Supplementary Material for Adaptive Spatial-Temporal Graph Learning-Enabled Short-Term Voltage Stability Assessment against Time-Varying Topological Conditions

Chao Deng, *Student Menber, IEEE*, Lipeng Zhu, *Senior Menber, IEEE*, Chang Liu, Hefeng Zhai, Baoye Tian, Zexiang Zhu, Jiayong Li, *Menber, IEEE*, and Cong Zhang, *Menber, IEEE*

***Abstract*—The emerging deep learning (DL) technology has recently exhibited great potential in data-driven short-term voltage stability (SVS) assessment of complex power grids. However, without sufficient attention to the time-varying topological structures of today's power grids, the majority of existing DL-based SVS assessment schemes could experience severe performance degradation in practice. To address this drawback, this paper proposes an adaptive spatial-temporal graph learning-enabled SVS assessment approach that can adapt well to various topological structural changes. First, considering the time-varying topological conditions of a given power grid, an adaptive graph representation matrix is automatically learned to effectively capture the complicated spatial correlations between individual buses within the grid. Then, to help better capture regional SVS features for subsequent learning processes, the adaptive graph representation matrix is properly adjusted by introducing a spatial attention mechanism. Further, with post-fault system trajectory data linked together via attention-based graph representation, a residual spatiotemporal graph convolutional network is carefully built with Optuna-based optimization to deeply mine system-wide spatiotemporal features and thus achieve structure-adaptive SVS assessment. Numerical test results on two representative sub-systems of a realistic provincial power grid in South China demonstrate the efficacy of the proposed approach in various topological structural conditions.**

***Index Terms*—Adaptive spatial-temporal graph learning, graph representation matrix, short-term voltage stability, time-varying topological conditions, attention mechanism.**

APPENDIX

A. Hyperparameters Setting

All approaches utilize Bayesian optimization based on the tree-structured parzen estimator (TPE) to perform 50 trials of hyperparameter optimization on the training samples of datasets B, with early stopping enabled. For the transfer-learning models (TL-CNN and TL-STGCN), the optimized hyperparameters are applied to the backbone networks trained on dataset B, after which a lightweight fine-tuning stage is performed using a small subset of cases from dataset C. Table S1 presents the set of hyperparameters to be optimized for each method.

In the experiments, the batch size is set to 64, the number of epochs is 50, and the initial learning rate is 0.005. The gradient optimization is performed using the adam algorithm. For DT, which employs C4.5 algorithm, its assessment performance is affected by hyperparameters such as criterion, min splits, and max depth. MLP's construction considers the number of model layers and the feature output dimensions of the hidden layers. CNN's performance is affected by kernel size, layers, and hidden features. The hyperparameters of LSTM include layers and hidden layer feature outputs. The hyperparameters of STGCN include layers, chebyshev filters, time filters, and hidden features. For the proposed ASTGL, it incorporates an additional regularization hyperparameter for adaptive graph learning compared to STGCN. The final layers of MLP, CNN, LSTM, STGCN, and ASTGL all contain a classification module with hidden features of 64, 64, 128, 256, 256, respectively. Additionally, due to the lack of ability to record the temporal order of features, DT's feature input needs to be correspondingly adjusted to a 1-D dimension. For TL-CNN and TL-STGCN, the backbone structures remain identical to their non-transfer counterparts (CNN and STGCN), and only a selected portion of layers is unfrozen during fine-tuning. After the backbone models are first trained on dataset B using the optimized hyperparameters, a lightweight fine-tuning stage is performed on a small subset of cases from dataset C. The hyperparameters considered for transfer learning therefore include the number of unfrozen layers and the learning-rate

TABLE S1
OPTIMIZING MODEL HYPERARAMETERS WITH OPTUNA

Model	Param	Options	Dataset B/C
DT	Criterion	{gini, entropy}	gini
	Min. split	{2, 3, …, 5}	2
	Max. depth	{1, 2, …, 30}	20
MLP	Layers	{1, 2, 3}	3
	Hidden features	{32, 64, 128, 256}	{32, 256, 64}
CNN	Layers	{1, 2, 3}	3
	Kernel size	{1, 3, 5}	3
	Hidden features	{32, 64, 128}	{64, 128, 64}
TL-CNN	Unfrozen layers	{1, 2, 3}	2
	Fine-tune LR	{0.001, 0.002, …, 0.01}	0.004
LSTM	Layers	{1, 2, 3}	3
	Hidden features	{64, 128, 256}	{128, 128, 128}
STGCN	Layers	{1, 2, 3}	2
	K_s	{2, 3}	3
	Cheb filters	{5, 10, 15, 20}	10
	Time filters	{5, 10, 15}	5
TL-STGCN	Unfrozen layers	{1, 2, 3}	1
	Fine-tune LR	{0.001, 0.002, …, 0.01}	0.003
ASTGL	Layers	{1, 2, 3}	2
	λ	{0.0001, 0.0002, …, 0.001}	0.0001
	K_s	{2, 3}	3
	Cheb filters	{5, 10, 15, 20}	10
	Time filters	{5, 10, 15}	5

TABLE S2
PRIMARY STRUCTURE OF THE PROPOSED ASTGL MODEL

Layer	Output Shape	Hyperparameters
Input–X1	$(B, 50, 69, F_{in} = 3)$	–
AGLM	$(B, 69, 69)$	Learnable parameter: $\boldsymbol{w}_{ij}^{A} \in \mathbb{R}^{B\times 69\times 3\times 50}$; activation: ReLU + Softmax.
SAM	$(B, 69, 69)$	Learnable parameter: $\boldsymbol{w}^{\text{sp}} \in \mathbb{R}^{B\times 69\times 69}$; activation: Sigmoid + Softmax.
Re-STGCN block1	–	Input: $X1$ $(B, 50, 69, 3)$ + A^{sp} $(B, 69, 69)$; contains: Cheb GCN + TCN + Residual Path.
(1) ChebGCN	$(B, 50, 69, 10)$	$K_s = 2$; filter: $\Theta \in \mathbb{R}^{2\times 3\times 10}$; activation: ReLU.
(2) TCN	$(B, 10, 69, 50)$	Conv2D kernel: 1×3; stride: [1,1]; padding: [0,1]; activation: ReLU.
(3) Residual Path–X2	$(B, 5, 69, 50)$	Conv2D kernel: 1×1; stride: [1,1]; LayerNorm.
Re-STGCN block2	–	Same architecture as block1; input: $X2$ $(B, 5, 69, 50)$ + A^{sp} $(B, 69, 69)$.
(1) ChebGCN	$(B, 50, 69, 10)$	$K_s = 2$; filter: $\Theta \in \mathbb{R}^{2\times 5\times 10}$; activation: ReLU.
(2) TCN	$(B, 10, 69, 50)$	Conv2D kernel: 1×3; stride: [1,1]; padding: [0,1]; activation: ReLU.
(3) Residual Path	$(B, 5, 69, 50)$	Conv2D kernel: 1×1; stride: [1,1]; LayerNorm.
CM	$(B, 2)$	Flatten $\rightarrow$ Linear($69 \times 5 \times 50 \rightarrow 2$); activation: Softmax.
Output	$(B, 2)$	Predicted class probabilities (stable / unstable).

Note: B represents the number of cases in each mini batch. L= 50 is the number of the OTW data points. N= 69 represents the number of the bus in Case study 1. $F_{in} = \{P, Q, V\}$ represents injection active power, reactive power, and voltage magnitude per bus. During backpropagation, the AGLM incorporates a regularization coefficient $\lambda = 0.0001$ to constrain the adaptive learnable matrix Aadp and maintain numerical stability.

(LR) scaling factor used during fine-tuning.

To provide a clear reference for reproducibility, Table S2 summarizes the primary structure of the proposed ASTGL model, including its layer organization, output dimensionality, and the hyperparameter settings adopted throughout this study.