipeline Parallelism. Each branch of ST-GTrend comprises two layers for encoder, and another two layers for decoder. We adopt an asynchronous macro-pipeline parallelism schema [37] to parallelize the computation of layers within each GAE, such that inter-layer synchronization is eliminated (without information loss, *i.e.,* the batched graphs from level II are independent of each other). A worker processing its graph batch starts the graph operation on the new layer as soon as it completes the non-linearity in previous layer [4].

Performance guarantee. To measure the parallelism of Para-GTrend, we extend parallel scalability [11, 31] to graph learning.

Definition 4.1. A training algorithm is *parallel scalable* if its parallel time cost is inverse proportional to the number of parallel processors, with additional cost that is independent of input size.

Theorem 4.2. *Algorithm Para-GTrend is parallel scalable with a total cost in* $O\left(\frac{T(G,M)}{|P|} + f(\theta)\right)$ *time, where* $T(G,M)$ *is the time cost of training an ST-GTrend model M over G without parallelism,* $|P|$ *is the number of processors, and* $f(\theta)$ *is a polynomial function that is independent of the size of G.*

Proof Sketch. Given the input G (a series of G_t) and model M and the total cost of ST-GTrend training without parallelism $T(G, M)$, since the maximum speed-up is bounded by the number of processors/workers utilized, which is $|P|$, the lower bound of the cost of Para-GTrend is $\frac{T(G,M)}{|P|}$. We denote the communication overhead between the coordinator and processors/workers as $f(\theta)$ which is in $O(k\lceil\frac{T}{W_{min}}\rceil e)$, where $w_{min} = \min\{w_1, w_2, \dots, w_{k+1}\}$. Since w_{min} and k are hyper-parameters, $f(\theta)$ is independent of the the number of nodes and edges. $f(\theta)$ is linear to the length of timeseries $|T|$. Therefore, the total cost of Para-GTrend is in $O\left(\frac{T(G,M)}{|P|} + f(\theta)\right)$ such that $f(\theta)$ is independent of size of the G_t and linear to the length of timeseries. We present the detailed cost analysis in [**?**].

System Deployment. ST-GTrend has been deployed on CWRU HPC CRADLE [38] connected with real-world company datasets. Our collaborators (*e.g.,* SolarEdge, CSI Solar, etc) has tested ST-GTrend on their proprietary datasets which provide feedback on the health of their PV systems. Fig. 5 illustrates how ST-GTrend (with Model Parallelism illustrated) is incorporated into CRADLE

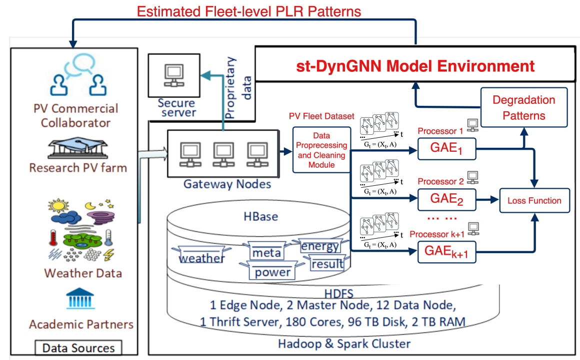


Figure 5: Deployment Pipeline of ST-GTrend in CRADLE [38] for PV Systems from Our Industry Partners (with Level I Parallelism illustrated).

system. Acquisition of high-quality complete data is challenging for PLR degradation analysis on real-world PV systems. We implement our proposed STD-GAE [13] to impute the missing and incomplete timeseries data and FAIRification pipeline [14]to ensure data quality. We have verified that ST-GTrend is robust over the case of random missing up to 60% missing rate. Thanks to the parallel-friendly structure of ST-GTrend, Para-GTrend accelerates training and inference of ST-GTrend by leveraging multiple CPUs/GPUs in parallel from CRADLE. The estimated fleet-level PLR results are forwarded to our PV commercial collaborators and research PV farms.

5 EXPERIMENTS

We conduct four sets of experiments to verify the performance of ST-GTrend and compared it with the baselines.

Datasets. We verify our model on diverse real-world datasets, including three PV datasets each exhibiting different degradation patterns and two datasets from Economy and Finance. Each PV dataset consists of 10-year power output timeseris from 100 inverters in 5 PV sites from Colorado, USA. Every power output timeseries includes sample interval of 15 minutes, amounting to 350,592 data points for each PV inverter. (1) **PV_Case1**: It refers to the PV datasets exhibiting global linear degradation patterns. (2) **PV_Case2**: It refers to the PV datasets showing a piecewise linear degradation pattern with a breakpoint (a change in the degradation rate at the second year). (3) **PV_Case3**: This refers to the PV dataset exhibiting a non-linear exponential degradation. To demonstrate the generalization of ST-GTrend for trend analysis in other domains, we also adopt the following extra datasets. (4) **Finance**[3]: This refers to weekly stock prices of the 500 large US corporations listed in S&P 500 from 12/13/2018 to 12/13/2023. (5) **Economy**[4]: the annual GDP of G20 countries in past 50 years from 1973 to 2022. For finance (resp. economy) data, each node represents a stock (resp. country). For both datasets, we used thresholded pairwise absolute pearson correlation among node attributes to establish edges.

Evaluation. We report the degradation estimation errors using *Euclidean Distance (ED)* and *Mean Absolute Percent Error* (MAPE); the smaller, the better PLR estimations are, as smaller ED and MAPE values indicate smaller gap between the estimated degradation

[3]https://finance.yahoo.com/

[4]https://data.worldbank.org/

Table 1: Performance Comparison between ST-GTrend and Baselines (best-performing are in bold, the second-best are italicized).

Models	PV_Case1		PV_Case2		PV_Case3		Finance		Economy	
	MAPE	ED	MAPE	ED	MAPE	ED	MAPE	ED	MAPE	ED
6K	11.97 ± 0.71	0.410 ± 0.024	11.11 ± 0.70	0.424 ± 0.025	12.14 ± 0.67	0.410 ± 0.024	-	-	-	-
PVUSA	4.66 ± 0.26	0.167 ± 0.010	4.59 ± 0.26	0.178 ± 0.010	4.71 ± 0.25	0.166 ± 0.010	-	-	-	-
XbX	4.21 ± 0.11	0.147 ± 0.004	4.20 ± 0.10	0.161 ± 0.004	4.18 ± 0.07	0.144 ± 0.003	-	-	-	-
XbX+UTC	1.27 ± 0.04	0.052 ± 0.002	1.22 ± 0.04	0.045 ± 0.002	*1.21 ± 0.04*	*0.044 ± 0.002*	-	-	-	-
QP-Aging-Detect	2.38 ± 0.21	0.071 ± 0.005	1.18 ± 0.07	0.043 ± 0.002	2.57 ± 0.24	0.073 ± 0.005	28.16 ± 1.52	2.12 ± 0.12	13.17 ± 0.97	0.45 ± 0.02
MSTL	1.84 ± 0.09	0.055 ± 0.003	1.90 ± 0.10	0.064 ± 0.003	1.45 ± 0.09	0.041 ± 0.003	24.47 ± 1.66	1.94 ± 0.13	10.70 ± 0.73	0.37 ± 0.02
SSA	2.03 ± 0.19	0.059 ± 0.004	1.13 ± 0.08	0.041 ± 0.003	1.96 ± 0.17	0.054 ± 0.004	21.72 ± 1.46	1.70 ± 0.11	17.37 ± 1.07	0.57 ± 0.04
STGAE1	0.61 ± 0.03	0.022 ± 0.002	2.00 ± 0.18	0.071 ± 0.006	2.23 ± 0.19	0.069 ± 0.005	22.12 ± 1.33	1.75 ± 0.11	16.67 ± 1.02	0.58 ± 0.03
STGAE2	*0.46 ± 0.02*	*0.018 ± 0.001*	2.04 ± 0.15	0.075 ± 0.005	1.81 ± 0.15	0.053 ± 0.004	21.85 ± 1.31	1.72 ± 0.11	10.53 ± 0.69	0.36 ± 0.01
ST-GTrend-NF	2.84 ± 0.19	0.094 ± 0.005	1.46 ± 0.11	0.052 ± 0.003	2.33 ± 0.14	0.076 ± 0.005	26.69 ± 1.48	2.31 ± 0.13	14.89 ± 0.86	0.51 ± 0.01
ST-GTrend-NS	0.69 ± 0.03	0.027 ± 0.001	*1.01 ± 0.04*	*0.038 ± 0.001*	1.95 ± 0.05	0.068 ± 0.001	*21.26 ± 1.27*	*1.68 ± 0.10*	*9.49 ± 0.65*	*0.35 ± 0.01*
ST-GTrend	**0.40 ± 0.02**	**0.015 ± 0.001**	**0.96 ± 0.04**	**0.036 ± 0.001**	**0.85 ± 0.03**	**0.031 ± 0.001**	**18.65 ± 1.15**	**1.44 ± 0.08**	**9.31 ± 0.61**	**0.31 ± 0.01**

pattern (EDP) and its "real" counterpart (RDP). (1) ED quantifies how well a model can estimate PLR pattern with two steps: (1) Rescaling: Given RDP and EDP, we rescale every data point in RDP and EDP by dividing them by their respective first values and (2) Error Calculation: We calculate ED between scaled RDP and EDP. (2) MAPE is defined as $MAPE = \sum_{L=1}^{N}\sum_{J=1}^{T}\frac{|EDP_{L_J}-RDP_{L_J}|}{N\times T\times|RDP_{L_J}|}\times 100\%$, where EDP_{L_J} is the $J-th$ coefficient of EDP for node L. Here the EDP of a node i is defined as $EDP=\frac{h_{a_i}(t)}{h_{a_i}(1)}$ where $h_{a_i}(1)$ is the first value in the aging sequence $h_{a_i}(t)$.

RDP simulation. As the real RDP values for PLR and trends of other datasets are unavailable, we exploit well-established physics-informed simulation and algorithmic tools to simulate "yardstick" ground truth RDP values. We remark that these approaches alone require careful manual tuning and domain knowledge. (1) We generate RDP for PV dataset using NSRDB Physical Solar Model v3 based on TMY Cyclic Weather [51] and Sandia Array Performance Model from pvlib [21]. (2) We obtain the RDP of stock and GDP timeseries using empirical mode decomposition [22] which is widely applied in financial and economic trend analysis [6, 9, 35, 41]. We split data by nodes, 50%, 25%, and 25% for training, validation, and testing.

Baselines. We compare ST-GTrend with the following eight baselines. (1) *6K* [23]: a domain-specific model, which incorporates irradiance and module temperature as a fraction of standard irradiance and difference from standard temperature; (2) *PVUSA* [29]: an established physics-based PLR estimation model. The assumption of the model is that the current of a solar panel is a function of the irradiance and the voltage is a function of the irradiance and the module temperature; (3) *XbX* [5]: a data-driven, multiple regression predictive model. The model enables change of point PV degradation pattern modeling; (4) *XbX + UTC* [25]: The XbX + UTC is based on the XbX model by introducing a universal temperature correction (UTC); (5) *MSTL* [46]: Multiple Seasonal-Trend Decomposition using Loess (MSTL) decomposes a time series into a: trend component, multiple seasonal components, and a residual component. MSTL uses STL to iteratively extract seasonal components from a time series; (6) *SSA* [1]: Singular Spectrum Analysis (SSA) allows extraction of alleged trend, seasonal and noise components from time series. The name comes from singular decomposition of a matrix into its spectrum of eigenvalues. The time series can be reconstructed by regrouping different important components; (7) *QP-Aging-Detect* [47]: The QP-Aging-Detect, is a mathematical model using Quadratic Programming to profile long-term degradation by decomposing timeseries into the aging and fluctuation terms. For fair comparison, we remove the monotonic constraint imposed on aging component when implementing it in CVXPY [10]; (8) *STGAEs*: Spatio-Temporal Graph Autoencoders (STGAEs) are variants of ST-GTrend, where the encoder and decoder layers in ST-GTrend are replaced by Graph Convolutional (GCN) Layer [30] (STGAE1) or graph attention layers [48] (STGAE2).

Training configuration. We use grid search based on the validation loss to identify the optimal set of hyper-parameters [39] for ST-GTrend. We varied the number of fluctuation term k in the set $\{1, 2, 3, 4\}$, network sparsity ϵ in the set $\{0, 0.25, 0.5, 0.75, 1\}$, number of epochs in the set $\{250, 500, 750, 1000\}$, learning rate from $\{0.001, 0.01, 0.05, 0.1\}$, regularization terms λ_1, λ_2, and λ_3 all in the set $\{1, 5, 10, 25, 50, 75, 100, 200\}$. Based on the validation loss, we choose k equal to 1, λ_1, λ_2, and λ_3 to be 5, 100, and 10 separately, epsilon to be 0.5, epochs to be 500, the learning rate to be 0.05.

ST-GTrend is trained by Adam optimizer [30] and implemented with PyTorch Geometric Temporal [40], on 12 Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 128 GB Memeory, 16 cores, and 8 32GB NVIDIA V100 GPU from CWRU CRADLE Cluster.

We next present our findings.

Exp-1: Accuracy . We evaluate the accuracy of PLR estimation of ST-GTrend, over diverse degradation patterns of PV systems. Table 1 reports PLR degradation pattern estimation errors of all the methods acorss three classes of PV datasets, each exhibiting a different type of degradation pattern: linear, linear with breakpoint, and non-linear; as well as two other datasets from finance and economy. We observe the followings: (1) ST-GTrend achieves the best PLR degradation pattern estimation measured by MAPE and ED across all datasets. Compared to the top two baselines XbX+UTC and STGAE2, ST-GTrend achieves a reduction of 34.74% and 33.66% on average in MAPE and ED. (2) ST-GTrend continues to outperform the other baselines significantly in more challenging degradation patterns Case2 and Case3. These results demonstrate that ST-GTrend models can provide accurate estimation of PLR patterns for all systems within a large PV fleet.

Exp-2: Ablation Analysis. To study how regularization terms imposed on learning objective affect the accuracy of ST-GTrend, we conduct two ablation studies (1) removing flatness regularization (2) removing smoothness regularization from our learning objective in Eq. 8 to analyze how they affect the accuracy of ST-GTrend.

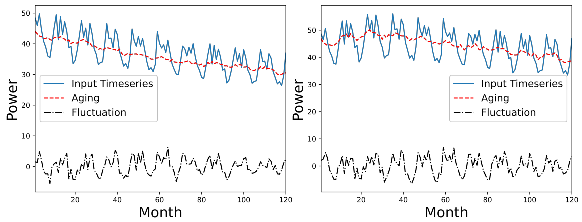


Figure 6: Example of Extracted Aging (in red) and Fluctuation Terms (in black) by ST-GTrend for a PV System (left: PV_Case 3, right: PV_Case 2).

As illustrated in Table 1, we observe that ST-GTrend with the flatness regularization reduces MAPE and ED by 58.10% and 54.92% on average compared with its counterpart without flatness regularization (ST-GTrend-NF). Similarly for smoothness regularization, we observe that ST-GTrend with smoothness regularization reduces MAPE and ED by 30.87% and 31.27% on average compared to its counterpart without smoothness regularization (ST-GTrend-NS). These results verify the effectiveness and necessity of regularization and smoothness terms in the design of ST-GTrend.

Exp-3: Effectiveness of Deployed ST-GTrend. Using real-world PV and domain datasets, we evaluate the effectiveness of the design of ST-GTrend and report the performance of the deployed ST-GTrend with the following cases.

Decomposition Results from ST-GTrend. Fig. 6 illustrates extracted aging and fluctuation terms of a PV system under linear with breakpoint and non-linear degradation patterns. From this figure, we observe that (1) ST-GTrend achieves a clear separation between aging and fluctuation terms; (2) Aging term successfully captures the initial upward trend in the cases 2 degradation pattern, and (3) Fluctuation term captures annual seasonality and noises.

Effectiveness of Encoder and Decoder Design. To further analyze the effectiveness of the encoder and decoder design of ST-GTrend, we quantify the estimation errors changes when replacing the graph convolution layers in ST-GTrend by either graph convolutional (GCN) layers in STGAE1 or graph attention networks (GAT) layers in STGAE2. We observe that STGAE1 and STGAE2 incurs larger PLR estimation errors, on average 89.8% and 65.7% larger MAPE, when compared to ST-GTrend. Our results verify the effectiveness of design of the layers of autoencoder in ST-GTrend.

Visual Analysis. We compare EDP extracted by ST-GTrend and top six best-performed baselines with RDP. Fig. 7 shows ST-GTrend can better recover real degradation pattern from input timeseries than all baselines since it show a closer match with RDP. We can see EDP extracted by ST-GTrend is the closest to RDP in both case 2 and case 3 figures followed by XbX+UTC and STGAE2, which is consistent with the comparison of degradation pattern estimation error results shown in Table. 1.

Exp-4: Scalability. We next evaluate the scalability of ST-GTrend by reporting the training cost of Para-GTrend. (1) We report the scalability of Level I (Model) Parallelism by distributing the training of GAE branches into parallel workers (CPUs). As shown in Fig. 8 (blue line), the training time over all three datasets are almost inverse proportional to the number of workers as the latter increases from 1 to 8. (2) We next report the result for the Level II (Data

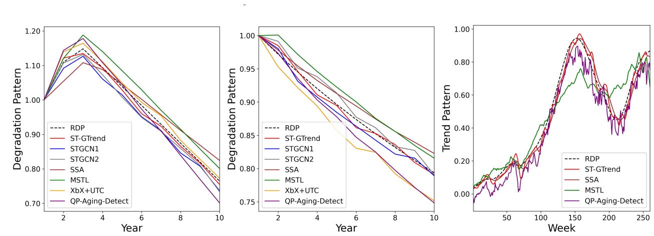
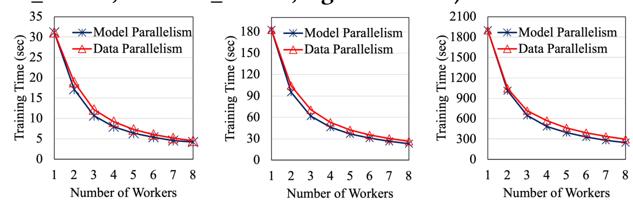


Figure 7: Comparison of Real Degradation/Trend Pattern with Patterns Extracted by ST-GTrend and Baselines (left: PV_Case 2; mid: PV_Case 3; right: Finance).



(a) Economy **(b) Finance** **(c) PV**

Figure 8: Scalability Test on Model and Data Parallelism (training time clocked by averaging 50 rounds).

Parallelism) where we use multiple workers (GPUs) to train local GAE by fitting the batched input spatio-temporal graph snapshots in parallel. Fig. 8 verifies that the parallel training of ST-GTrend scales well with more workers. For example, Para-GTrend improves the training of ST-GTrend by 7.56 times (on average over all the three datasets) as the number of workers increases from 1 to 8.

6 CONCLUSION

We have introduced ST-GTrend, a deployed spatiotemporal GNN-based system that adopts parallelized graph autoencoder array architecture to decompose input PV power timeseries into aging (performance loss) and fluctuation terms for long-term PLR estimation. Each module exploits spatial coherence from neighboring PV inverters and temporal correlations within-series to perform aging and fluctuation extractions for PV fleet-level analysis. The loss function of ST-GTrend ensures the clear disentanglement between aging and fluctuation through smoothness and flatness regularizations. We have shown that the "parallel-friendly" architecture of ST-GTrend enables multi-level parallel training algorithm, combining both model and data parallelism for large, fleet-level PV analysis. Our experiments have verified that ST-GTrend outperforms SOTA methods in both accuracy and learning efficiency, scale well for large-scale analysis, and readily generalizes to other applications.

Acknowledgments. This work is supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under Solar Energy Technologies Office (SETO) Agreement Number DE-EE0009353. The authors thank our collaborators from SolarEdge and CSI Solar for providing real-world PV datasets.

REFERENCES

[1] Halit Apaydin, Mohammad Taghi Sattari, Kambiz Falsafian, and Ramendra Prasad. 2021. Artificial intelligence modelling integrated with Singular Spectral analysis and Seasonal-Trend decomposition using Loess approaches for streamflow predictions. *Journal of Hydrology* 600 (2021), 126506.

[2] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in Neural*

Information Processing Systems 33 (2020), 17804–17815.
[3] Tal Ben-Nun and Torsten Hoefler. 2019. Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. *ACM Computing Surveys (CSUR)* 52, 4 (2019), 1–43.
[4] Maciej Besta and Torsten Hoefler. 2022. Parallel and distributed graph neural networks: An in-depth concurrency analysis. *arXiv preprint arXiv:2205.09702* (2022).
[5] Alan J Curran, Yang Hu, Rojiar Haddadian, Jennifer L Braid, David Meakin, Timothy J Peshek, and Roger H French. 2017. Determining the power rate of change of 353 plant inverters time-series data across multiple climate zones, using a month-by-month data science analysis. In *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*. IEEE, 1927–1932.
[6] Zhifeng Dai and Huan Zhu. 2020. Forecasting stock market returns by combining sum-of-the-parts and ensemble empirical mode decomposition. *Applied Economics* 52, 21 (2020), 2309–2323.
[7] Seth B Darling, Fengqi You, Thomas Veselka, and Alfonso Velosa. 2011. Assumptions and the levelized cost of energy for photovoltaics. *Energy & environmental science* 4, 9 (2011), 3133–3139.
[8] Chris Deline, Mike Deceglie, Dirk Jordan, Matt Muller, Kevin Anderson, Kirsten Perry, Robert White, and Mark Bolinger. 2022. *Reducing Uncertainty of Fielded Photovoltaic Performance (Final Technical Report)*. Technical Report NREL/TP-5K00-82816. National Renewable Energy Lab. (NREL), Golden, CO (United States). https://doi.org/10.2172/1880076
[9] Changrui Deng, Yanmei Huang, Najmul Hasan, and Yukun Bao. 2022. Multi-step-ahead stock price index forecasting using long short-term memory model with multivariate empirical mode decomposition. *Information Sciences* 607 (2022), 297–321.
[10] Steven Diamond and Stephen Boyd. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research* 17, 83 (2016), 1–5.
[11] Wenfei Fan, Xin Wang, Yinghui Wu, and Dong Deng. 2014. Distributed graph simulation: Impossibility and possibility. *Proceedings of the VLDB Endowment* 7, 12 (2014), 1083–1094.
[12] Yangxin Fan, Xuanji Yu, Raymond Wieser, David Meakin, Avishai Shaton, Jean-Nicolas Jaubert, Robert Flottemesch, Michael Howell, Jennifer Braid, Laura Bruckman, Roger French, and Yinghui Wu. 2023. Spatio-Temporal Denoising Graph Autoencoders with Data Augmentation for Photovoltaic Data Imputation. *Proc. ACM Manag. Data* 1, 1, Article 50 (may 2023), 19 pages. https://doi.org/10.1145/3588730
[13] Yangxin Fan, Xuanji Yu, Raymond Wieser, David Meakin, Avishai Shaton, Jean-Nicolas Jaubert, Robert Flottemesch, Michael Howell, Jennifer Braid, Laura S Bruckman, and Yinghui Wu. 2023. Spatio-Temporal Denoising Graph Autoencoders with Data Augmentation for Photovoltaic Timeseries Data Imputation. In *Proceedings of the 2023 International Conference on Management of Data*.
[14] Yangxin Fan, Xuanji Yu, Raymond Wieser, Yinghui Wu, and Roger French. 2024. *PVplr-stGNN*. SDLE Research Center, Case Western Reserve Universitu. https://pypi.org/project/PVplr-stGNN/ PyPI.
[15] Jie Feng, Yong Li, Ziqian Lin, Can Rong, Funing Sun, Diansheng Guo, and Depeng Jin. 2021. Context-aware spatial-temporal neural network for citywide crowd flow prediction via modeling long-range spatial dependency. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16, 3 (2021), 1–21.
[16] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.
[17] Roger H French, Laura S Bruckman, David Moser, EURAC Lindig, Mike van Iseghem, Joshua S Stein, Maurice Richter, Magnus Herz, Wilfried van Sark, Franz Baumgartner, et al. 2021. Assessment of performance loss rate of PV power systems. (2021).
[18] Xinyi Gao, Wentao Zhang, Yingxia Shao, Quoc Viet Hung Nguyen, Bin Cui, and Hongzhi Yin. 2022. Efficient Graph Neural Network Inference at Large Scale. *arXiv preprint arXiv:2211.00495* (2022).
[19] Nouha Gazbour, Guillaume Razongles, Elise Monnier, Maryline Joanny, Carole Charbuillet, Françoise Burgun, and Christian Schaeffer. 2018. A path to reduce variability of the environmental footprint results of photovoltaic systems. *Journal of cleaner production* 197 (2018), 1607–1618.
[20] Nima Gorjian, Lin Ma, Murthy Mittinty, Prasad Yarlagadda, and Yong Sun. 2010. A review on degradation models in reliability analysis. In *Engineering Asset Lifecycle Management: Proceedings of the 4th World Congress on Engineering Asset Management (WCEAM 2009), 28-30 September 2009*. Springer, 369–384.
[21] William F Holmgren, Clifford W Hansen, and Mark A Mikofski. 2018. pvlib python: A python package for modeling solar energy systems. *Journal of Open Source Software* (2018).
[22] Norden E Huang, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H Liu. 1998. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences* 454, 1971 (1998), 903–995.
[23] Thomas Huld, Gabi Friesen, Artur Skoczek, Robert P Kenny, Tony Sample, Michael Field, and Ewan D Dunlop. 2011. A power-rating model for crystalline silicon PV modules. *Solar Energy Materials and Solar Cells* 95, 12 (2011), 3359–3369.
[24] Philip Ingenhoven, Giorgio Belluardo, and David Moser. 2017. Comparison of statistical and deterministic smoothing methods to reduce the uncertainty of performance loss rate estimates. *IEEE Journal of Photovoltaics* 8, 1 (2017), 224–232.
[25] Dirk C Jordan, Chris Deline, Sarah R Kurtz, Gregory M Kimball, and Mike Anderson. 2017. Robust PV degradation methodology and application. *IEEE Journal of photovoltaics* 8, 2 (2017), 525–531.
[26] Dirk C. Jordan, Timothy J. Silverman, Bill Sekulic, and Sarah R. Kurtz. 2017. PV Degradation Curves: Non-Linearities and Failure Modes. *Progress in Photovoltaics: Research and Applications* 25, 7 (July 2017), 583–591. https://doi.org/10.1002/pip.2835
[27] Ismail Kaaya, Sascha Lindig, Karl-Anders Weiss, Alessandro Virtuani, Mariano Sidrach de Cardona Ortin, and David Moser. 2020. Photovoltaic lifetime forecast model based on degradation patterns. *Progress in Photovoltaics: Research and Applications* 28, 10 (2020), 979–992.
[28] Ahmad Maroof Karimi, Yinghui Wu, Mehmet Koyuturk, and Roger H French. 2021. Spatiotemporal Graph Neural Network for Performance Prediction of Photovoltaic Power Systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
[29] David L King, Jay A Kratochvil, and William E Boyson. 1997. Field experience with a new performance characterization procedure for photovoltaic arrays. *Sandia National Lab.(SNL-NM), Albuquerque, NM (United States)* (1997).
[30] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.
[31] Vipin Kumar and Anshul Gupta. 1991. Analysis of scalability of parallel algorithms and architectures: A survey. In *Proceedings of the 5th international conference on Supercomputing*. 396–405.
[32] Sascha Lindig, Ismail Kaaya, Karl-Anders Weiß, David Moser, and Marko Topic. 2018. Review of statistical and analytical degradation models for photovoltaic modules and systems as well as related improvements. *IEEE Journal of Photovoltaics* 8, 6 (2018), 1773–1786.
[33] Sascha Lindig, Marios Theristis, and David Moser. 2022. Best practices for photovoltaic performance loss rate calculations. *Progress in Energy* 4, 2 (2022), 022003.
[34] George Makrides, Bastian Zinsser, Markus Schubert, and George E Georghiou. 2014. Performance loss rate of twelve photovoltaic technologies under field conditions using statistical techniques. *Solar Energy* 103 (2014), 28–42.
[35] Xuegeng Mao, Albert C Yang, Chung-Kang Peng, and Pengjian Shang. 2020. Analysis of economic growth fluctuations based on EEMD and causal decomposition. *Physica A: Statistical Mechanics and its Applications* 553 (2020), 124661.
[36] Bennet Meyers. 2022. Estimation of Soiling Losses in Unlabeled PV Data. In *2022 IEEE 49th Photovoltaics Specialists Conference (PVSC)*. 0930–0936. https://doi.org/10.1109/PVSC48317.2022.9938567
[37] Deepak Narayanan, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R Devanur, Gregory R Ganger, Phillip B Gibbons, and Matei Zaharia. 2019. PipeDream: Generalized pipeline parallelism for DNN training. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles*. 1–15.
[38] Arafath Nihar, Alan J Curran, Ahmad M Karimi, Jennifer L Braid, Laura S Bruckman, Mehmet Koyutürk, Yinghui Wu, and Roger H French. 2021. Toward findable, accessible, interoperable and reusable (fair) photovoltaic system time series data. In *2021 IEEE 48th Photovoltaic Specialists Conference (PVSC)*. IEEE, 1701–1706.
[39] Fabrício José Pontes, GF Amorim, Pedro Paulo Balestrassi, AP Paiva, and João Roberto Ferreira. 2016. Design of experiments and focused grid search for neural network parameter optimization. *Neurocomputing* 186 (2016), 22–34.
[40] Benedek Rozemberczki, Paul Scherer, Yixuan He, George Panagopoulos, Alexander Riedel, Maria Astefanoaei, Oliver Kiss, Ferenc Beres, Guzman Lopez, Nicolas Collignon, and Rik Sarkar. 2021. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*.
[41] Foued Saâdaoui and Othman Ben Messaoud. 2020. Multiscaled neural autoregressive distributed lag: A new empirical mode decomposition model for nonlinear time series forecasting. *International Journal of Neural Systems* 30, 08 (2020), 2050039.
[42] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*. Springer, 362–373.
[43] Ameneh Forouzandeh Shahraki, Om Parkash Yadav, and Haitao Liao. 2017. A review on degradation modelling and its engineering applications. *International Journal of Performability Engineering* 13, 3 (2017), 299.
[44] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. 2020. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509* (2020).
[45] Jelena Simeunovic, Baptiste Schubnel, Pierre-Jean Alet, and Rafael E. Carrillo. 2021. Spatio-temporal graph neural networks for multi-site PV power forecasting. *CoRR* abs/2107.13875 (2021). arXiv:2107.13875 https://arxiv.org/abs/2107.13875

[46] Oscar Trull, J Carlos García-Díaz, and Angel Peiró-Signes. 2022. Multiple seasonal STL decomposition with discrete-interval moving seasonalities. *Appl. Math. Comput.* 433 (2022), 127398.

[47] Liudmila Ulanova, Tan Yan, Haifeng Chen, Guofei Jiang, Eamonn Keogh, and Kai Zhang. 2015. Efficient long-term degradation profiling in time series for complex physical systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2167–2176.

[48] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *stat* 1050, 20 (2017), 10–48550.

[49] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. 2022. CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting. In *International Conference on Learning Representations*.

[50] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[51] Dazhi Yang. 2018. A correct validation of the national solar radiation data base (NSRDB). *Renewable and Sustainable Energy Reviews* 97 (2018), 152–155.

[52] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3634–3640.

[53] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1234–1241.

[54] Da Zheng, Chao Ma, Minjie Wang, Jinjing Zhou, Qidong Su, Xiang Song, Quan Gan, Zheng Zhang, and George Karypis. 2020. Distdgl: distributed graph neural network training for billion-scale graphs. In *2020 IEEE/ACM 10th Workshop on Irregular Applications: Architectures and Algorithms (IA3)*. IEEE, 36–44.