Replication of "Self-Supervised Learning for Large-Scale Preventive Security Constrained DC Optimal Power Flow"

Vinoothna Mahankali

10th March

1 Mathematical Formulation

1.1 Primal-Dual Learning for SCOPF

Input

System configuration

$$x = (d, c, \underline{g}, \overline{g})$$

Input Parameters

Load demand vector	d
Generation cost coefficients	c
Generator limits	$\underline{g}, \overline{g}$

Step 1 — Neural Network Prediction

The primal network first produces an initial estimate of the nominal dispatch

$$\breve{g} = P_\theta(x) \tag{1}$$

This output satisfies generator bounds.

Step 2 — Power Balance Repair Layer

$$\tilde{g} = \begin{cases} (1-\zeta^{\uparrow})\breve{g} + \zeta^{\uparrow}\overline{g} & \text{if } 1^T\breve{g} < 1^T d \\ (1-\zeta^{\downarrow})\breve{g} + \zeta^{\downarrow}\underline{g} & \text{otherwise} \end{cases}$$

where

$$\zeta^{\uparrow} = \frac{1^T d - 1^T\breve{g}}{1^T\overline{g} - 1^T\breve{g}} \qquad \zeta^{\downarrow} = \frac{1^T\breve{g} - 1^T d}{1^T\breve{g} - 1^T\underline{g}}$$

This ensures $\mathbf{1}^T\tilde{g} = \mathbf{1}^T d$.

Variables

Initial generator dispatch predicted by neural network	$\breve{g}$
Corrected dispatch after repair layer	$\tilde{g}$
Vector of ones	$\mathbf{1}$
Scaling factor (generation deficit)	$\zeta^{\uparrow}$
Scaling factor (generation surplus)	$\zeta^{\downarrow}$

Step 3 — Binary Search Layer for Generator Contingencies

$$\tilde{g}_{k,i} = \min(g_i + n_k\gamma_i\hat{g}_i, \bar{g}_i)$$

$$\tilde{g}_{k,k} = 0$$

Binary search initialization

$$n_k = 0.5, \quad n_{\min} = 0, \quad n_{\max} = 1$$

Iteration

$$g_{k,i}^{(j)} = \min(g_i + n_k\gamma_i\hat{g}_i, \bar{g}_i)$$

$$e_k = \mathbf{1}^T g_k^{(j)} - \mathbf{1}^T d$$

Update

$$n_{\max} = n_k \text{ if } e_k > 0 \qquad n_{\min} = n_k \text{ otherwise}$$

$$n_k = \frac{n_{\max} + n_{\min}}{2}$$

After convergence

$$\rho_{k,i} = \begin{cases} 1 & g_i + n_k \gamma_i \hat{g}_i > \bar{g}_i \\ 0 & \text{otherwise} \end{cases}$$

Variables

Dispatch of generator i under contingency k	$\tilde{g}_{k,i}$
Base-case dispatch of generator i	$\tilde{g}_i$
APR response signal	n_k
Droop coefficient	γ_i
Generator capacity	$\hat{g}_i = \bar{g}_i - \underline{g}_i$
Binary upper-limit indicator	$\rho_{k,i}$
Generator contingency index	k

Step 4 — Slack Variables for Thermal Limits

$$\tilde{\eta}_0 = \max\{0, f - \bar{f}, \underline{f} - f\}$$

$$f = K(d - B\tilde{g})$$

$$\tilde{\eta}_k = \max\{0, K(d - B\tilde{g}_k) - \bar{f}, \underline{f} - K(d - B\tilde{g}_k)\}$$

$$\tilde{\eta}_k = \max\{0, f + f_k L_k - \bar{f}, \underline{f} - (f + f_k L_k)\}$$

Variables

Base-case slack variable	$\tilde{\eta}_0$
Contingency slack variable	$\tilde{\eta}_k$
Base-case power flow vector	f
Power flow definition	$f = K(d - B\tilde{g})$
Contingency power flow	f_k
Line flow limits	$\underline{f}, \bar{f}$
PTDF matrix	K
Generator–bus incidence matrix	B
LODF column	L_k

Step 5 — Relaxed Constraints

- Power balance and generator limits enforced by repair layer
- APR constraints satisfied by binary search layer
- Slack bounds satisfied through slack computation

Remaining constraint

$$h_x(y)_k = \mathbf{1}^T \tilde{g}_k - \mathbf{1}^T d$$

Step 6 — Dual Network

$$\lambda = D_\phi(x)$$

Variables

Dual neural network	D_ϕ
Network parameters	ϕ
Lagrange multipliers	λ
Input configuration	x

References

[1] S. Park and P. Van Hentenryck, “Self-Supervised Learning for Large-Scale Preventive Security-Constrained DC Optimal Power Flow,” *IEEE Transactions on Power Systems*, vol. 40, no. 3, pp. 2205–2216, 2025.