

Hybrid Learning Approaches for the AC-OPF

I. Repiso S. Pineda J. M. Morales

OASYS research group, University of Málaga (Spain)

SEIO 2026 | IS Optimization and Statistical Learning in Energy

Santiago de Compostela, 3 September 2026

Why?

- **Power-system operation** requires repeated decisions under **strict time limits**.
- **Exact AC physics** yields **nonconvex and computationally demanding** problems.
- **Convex approximations** scale efficiently but may be **AC-inconsistent**.
- **Learning from past AC solutions** enables **fast and physically reliable** decisions.

$$\min_{\mathbf{p}_g, \mathbf{q}_g, \mathbf{v}, \theta} \sum_{g \in \mathcal{G}} c_g p_g \quad \text{s.t.} \quad (\mathbf{p}_g, \mathbf{q}_g, \mathbf{v}, \theta) \in \mathcal{X}^{\text{AC}}(\mathbf{p}^d, \mathbf{q}^d)$$

$$\min_{\mathbf{p}_g, \mathbf{q}_g, \mathbf{v}, \theta} \sum_{g \in \mathcal{G}} c_g p_g \quad \text{s.t.} \quad (\mathbf{p}_g, \mathbf{q}_g, \mathbf{v}, \theta) \in \mathcal{X}^{\text{AC}}(\mathbf{p}^d, \mathbf{q}^d)$$

$$\mathcal{X}^{\text{AC}}(\mathbf{p}^d, \mathbf{q}^d)$$

$$p_i^g - p_i^d = \sum_{(i,j) \in \mathcal{E}_i} p_{ij}, \quad q_i^g - q_i^d = \sum_{(i,j) \in \mathcal{E}_i} q_{ij}$$

$$p_{ij} = G_{ij}^{\text{ff}} v_i^2 + v_i v_j (G_{ij}^{\text{ft}} \cos \theta_{ij} + B_{ij}^{\text{ft}} \sin \theta_{ij})$$

$$q_{ij} = -B_{ij}^{\text{ff}} v_i^2 + v_i v_j (G_{ij}^{\text{ft}} \sin \theta_{ij} - B_{ij}^{\text{ft}} \cos \theta_{ij})$$

thermal limits

voltage & angle bounds

generation bounds

$$\min_{\mathbf{p}_g, \mathbf{q}_g, \mathbf{v}, \theta} \sum_{g \in \mathcal{G}} c_g p_g \quad \text{s.t.} \quad (\mathbf{p}_g, \mathbf{q}_g, \mathbf{v}, \theta) \in \mathcal{X}^{\text{AC}}(\mathbf{p}^d, \mathbf{q}^d)$$

$$\mathcal{X}^{\text{AC}}(\mathbf{p}^d, \mathbf{q}^d)$$

$$p_i^g - p_i^d = \sum_{(i,j) \in \mathcal{E}_i} p_{ij}, \quad q_i^g - q_i^d = \sum_{(i,j) \in \mathcal{E}_i} q_{ij}$$

$$p_{ij} = G_{ij}^{\text{ff}} v_i^2 + v_i v_j (G_{ij}^{\text{ft}} \cos \theta_{ij} + B_{ij}^{\text{ft}} \sin \theta_{ij})$$

$$q_{ij} = -B_{ij}^{\text{ff}} v_i^2 + v_i v_j (G_{ij}^{\text{ft}} \sin \theta_{ij} - B_{ij}^{\text{ft}} \cos \theta_{ij})$$

thermal limits

voltage & angle bounds

generation bounds

$(\mathbf{v}, \theta) \mapsto (p_{ij}, q_{ij})$ via $v_i v_j^* \Rightarrow \mathcal{X}^{\text{AC}}(\mathbf{p}^d, \mathbf{q}^d)$ **nonconvex** $\Rightarrow$ AC-OPF is well known to be **NP-hard**!

- Fast proxies (data-driven), **no guarantees**.
- **Lower bounds**, optimality gaps, and **infeasibility certificates**.
- Jabr **SOCP** is **cheap, scalable** and easy to **embed** in grid decisions.

- Fast proxies (data-driven), **no guarantees**.
- **Lower bounds**, optimality gaps, and **infeasibility certificates**.
- Jabr **SOCP** is **cheap, scalable** and easy to **embed** in grid decisions.

$$w_i := v_i^2, \quad w_{ij}^r := v_i v_j \cos \theta_{ij}, \quad w_{ij}^i := v_i v_j \sin \theta_{ij}, \quad \xi = (w, w^r, w^i)$$

- Fast proxies (data-driven), **no guarantees**.
- **Lower bounds**, optimality gaps, and **infeasibility certificates**.
- Jabr **SOCP** is **cheap, scalable** and easy to **embed** in grid decisions.

$$w_i := v_i^2, \quad w_{ij}^r := v_i v_j \cos \theta_{ij}, \quad w_{ij}^i := v_i v_j \sin \theta_{ij}, \quad \xi = (\mathbf{w}, \mathbf{w}^r, \mathbf{w}^i)$$

$$\min_{\xi} \sum_{g \in \mathcal{G}} c_g p_g \quad \text{s.t.} \quad (\mathbf{p}_g, \mathbf{q}_g, \xi) \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d)$$

- Fast proxies (data-driven), **no guarantees**.
- **Lower bounds**, optimality gaps, and **infeasibility certificates**.
- Jabr **SOCP** is **cheap**, **scalable** and easy to **embed** in grid decisions.

$$w_i := v_i^2, \quad w_{ij}^r := v_i v_j \cos \theta_{ij}, \quad w_{ij}^i := v_i v_j \sin \theta_{ij}, \quad \xi = (w, w^r, w^i)$$

$$\min_{\xi} \sum_{g \in \mathcal{G}} c_g p_g \quad \text{s.t.} \quad (p_g, q_g, \xi) \in \mathcal{X}^{\text{JB}}(p^d, q^d)$$

(SOCP relaxation)

$$p_i^g - p_i^d = \sum_{(i,j) \in \mathcal{E}_i} p_{ij}, \quad q_i^g - q_i^d = \sum_{(i,j) \in \mathcal{E}_i} q_{ij}$$

$$p_{ij} = G_{ij}^{\text{ff}} w_i + G_{ij}^{\text{ft}} w_{ij}^r + B_{ij}^{\text{ft}} w_{ij}^i$$

$$q_{ij} = -B_{ij}^{\text{ff}} w_i - B_{ij}^{\text{ft}} w_{ij}^r + G_{ij}^{\text{ft}} w_{ij}^i$$

$$(w_{ij}^r)^2 + (w_{ij}^i)^2 \leq w_i w_j$$

thermal limits, generation bounds, lifted voltage bounds

Analytic relaxation strengthening

- ✓ **Valid lower bounds**, so every solution comes certified.
- ✓ **Physics stays inside**, network constraints are never approximated.

Analytic relaxation strengthening

- ✓ **Valid lower bounds**, so every solution comes certified.
- ✓ **Physics stays inside**, network constraints are never approximated.
- ✗ **Large formulation overhead**, lifted variables and cuts increase model size.
- ✗ **OBBT and cuts cost solves**, tightness is bought with runtime.

Analytic relaxation strengthening

- ✓ **Valid lower bounds**, so every solution comes certified.
- ✓ **Physics stays inside**, network constraints are never approximated.
- ✗ **Large formulation overhead**, lifted variables and cuts increase model size.
- ✗ **OBBT and cuts cost solves**, tightness is bought with runtime.

Data-driven surrogates

- ✓ **Training pays off once**, inference is then near-instantaneous.
- ✓ **Regime adaptivity**, structure shared across instances is learned.

Analytic relaxation strengthening

- ✓ **Valid lower bounds**, so every solution comes certified.
- ✓ **Physics stays inside**, network constraints are never approximated.
- ✗ **Large formulation overhead**, lifted variables and cuts increase model size.
- ✗ **OBBT and cuts cost solves**, tightness is bought with runtime.

Data-driven surrogates

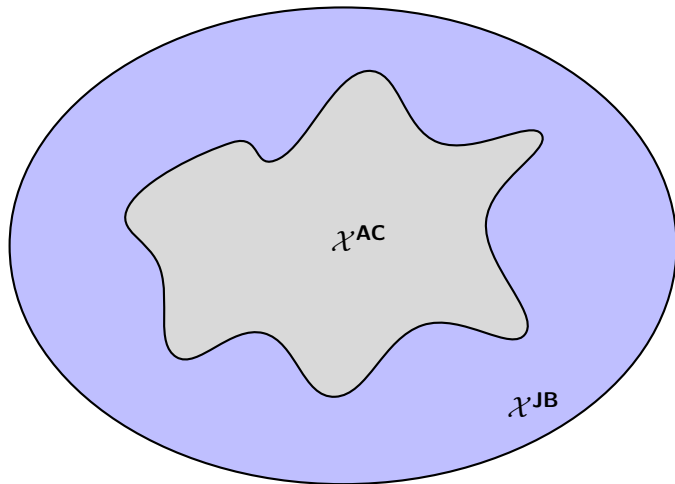
- ✓ **Training pays off once**, inference is then near-instantaneous.
- ✓ **Regime adaptivity**, structure shared across instances is learned.
- ✗ **No feasibility, no certificate**, predictions carry no guarantee.
- ✗ **Out-of-distribution fragility**, accuracy decays away from training data.

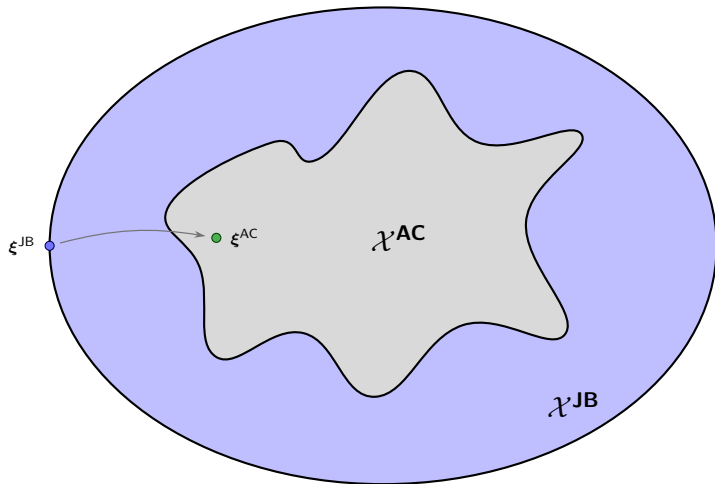
Can **data-driven guidance** steer the SOCP relaxation toward the **AC-feasible manifold** while preserving convex tractability?

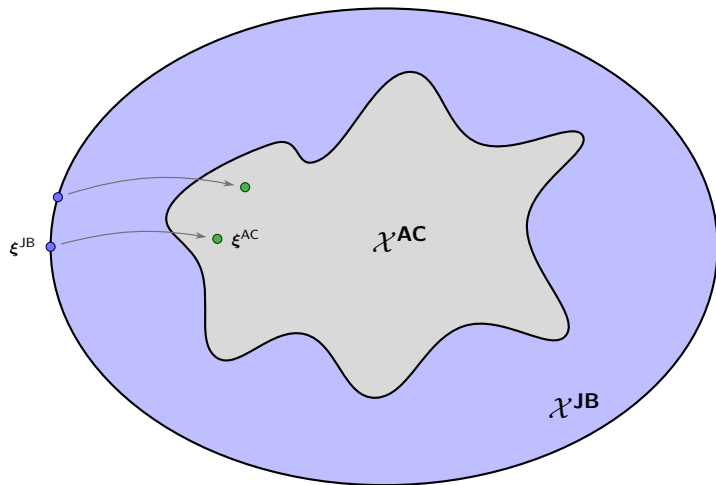
Can **data-driven guidance** steer the SOCP relaxation toward the **AC-feasible manifold** while preserving convex tractability?

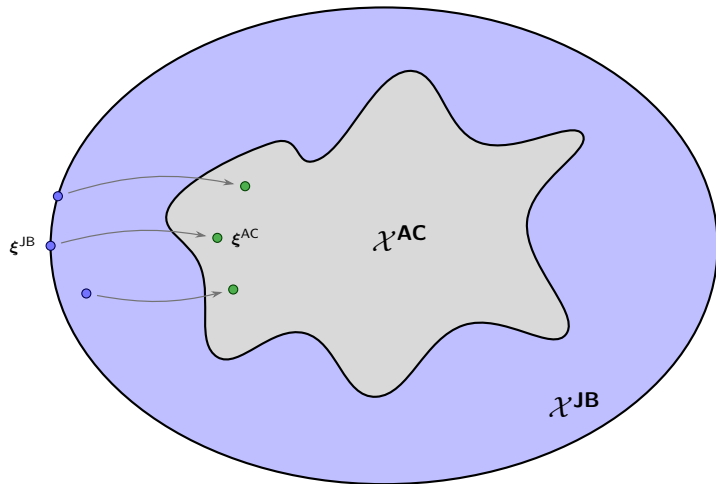
So what can we do?

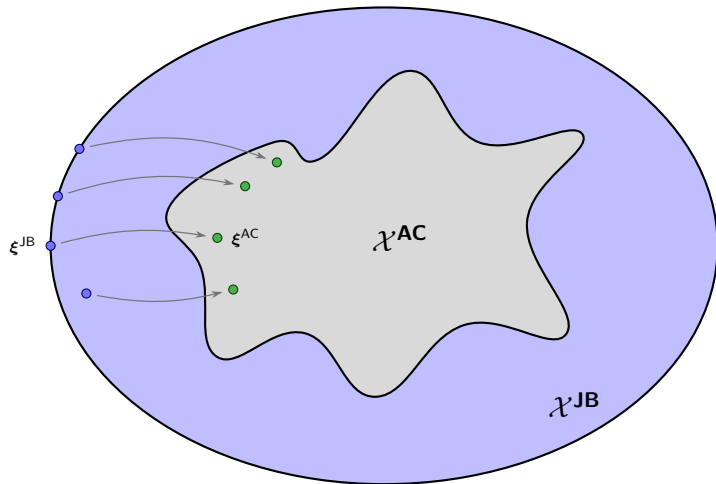
- From strengthening, **convexity** and network constraints satisfied by construction.
- From learning, **regime adaptivity** and guidance toward the AC manifold.
- Combined, a **single conic program** with an interpretable correction.

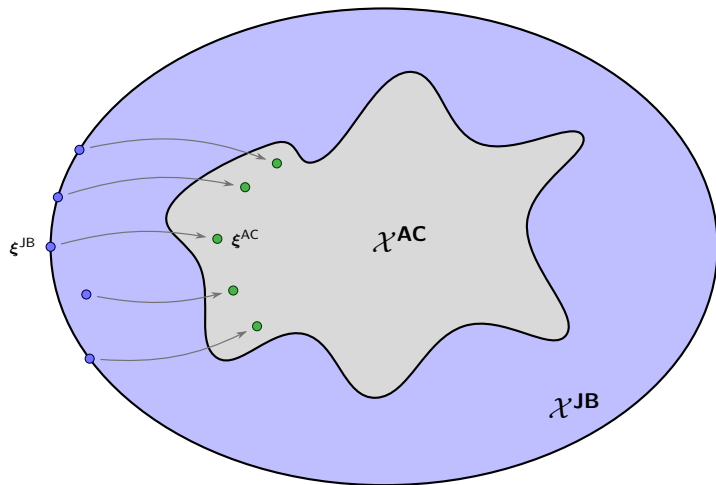


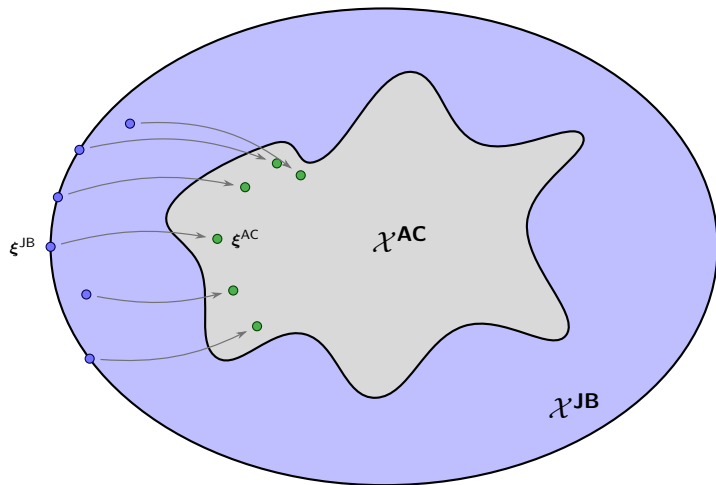


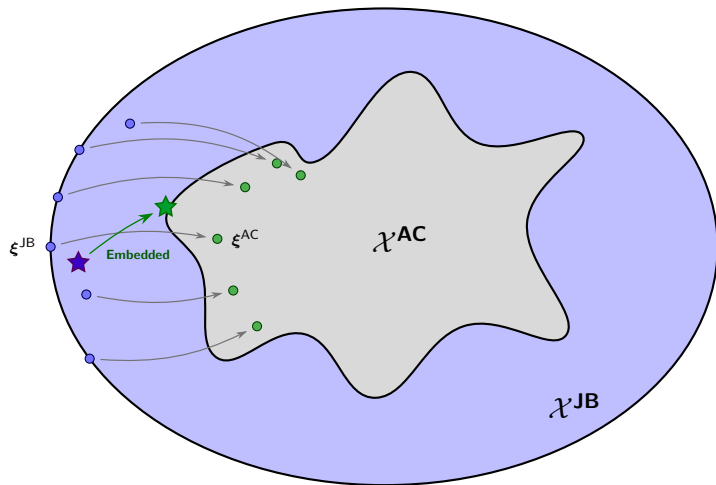


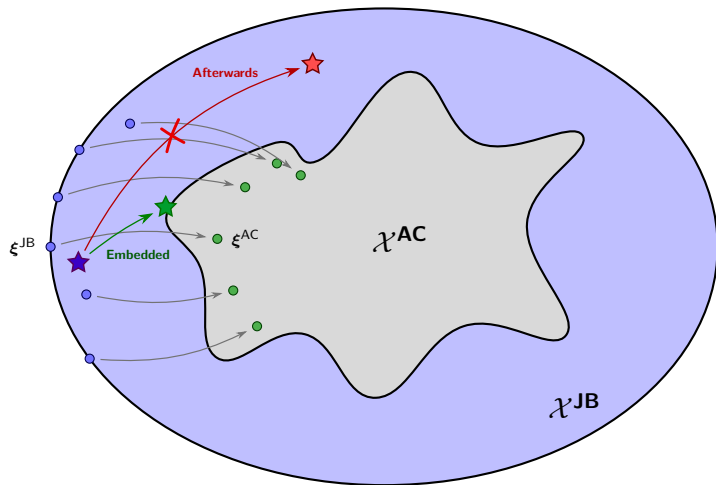












Offline fit

$$\mathcal{D}_\xi = \{(\xi_n^{\text{JB}}, \xi_n^{\text{AC}})\}_{n=1}^N, \quad \Theta^* = \arg \min_{\Theta} \frac{1}{N} \sum_n \ell(h_{\Theta}(\xi_n^{\text{JB}}), \xi_n^{\text{AC}})$$

- Solve both problems on the same demand, then pair the solutions.
- $\xi_n^{\text{JB}} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d)$ and $\xi_n^{\text{AC}} \in \mathcal{X}^{\text{AC}}(\mathbf{p}^d, \mathbf{q}^d)$, both in $\mathbb{R}^{|\mathcal{N}|+2|\mathcal{E}|}$.
- The estimator learns the **Jabr-to-AC correction** $\xi^{\text{JB}} \mapsto \xi^{\text{AC}}$, once and offline.

Offline fit

$$\mathcal{D}_\xi = \{(\xi_n^{\text{JB}}, \xi_n^{\text{AC}})\}_{n=1}^N, \quad \Theta^* = \arg \min_{\Theta} \frac{1}{N} \sum_n \ell(h_{\Theta}(\xi_n^{\text{JB}}), \xi_n^{\text{AC}})$$

- Solve both problems on the same demand, then pair the solutions.
- $\xi_n^{\text{JB}} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d)$ and $\xi_n^{\text{AC}} \in \mathcal{X}^{\text{AC}}(\mathbf{p}^d, \mathbf{q}^d)$, both in $\mathbb{R}^{|\mathcal{N}|+2|\mathcal{E}|}$.
- The estimator learns the **Jabr-to-AC correction** $\xi^{\text{JB}} \mapsto \xi^{\text{AC}}$, once and offline.

Online embedding

$$\xi^{\text{JB}} \xrightarrow{h_{\Theta^*}} \hat{\xi} \xrightarrow{\|\tilde{\xi} - \hat{\xi}\|_2 \leq \varepsilon} \tilde{\xi} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d)$$

- The map **predicts** where the AC solution should lie, $\hat{\xi}$.
- The cone **keeps the physics**, so $\tilde{\xi}$ stays feasible while tracking $\hat{\xi}$.
- Both steps live in **one SOCP**, coupled by linear constraints only.

$$\min_{\xi^{\text{JB}}, \tilde{\xi}} (1 - \beta) \sum_{g \in \mathcal{G}} c_g p_g^{\text{JB}} + \beta \sum_{g \in \mathcal{G}} c_g \tilde{p}_g$$

$$\begin{aligned}
 \min_{\xi^{\text{JB}}, \tilde{\xi}} \quad & (1 - \beta) \sum_{g \in \mathcal{G}} c_g p_g^{\text{JB}} + \beta \sum_{g \in \mathcal{G}} c_g \tilde{p}_g \\
 \text{s.t.} \quad & \xi^{\text{JB}} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d) \qquad \text{base relaxation}
 \end{aligned}$$

$$\begin{aligned}
 \min_{\xi^{\text{JB}}, \tilde{\xi}} \quad & (1 - \beta) \sum_{g \in \mathcal{G}} c_g p_g^{\text{JB}} + \beta \sum_{g \in \mathcal{G}} c_g \tilde{p}_g \\
 \text{s.t.} \quad & \xi^{\text{JB}} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d) && \text{base relaxation} \\
 & \hat{\xi} = h_{\Theta^*}(\xi^{\text{JB}}) && \text{learned target}
 \end{aligned}$$

$$\min_{\xi^{\text{JB}}, \tilde{\xi}} \quad (1 - \beta) \sum_{g \in \mathcal{G}} c_g p_g^{\text{JB}} + \beta \sum_{g \in \mathcal{G}} c_g \tilde{p}_g$$

$$\text{s.t.} \quad \xi^{\text{JB}} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d)$$

base relaxation

$$\hat{\xi} = h_{\Theta^*}(\xi^{\text{JB}})$$

learned target

$$\tilde{\xi} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d)$$

corrected point

$$\begin{aligned}
& \min_{\xi^{\text{JB}}, \tilde{\xi}} \quad (1 - \beta) \sum_{g \in \mathcal{G}} c_g p_g^{\text{JB}} + \beta \sum_{g \in \mathcal{G}} c_g \tilde{p}_g \\
& \text{s.t.} \quad \xi^{\text{JB}} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d) && \text{base relaxation} \\
& \quad \hat{\xi} = h_{\Theta^*}(\xi^{\text{JB}}) && \text{learned target} \\
& \quad \tilde{\xi} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d) && \text{corrected point} \\
& \quad \|\tilde{\xi} - \hat{\xi}\|_2 \leq \varepsilon && \text{trust region}
\end{aligned}$$

$$\begin{aligned}
& \min_{\xi^{\text{JB}}, \tilde{\xi}} \quad (1 - \beta) \sum_{g \in \mathcal{G}} c_g p_g^{\text{JB}} + \beta \sum_{g \in \mathcal{G}} c_g \tilde{p}_g \\
& \text{s.t.} \quad \xi^{\text{JB}} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d) && \text{base relaxation} \\
& \quad \hat{\xi} = h_{\Theta^*}(\xi^{\text{JB}}) && \text{learned target} \\
& \quad \tilde{\xi} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d) && \text{corrected point} \\
& \quad \|\tilde{\xi} - \hat{\xi}\|_2 \leq \varepsilon && \text{trust region} \\
& \quad \beta \in (0, 1), \quad \varepsilon > 0
\end{aligned}$$

$$\begin{aligned}
& \min_{\xi^{\text{JB}}, \tilde{\xi}} \quad (1 - \beta) \sum_{g \in \mathcal{G}} c_g p_g^{\text{JB}} + \beta \sum_{g \in \mathcal{G}} c_g \tilde{p}_g \\
& \text{s.t.} \quad \xi^{\text{JB}} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d) && \text{base relaxation} \\
& \quad \hat{\xi} = h_{\Theta^*}(\xi^{\text{JB}}) && \text{learned target} \\
& \quad \tilde{\xi} \in \mathcal{X}^{\text{JB}}(\mathbf{p}^d, \mathbf{q}^d) && \text{corrected point} \\
& \quad \|\tilde{\xi} - \hat{\xi}\|_2 \leq \varepsilon && \text{trust region} \\
& \quad \beta \in (0, 1), \quad \varepsilon > 0
\end{aligned}$$

- $\beta \rightarrow 0$: cheapest Jabr point whose prediction admits a Jabr-feasible correction.
- $\beta \rightarrow 1$: cheapest Jabr-feasible correction within the regression trust region.
- $\varepsilon \rightarrow 0$: the learned prediction must be Jabr-feasible.
- $\varepsilon \rightarrow \infty$: base and corrected points become decoupled.

Offline estimation

Sample $(\mathbf{p}^d, \mathbf{q}^d)$ and solve the Jabr SOCP and the AC-OPF



Assemble $\mathcal{D}_\xi = \{(\xi_n^{\text{JB}}, \xi_n^{\text{AC}})\}_{n=1}^N$ and train the predictor h_{Θ^*}

Offline estimation

Sample $(\mathbf{p}^d, \mathbf{q}^d)$ and solve the Jabr SOCP and the AC-OPF



Assemble $\mathcal{D}_\xi = \{(\xi_n^{\text{JB}}, \xi_n^{\text{AC}})\}_{n=1}^N$ and train the predictor h_{Θ^*}

Θ^*

Online conic program

Obtain $\tilde{\xi}$ and $(\tilde{\mathbf{p}}_g, \tilde{\mathbf{q}}_g)$ by solving the learning-corrected SOCP

Offline estimation

Sample $(\mathbf{p}^d, \mathbf{q}^d)$ and solve the Jabr SOCP and the AC-OPF



Assemble $\mathcal{D}_\xi = \{(\xi_n^{\text{JB}}, \xi_n^{\text{AC}})\}_{n=1}^N$ and train the predictor h_{Θ^*}

Θ^*

Online conic program

Obtain $\tilde{\xi}$ and $(\tilde{\mathbf{p}}_g, \tilde{\mathbf{q}}_g)$ by solving the learning-corrected SOCP

Recovery and assessment

AC power flow restoration



Accuracy, constraint violations and ℓ_1 re-dispatch

- **Four PGLib-OPF networks**, 14–162 buses.
- Demands perturbed $\pm 20\%$ around nominal.
- Pooled out-of-sample averages, **five 10,000/500** train-test splits.
- Conic solved with **MOSEK 11.2.0**, AC with **IPOPT 3.12.13**.
- Balanced setting $\beta = 0.5$ and $\varepsilon = 10^{-6}$.
- Approaches compared:
 - **SOCP**: the standard Jabr relaxation.
 - **SOCP-LR**: bi-objective SOCP with affine correction.

Test case	App.	Voltage magnitude		Reactive generation		ΔC [%]
		RMSE [p.u.]	VIOL [%]	RMSE [p.u.]	VIOL [%]	
case14_ieee	SOCP	4.09×10^{-4}	5.06×10^{-11}	7.11×10^{-3}	1.31×10^0	0.193
	SOCP-LR	3.11×10^{-5}	0	4.40×10^{-4}	2.93×10^{-4}	0.001
case39_epri	SOCP	6.68×10^{-3}	1.02×10^{-1}	2.00×10^{-1}	6.74×10^{-2}	5.378
	SOCP-LR	1.56×10^{-3}	1.35×10^{-3}	3.14×10^{-2}	2.49×10^{-2}	0.003
case57_ieee	SOCP	2.52×10^{-3}	3.61×10^{-2}	7.52×10^{-2}	1.11×10^0	0.014
	SOCP-LR	1.84×10^{-4}	3.72×10^{-4}	9.11×10^{-3}	3.83×10^{-4}	0.002
case162_ieee	SOCP	4.27×10^{-2}	2.69×10^0	1.00×10^0	0	0.658
	SOCP-LR	1.85×10^{-2}	4.42×10^{-1}	4.16×10^{-1}	2.84×10^{-2}	0.312

Test case	App.	Voltage magnitude		Reactive generation		ΔC [%]
		RMSE [p.u.]	VIOL [%]	RMSE [p.u.]	VIOL [%]	
case14_ieee	SOCP	4.09×10^{-4}	5.06×10^{-11}	7.11×10^{-3}	1.31×10^0	0.193
	SOCP-LR	3.11×10^{-5}	0	4.40×10^{-4}	2.93×10^{-4}	0.001
case39_epri	SOCP	6.68×10^{-3}	1.02×10^{-1}	2.00×10^{-1}	6.74×10^{-2}	5.378
	SOCP-LR	1.56×10^{-3}	1.35×10^{-3}	3.14×10^{-2}	2.49×10^{-2}	0.003
case57_ieee	SOCP	2.52×10^{-3}	3.61×10^{-2}	7.52×10^{-2}	1.11×10^0	0.014
	SOCP-LR	1.84×10^{-4}	3.72×10^{-4}	9.11×10^{-3}	3.83×10^{-4}	0.002
case162_ieee	SOCP	4.27×10^{-2}	2.69×10^0	1.00×10^0	0	0.658
	SOCP-LR	1.85×10^{-2}	4.42×10^{-1}	4.16×10^{-1}	2.84×10^{-2}	0.312

- Reactive violations drop by **at least three orders of magnitude** on smaller systems.
- Voltage violations fall by **almost two orders of magnitude** on the 39- and 57-bus systems.
- Both voltage and reactive **RMSE are substantially reduced** across all networks.
- The corrective effort ΔC is **reduced across all systems**, by up to three orders of magnitude.

- SOCP relaxation can be corrected by a **map learned from past solutions**.
- Embedded as **affine constraints**, so **convexity** is preserved.
- **RMSEs** drop by up to $10\times$, **violations** by at least $10^3\times$, and **AC-recovery cost** by up to $10^3\times$.
- Ongoing work targets **structured regression** and **ICNNs**.

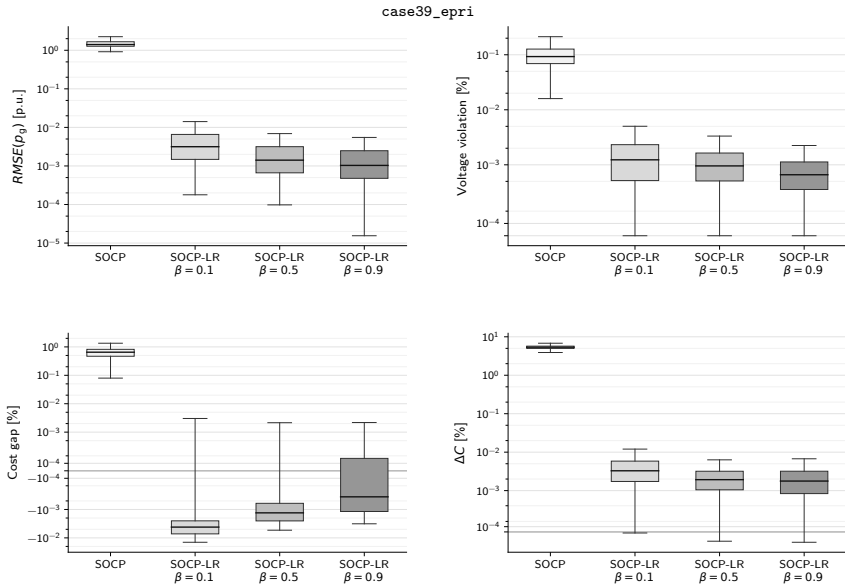


Thanks for your attention! Questions?

More info: oasys.uma.es

Email: irepiso@uma.es

Influence of the trade-off parameter β



Test case	App.	v [p.u.]	θ [rad]	p_g [p.u.]	q_g [p.u.]
case14_ieee	SOCP	4.09×10^{-4}	8.01×10^{-5}	1.84×10^{-5}	7.11×10^{-3}
	$\beta = 0.1$	6.09×10^{-5}	1.36×10^{-5}	1.42×10^{-5}	8.87×10^{-4}
	$\beta = 0.5$	3.11×10^{-5}	8.21×10^{-6}	1.38×10^{-5}	4.40×10^{-4}
	$\beta = 0.9$	1.25×10^{-5}	5.70×10^{-6}	1.37×10^{-5}	1.54×10^{-4}
case39_epri	SOCP	6.68×10^{-3}	4.73×10^{-2}	1.49×10^0	2.00×10^{-1}
	$\beta = 0.1$	2.22×10^{-3}	7.90×10^{-4}	6.66×10^{-3}	4.11×10^{-2}
	$\beta = 0.5$	1.56×10^{-3}	5.23×10^{-4}	2.70×10^{-3}	3.14×10^{-2}
	$\beta = 0.9$	1.21×10^{-3}	4.00×10^{-4}	2.43×10^{-3}	2.54×10^{-2}
case57_ieee	SOCP	2.52×10^{-3}	4.55×10^{-3}	1.11×10^{-1}	7.52×10^{-2}
	$\beta = 0.1$	2.53×10^{-4}	7.59×10^{-4}	1.37×10^{-2}	1.25×10^{-2}
	$\beta = 0.5$	1.84×10^{-4}	4.86×10^{-4}	7.95×10^{-3}	9.11×10^{-3}
	$\beta = 0.9$	1.48×10^{-4}	3.28×10^{-4}	4.99×10^{-3}	7.45×10^{-3}
case162_ieee	SOCP	4.27×10^{-2}	5.18×10^{-2}	1.10×10^0	1.00×10^0
	$\beta = 0.1$	5.24×10^{-3}	1.76×10^{-2}	1.92×10^{-1}	1.90×10^{-1}
	$\beta = 0.5$	1.85×10^{-2}	2.43×10^{-2}	3.99×10^{-1}	4.16×10^{-1}
	$\beta = 0.9$	3.04×10^{-2}	3.01×10^{-2}	5.04×10^{-1}	5.87×10^{-1}

Test case	App.	$\mathbf{v}$ [%]	$\mathbf{p}_g$ [%]	$\mathbf{q}_g$ [%]	$\mathbf{s}$ [%]
case14_ieee	SOCP	5.06×10^{-11}	2.00×10^{-7}	1.31×10^0	0
	$\beta = 0.1$	0	1.23×10^{-9}	2.14×10^{-4}	0
	$\beta = 0.5$	0	2.13×10^{-7}	2.93×10^{-4}	0
	$\beta = 0.9$	0	2.27×10^{-7}	3.03×10^{-4}	0
case39_epri	SOCP	1.02×10^{-1}	0	6.74×10^{-2}	8.60×10^{-1}
	$\beta = 0.1$	1.75×10^{-3}	0	3.94×10^{-2}	4.17×10^{-4}
	$\beta = 0.5$	1.35×10^{-3}	3.68×10^{-7}	2.49×10^{-2}	3.64×10^{-4}
	$\beta = 0.9$	1.05×10^{-3}	9.76×10^{-7}	1.97×10^{-2}	4.39×10^{-4}
case57_ieee	SOCP	3.61×10^{-2}	1.80×10^{-1}	1.11×10^0	0
	$\beta = 0.1$	3.52×10^{-4}	4.52×10^{-6}	5.49×10^{-4}	0
	$\beta = 0.5$	3.72×10^{-4}	1.29×10^{-5}	3.83×10^{-4}	0
	$\beta = 0.9$	4.00×10^{-4}	3.12×10^{-5}	4.49×10^{-4}	0
case162_ieee	SOCP	2.69×10^0	0	0	6.14×10^{-2}
	$\beta = 0.1$	4.76×10^{-2}	0	8.23×10^{-3}	1.76×10^{-4}
	$\beta = 0.5$	4.42×10^{-1}	0	2.84×10^{-2}	1.15×10^{-3}
	$\beta = 0.9$	1.27×10^0	2.70×10^{-9}	3.03×10^{-4}	2.22×10^{-3}

Test case	App.	CG [%]	Δp_g	Δv_g	ΔC [%]
case14_ieee	SOCP	0.001	5.70×10^{-4}	1.16×10^{-4}	0.193
	$\beta = 0.1$	-0.001	6.30×10^{-7}	1.35×10^{-8}	0.001
	$\beta = 0.5$	-0.001	5.99×10^{-7}	1.68×10^{-8}	0.001
	$\beta = 0.9$	-0.001	6.25×10^{-7}	1.78×10^{-8}	0.001
case39_epri	SOCP	0.681	6.05×10^{-1}	7.29×10^{-3}	5.378
	$\beta = 0.1$	-0.008	3.10×10^{-4}	9.72×10^{-5}	0.007
	$\beta = 0.5$	-0.002	2.31×10^{-4}	8.66×10^{-5}	0.003
	$\beta = 0.9$	-0.001	2.85×10^{-4}	8.23×10^{-5}	0.003
case57_ieee	SOCP	0.116	2.91×10^{-2}	1.91×10^{-3}	0.014
	$\beta = 0.1$	-0.007	2.22×10^{-4}	3.04×10^{-5}	0.005
	$\beta = 0.5$	-0.003	3.18×10^{-4}	4.20×10^{-5}	0.002
	$\beta = 0.9$	-0.002	3.03×10^{-4}	5.27×10^{-5}	0.002
case162_ieee	SOCP	6.338	5.53×10^{-1}	2.26×10^{-2}	0.658
	$\beta = 0.1$	0.660	7.73×10^{-2}	5.24×10^{-3}	0.192
	$\beta = 0.5$	1.326	1.68×10^{-1}	1.31×10^{-2}	0.312
	$\beta = 0.9$	1.740	2.17×10^{-1}	1.88×10^{-2}	0.384