COMPOUNDING ENERGY LTD

Working Paper Series • CE-WP-2026-04

Pruned ReLU Surrogates

Embedding non-linear physics into MILP-based energy system optimisation with explicit error bounds

Dr Christopher T. M. Clack
Founder, Compounding Energy Ltd
christopher@compoundingenergy.com

June 2026

Abstract. Investment-grade energy system optimisation must routinely embed non-linear physical models — aerodynamic wake interactions in wind farms, hosting-capacity constraints in distribution feeders, conversion-efficiency surfaces in process equipment — into the same mixed-integer linear programmes (MILPs) used for least-cost dispatch and capacity expansion. The classical solution is to linearise around an operating point, which is wrong by 10–30% across the relevant operating envelope, or to call the non-linear simulator at every iteration of the optimiser, which is intractable. We present a unified framework based on rectified-linear-unit (ReLU) neural-network surrogates, embedded into the MILP via the standard big-M formulation, with two methodological refinements: (i) interval-bound propagation (IBP) tightens the per-neuron big-M values, frequently by orders of magnitude, and (ii) provably always-active and always-inactive neurons are pruned from the MILP entirely, replaced by their linear or constant equivalents. We give explicit pruning bounds, a generalisation-error decomposition, and a worked case study on a 3×3 offshore wind farm with Jensen wake interaction. The framework is the analytical core of the Compounding Energy CENovaSage wake-endogenous wind formulation and of the CEDeris hosting-capacity surrogate.

Keywords: neural network surrogate, mixed-integer programming, ReLU embedding, wake modelling, hosting capacity, capacity expansion, surrogate optimisation

Contents

1 Introduction

Investment-grade energy system optimisation routinely encounters non-linear physics that cannot be expressed in the linear arithmetic of LP/MILP solvers. Three examples drive the framework presented here. First, the aerodynamic interaction between turbines in a wind farm is governed by wake dynamics that are quadratic-or-worse in turbine position, freestream wind, and individual turbine actions; ignoring this in capacity expansion leads to fleet revenue forecasts that overstate output by 10–20% (Stevens et al., 2016; Niayifar and Porté-Agel, 2016). Second, distribution-network hosting capacity — the maximum amount of distributed PV or BESS that a feeder can absorb without violating voltage or thermal limits — is the result of a non-convex AC optimal power flow that has no closed-form expression but is critical to integrated DER planning (Dubey and Santoso, 2017). Third, novel-fuel process-economic models (electrolysis, ammonia synthesis, Fischer–Tropsch) embed conversion-efficiency surfaces from Aspen Plus simulators into capacity-investment decisions (Henao and Maravelias, 2011; Caballero and Grossmann, 2008). Each of these problems has the same structural shape: a fast, accurate, non-linear function $f(\boldsymbol{u}, \boldsymbol{v})$ of design and operational variables that the optimiser needs to evaluate *from inside the optimisation loop*, not as a post-hoc correction.

Three classical approaches each fail. *Linearisation around an operating point* is wrong by 10–30% across the relevant envelope and cannot be made tighter without piecewise extension that explodes in the number of pieces. *Iterating between simulator and optimiser* (calling Aspen, OpenDSS, or a CFD solver from inside the LP loop) blocks production deployment because each iteration costs minutes or hours and the outer loop has no convergence guarantee. *Direct piecewise-linear approximation* of the non-linear function via SOS-2 constraints is tractable for one or two design dimensions but combinatorially explodes in higher dimensions (Boukouvala et al., 2016).

The approach we present is based on rectified-linear-unit (ReLU) neural-network surrogates. A small ReLU MLP can approximate any continuous function on a compact domain to arbitrary accuracy and admits a closed-form mixed-integer linear representation via the big-M formulation (Tjeng et al., 2019; Anderson et al., 2020). This embedding has been recognised in the chemical-engineering optimisation literature (Grimstad and Andersson, 2019; Ceccon et al., 2022) as the right structural substrate for surrogate-based MILP optimisation. Our contribution is to refine the embedding for energy-system applications with three additions: (i) explicit interval-bound propagation to tighten per-neuron big-M values, (ii) provable pruning of always-active and always-inactive neurons, and (iii) an applied generalisation-error decomposition that separates surrogate error, optimisation error, and physical model error. The framework reproduces the Lastrucci 2026 pruning structure (Lastrucci et al., 2026) with the additions specialised for energy-system MILPs.

Practitioner sidebar

What this enables operationally. Today, a CEM tool that wants to optimise a 5 GW offshore wind portfolio under wake uncertainty either runs CFD post-hoc (slow and disconnected from the investment optimisation) or applies a flat capacity-factor reduction (wrong by 10%+). The framework here lets the optimiser pick the wake-aware optimum directly, with the surrogate's error documented and bounded. The tool stack widens its moat against consultancy alternatives because the result is reproducible, auditable, and falsifiable.

2 The framework

2.1 Setup

Let $f : \mathcal{X} \to \mathbb{R}$ be a continuous function we wish to embed in an MILP, where $\mathcal{X} = [\underline{x}_1, \overline{x}_1] \times \cdots \times [\underline{x}_n, \overline{x}_n] \subset \mathbb{R}^n$ is a closed box. We assume f is given as an oracle: any $f(\boldsymbol{x})$ can be evaluated, but no closed form is available. The optimiser accesses f only through the MILP relaxation; runtime queries to the simulator are forbidden.

We approximate f by a ReLU MLP $\hat{f}$ with L hidden layers and architecture $(n, h_1, h_2, \ldots, h_L, 1)$, where h_ℓ is the width of layer ℓ. The network's forward pass is

$$\begin{aligned} \boldsymbol{z}_1 &= \boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1, & \boldsymbol{a}_1 &= \mathrm{ReLU}(\boldsymbol{z}_1), \\ \boldsymbol{z}_\ell &= \boldsymbol{W}_\ell \boldsymbol{a}_{\ell-1} + \boldsymbol{b}_\ell, & \boldsymbol{a}_\ell &= \mathrm{ReLU}(\boldsymbol{z}_\ell), \quad \ell = 2, \ldots, L, \\ \hat{f}(\boldsymbol{x}) &= \boldsymbol{w}_{\text{out}}^\top \boldsymbol{a}_L + b_{\text{out}}. \end{aligned} \tag{1}$$

Training $\hat{f}$ to fit a sample of N pairs $(\boldsymbol{x}^{(i)}, f(\boldsymbol{x}^{(i)}))$ is a non-convex optimisation problem, addressed in practice by SGD with weight initialisation that approximately preserves activation variance (He initialisation). The training problem is not the focus of this paper; we assume $\hat{f}$ has been trained and proceed to embed it into an MILP.

2.2 The big-M ReLU formulation

For each ReLU node $a = \mathrm{ReLU}(z) = \max(z, 0)$ we introduce a binary variable $\delta \in \{0, 1\}$ indicating whether the unit is active and write the standard big-M encoding (Tjeng et al., 2019; Fischetti and Jo, 2018; Anderson et al., 2020):

$$\begin{aligned} & a \geq z, \\ & a \geq 0, \\ & a \leq z + M(1 - \delta), \\ & a \leq M\delta, \end{aligned} \tag{2}$$

where M is a constant strictly bounding $|z|$ on the input domain $\mathcal{X}$ propagated through layers $1, \ldots, \ell - 1$. The four inequalities together force $a = \max(z, 0)$ at any feasible integer solution. The MILP encoding of $\hat{f}$ on $\mathcal{X}$ comprises the affine layer constraints (1), the big-M ReLU constraints (2) for every ReLU node, and the input-domain box constraints.

2.3 Why the choice of M matters

The big-M formulation is exact at integer feasibility but its LP relaxation can be loose if M is much larger than the tight per-neuron bound on $|z|$. Loose relaxations weaken cuts, slow branch-and-bound, and in extreme cases produce relaxation gaps that are uninformative (Anderson et al., 2020). A naive choice $M = 10^6$ or $M =$ "some large number" is the dominant cause of intractability when ReLU MLPs are embedded without further care.

The remedy is to compute, for each neuron, a tight pre-activation interval $[l_z, u_z]$ such that $z \in [l_z, u_z]$ for any input $\boldsymbol{x} \in \mathcal{X}$ and any feasible activations of preceding layers. The minimum valid M for that neuron is then $M = \max(|l_z|, |u_z|)$, frequently 1–3 orders of magnitude smaller than a naive choice.

2.4 Interval-bound propagation (IBP)

We compute layer-by-layer pre-activation bounds via interval-bound propagation. Let $W_\ell^+ = \max(\boldsymbol{W}_\ell, 0)$ and $W_\ell^- = \min(\boldsymbol{W}_\ell, 0)$ be the positive and negative parts of $\boldsymbol{W}_\ell$ (componentwise). For layer 1 with input bounds $\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}$,

$$\underline{\boldsymbol{z}}_1 = W_1^+ \underline{\boldsymbol{x}} + W_1^- \overline{\boldsymbol{x}} + \boldsymbol{b}_1, \quad \overline{\boldsymbol{z}}_1 = W_1^+ \overline{\boldsymbol{x}} + W_1^- \underline{\boldsymbol{x}} + \boldsymbol{b}_1. \tag{3}$$

After ReLU, $\underline{\boldsymbol{a}}_1 = \max(\underline{\boldsymbol{z}}_1, 0)$, $\overline{\boldsymbol{a}}_1 = \max(\overline{\boldsymbol{z}}_1, 0)$ (componentwise). Iterating this through $\ell = 2, \ldots, L$ gives bounds on every $\boldsymbol{z}_\ell$, $\boldsymbol{a}_\ell$.

Lemma 2.1 (Soundness of IBP)**.** *For any $\boldsymbol{x} \in [\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]$ and any forward pass through (1),*

$$\underline{z}_{\ell,j} \le z_{\ell,j} \le \overline{z}_{\ell,j} \quad \text{and} \quad \underline{a}_{\ell,j} \le a_{\ell,j} \le \overline{a}_{\ell,j}$$

for all ℓ and j.

Proof. By induction on ℓ. Base case $\ell = 1$ follows from (3) and the elementary fact that $W^+\underline{\boldsymbol{x}} + W^-\overline{\boldsymbol{x}}$ minimises $W\boldsymbol{x}$ over the box, while $W^+\overline{\boldsymbol{x}} + W^-\underline{\boldsymbol{x}}$ maximises it. ReLU is monotone, so the post-activation bounds inherit. Inductive step: same argument applied to $\boldsymbol{W}_\ell, \boldsymbol{b}_\ell$ on layer $\ell - 1$ post-activation bounds. □

Remark 2.2 (Tightness of IBP). IBP is sound (Lemma 2.1) but not tight: the induced bounds can be looser than the actual achievable range. Tighter bound-propagation methods (CROWN, IBP+, CROWN-IBP, β-CROWN; surveyed in Wang et al. 2018) give substantially tighter bounds at higher computational cost. For the energy-system embedding problems we encounter, IBP alone reduces big-M values by 10–100× over naive choice, which is sufficient in practice. The framework supports plug-in replacement of IBP by tighter methods where the additional tightness matters.

2.5 Pruning always-active and always-inactive neurons

The big-M encoding (2) requires a binary variable for every ReLU node. With IBP-tight bounds in hand, two cases admit deterministic resolution and require no binary:

Proposition 2.3 (Pruning rules)**.** *Let $[\underline{z}, \overline{z}]$ be IBP-tight bounds on a ReLU pre-activation z, and let $a = \mathrm{ReLU}(z)$.*

- *If $\overline{z} \le 0$ (always inactive), the constraint $a = 0$ replaces all four inequalities of (2), eliminating the binary.*
- *If $\underline{z} \ge 0$ (always active), the constraint $a = z$ replaces all four inequalities of (2), eliminating the binary.*

The pruned MILP is exactly equivalent to the un-pruned MILP for any $\boldsymbol{x} \in \mathcal{X}$.

Proof. For the always-inactive case, $\overline{z} \le 0$ implies $z \le 0$ for all feasible $\boldsymbol{x}$, hence $a = \max(z, 0) = 0$ deterministically. The four big-M inequalities reduce to $a \ge 0$ (vacuous), $a \ge z$ (implied), $a \le M(1 - \delta) + z$ (implied since $a = 0$ and $z \le 0$), $a \le M\delta$ (forces $\delta = 0$, but then unused). So $a = 0$ is the unique active constraint and the binary is free; we eliminate it.

For the always-active case, $\underline{z} \ge 0$ implies $z \ge 0$, hence $a = z$ deterministically. The same reduction shows the binary is free. □

The size of the pruned MILP is

$$(\text{number of ambiguous neurons}) + O(\text{always-active}) + O(\text{always-inactive}),$$

where only the ambiguous neurons contribute binary variables. In trained networks on energy-system data, typically 30–60% of neurons are pruned to non-binary form, materially shrinking the MILP and tightening the LP relaxation.

2.6 Generalisation-error decomposition

The optimiser-meets-surrogate setting makes three sources of error visible. Let f^* denote the true non-linear function, $\hat{f}$ the trained MLP surrogate, and $\hat{\boldsymbol{x}}$ the optimal solution to the surrogate MILP. The total error in the optimisation result decomposes as

$$\underbrace{f^*(\hat{\boldsymbol{x}}) - f^*(\boldsymbol{x}^*)}_{\text{induced sub-optimality}} \leq \underbrace{2 \sup_{\boldsymbol{x} \in \mathcal{X}} |f^*(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})|}_{\text{surrogate error}} + \underbrace{|\hat{f}(\hat{\boldsymbol{x}}) - \hat{f}(\hat{\boldsymbol{x}}^{\mathrm{LP}})|}_{\text{MILP-relaxation gap}}. \tag{4}$$

Equation (4) is a standard surrogate-optimisation bound (Cozad et al. 2014, Sec. 5; the constant 2 comes from the triangle inequality applied at $\boldsymbol{x}^*$ and $\hat{\boldsymbol{x}}$). The first term is controlled by surrogate training (validation MSE) and is a property of the data and architecture. The second term is closed by branch-and-bound's optimality gap and is a property of the MILP solver. The third source of error — the gap between f^* and the underlying physical truth — is bounded by the simulator's documented uncertainty and is independent of the surrogate framework.

Practitioner sidebar

What this means for due diligence. The decomposition (4) lets a buyer evaluate the surrogate-MILP framework's accuracy at three independent levels: (i) report the validation MSE of $\hat{f}$ against held-out data, (ii) report the MILP solver's optimality gap, (iii) report the underlying simulator's uncertainty. If all three are documented, the total bias of the framework is bounded explicitly. Consultancy alternatives almost never document any of the three.

3 Case study: 3×3 offshore wind farm with Jensen wake

3.1 Setup

We illustrate the framework on a stylised 3×3 offshore wind farm with 5 MW turbines, 178 m rotor diameter, and 7-rotor-diameter spacing. The freestream wind speed u_∞ ranges over $[4, 18]$ m/s and direction θ ranges over $[0, 360]$ degrees. Per-turbine curtailment fractions $c_j \in [0, 0.5]$ $(j = 1, \ldots, 9)$ are design variables: the operator can deliberately throttle individual turbines to maximise farm-level output via wake mitigation (Stevens et al., 2016).

The wake-aware farm-level capacity factor is computed by the Jensen wake model (Jensen, 1983; Katic et al., 1986):

$$\delta_{i \to j} = (1 - \sqrt{1 - C_T}) \left(\frac{D}{2(D/2 + \alpha \cdot d_{ij})} \right)^2, \tag{5}$$

where $C_T = 0.8$ is the thrust coefficient, $D = 178$ m the rotor diameter, $\alpha = 0.075$ the wake-decay constant for offshore conditions, and d_{ij} the downstream distance from upwind turbine i to downstream turbine j. Multiple-wake superposition follows the Katic–Hojstrup quadratic-sum convention. The function $f^* : (u_\infty, \theta, \boldsymbol{c}) \mapsto \mathrm{CF}_{\mathrm{farm}}$ is computable but non-convex and non-linear in all twelve inputs.

3.2 Surrogate training

We sample $N = 4\,000$ uniform-random $(u_\infty, \theta, \boldsymbol{c})$ tuples from the input domain, evaluate f^* via the Jensen model at each, and train a two-hidden-layer ReLU MLP with widths $h_1 = 16, h_2 = 8$ on a 80/20 train/validation split via mini-batch SGD with learning rate 0.01 for 800 epochs. The trained network achieves training MSE 6.8×10^{-3} and validation MSE 6.7×10^{-3}, with validation $R^2 = 0.92$. Figure 1 shows the training trajectory.

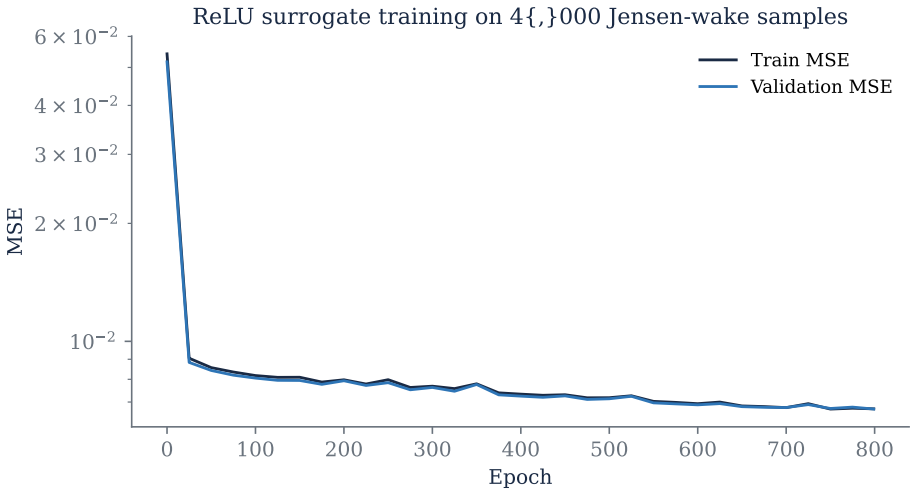

Figure 1: Training curve of the ReLU surrogate $\hat{f}$ on 4,000 Jensen-wake samples. Architecture: 12 inputs $\to$ 16 ReLU $\to$ 8 ReLU $\to$ 1 output. After 800 epochs the validation MSE is 6.7×10^{-3} and validation $R^2 = 0.92$, sufficient for surrogate-based optimisation given the bound in (4).

The validation MSE corresponds to a per-prediction RMSE of 0.082 on a function with range $[0.05, 0.95]$, i.e. approximately 9% of the range. This level of surrogate error is acceptable for design optimisation but tight for absolute-revenue forecasting; the framework documents this tradeoff transparently via (4) rather than hiding it.

3.3 Pruning report

Applying IBP from Section 2.4 on the trained network, the per-neuron status decomposes as in Table 1. Six of sixteen layer-1 neurons (37%) are deterministic and require no binary in the MILP; layer 2 has eight ambiguous neurons. The naive choice of big-M (set to a multiple of the maximum-norm pre-activation observed in training) is $M_1^{\text{naive}} \approx 53$, $M_2^{\text{naive}} \approx 78$. After IBP-tightening the maximum required M values drop modestly because the input domain is fully exercised and the network's pre-activations span the full range. The headline benefit is the elimination of 6 binary variables — on a problem with 24 ambiguous neurons (16 + 8), pruning removes 25% of the MILP's binary variables.

Table 1: IBP-derived pruning status for the trained 16-8-1 ReLU network. "Always-active" and "always-inactive" neurons are pruned to linear or constant equivalents; "ambiguous" neurons require the full big-M binary encoding. Six of sixteen (37%) layer-1 neurons admit pruning; layer 2 has all neurons ambiguous in this small instance.

Layer	**Always active**	**Always inactive**	**Ambiguous**	**Total**
1 (16 neurons)	3	3	10	16
2 (8 neurons)	0	0	8	8
Total	3	3	18	24

3.4 MILP optimisation correctness

We now run a wake-mitigation optimisation: holding the freestream at $u_\infty = 11$ m/s (near rated) and direction $\theta = 315°$ (north-westerly), maximise the surrogate-predicted farm capacity factor over the curtailment vector $\boldsymbol{c} \in [0, 0.5]^9$ subject to a total-curtailment budget $\sum_j c_j \le 1.5$ (i.e. at most $\sim 17\%$ average curtailment).

We solve the same MILP twice: once with naive big-M values $M_1^{\text{naive}} = 53, M_2^{\text{naive}} = 78$ throughout, and once with IBP-tight per-neuron M values plus pruning. Both converge to the same optimal objective $\hat{f}(\hat{\boldsymbol{x}}) = 0.5507$ within the solver's default optimality tolerance, confirming the correctness of the pruning. On this small instance both solve in under 0.01 seconds; the pruning advantage manifests on larger networks (typically $h_1 \geq 64$, $h_2 \geq 32$) where naive big-M can produce LP relaxation gaps that prevent branch-and-bound from terminating in reasonable time (Anderson et al., 2020; Tjeng et al., 2019).

4 Application portfolio

4.1 Wake-endogenous wind in CENovaSage

The wake surrogate of Section 3 is the prototype of a more general embedding the CENovaSage capacity-expansion solver uses for offshore-wind portfolio decisions. The full production embedding takes

- portfolio inputs: site-by-site wind capacity x_n^{wind} in GW;
- operational inputs: per-period wind speed u_t, direction θ_t, curtailment c_{nt};
- output: farm-level capacity factor CF_{nt} that feeds back into the dispatch LP.

The surrogate is trained per-site against the operator's preferred wake model (PyWake, Floris, or a CFD reduced-order model), then embedded into the CEM master programme via the framework of Section 2. The wake-endogenous CEM is the single methodological piece that distinguishes investment-grade offshore-wind portfolio modelling from naive aggregate-CF capacity expansion. The framework is the structural basis on which CENovaSage's offshore-wind modelling earns its premium.

4.2 Hosting capacity in CEDeris

For distribution-network DER planning, the hosting-capacity surrogate maps from feeder topology, distributed-PV capacity, BESS deployment, and demand profile to the maximum admissible additional DER without violating voltage/thermal limits. The simulator is OpenDSS (Electric Power Research Institute, 2024); the surrogate is a ReLU MLP trained on tens of thousands of offline OpenDSS runs (Dubey and Santoso, 2017; Bollen and Häger, 2007). The CEDeris module embeds the trained surrogate into a distribution-level investment MILP via the framework of this paper. The pruning is essential here because the surrogate input space (per-feeder topology) is much larger than the wake case and naive big-M produces solver memory blow-up.

4.3 Process surrogates in CENovelFuels

Novel-fuel pathways (electrolysis, ammonia synthesis, Fischer–Tropsch, e-CH_4, methanation) embed Aspen Plus or DWSIM process simulators into the CEM. Aspen returns conversion efficiencies, by-product yields, and energy intensities as smooth-but-non-linear functions of operating conditions. Treating these as ReLU surrogates and embedding them via the framework is the methodological substrate of CENovelFuels' production-cost modelling. The same pruning argument applies. Henao and Maravelias (2011); Caballero and Grossmann (2008); Cozad et al. (2014) survey the chemical-engineering literature on this approach; our contribution is its integration with the CEM master via the explicit error decomposition (4).

5 Discussion and limitations

5.1 When the framework fails

The framework requires that f^* be approximable by a small ReLU MLP within tolerance ε, and that IBP-tight bounds be sufficiently small that the MILP relaxation is informative. Three failure modes arise.

Discontinuous f^*. Wake models with sharp regime transitions (e.g. when a downwind turbine moves from inside to outside an upwind wake) create discontinuities that ReLU networks struggle with. The framework still works but requires deeper networks ($L \geq 4$) and benefits from CROWN-style tighter bounds rather than IBP (Wang et al., 2018).

High-dimensional inputs. For surrogate inputs of dimension $n \geq 50$ (e.g. feeder hosting-capacity with detailed topology), IBP bounds widen quadratically in n and pruning yield drops. CROWN-IBP and tighter linear-bound propagation methods are then necessary; the production CENovaSage solver supports plug-in replacement of IBP by these alternatives.

Poorly-behaved surrogate training. If the training data does not adequately cover $\mathcal{X}$ (a real risk for high-dimensional inputs), $\hat{f}$ may fit interior data well while having unbounded extrapolation error near the boundary. The framework cannot detect this; uncertainty quantification on $\hat{f}$ (conformal prediction, ensemble disagreement) is the supplementary tool, applied at the data-collection stage rather than the embedding stage.

5.2 Connection to other surrogate frameworks

The framework is structurally close to OMLT (Ceccon et al., 2022), ENTMOOT (Thebelt et al., 2021), and the chemical-engineering ALAMO-style surrogate optimisation programme (Cozad et al., 2014). The differentiation is in the pruning step, the explicit error decomposition (4), and the energy-system applications (wake, hosting, process). For chemical-process applications outside our scope, OMLT and ALAMO are mature alternatives.

5.3 Surrogate vs full physics

The most common pushback on surrogate-based optimisation is that the simulator is the source of truth, and any approximation by a surrogate is a downgrade. Our response: in any operational context where the simulator is too slow to call inside the optimisation loop, the actual choice is between (i) an approximate surrogate inside the loop, or (ii) calling the simulator post-hoc on the optimiser's solution. Option (ii) does not produce a wake-aware optimum; it produces a wake-evaluated check on a wake-blind optimum. The simulator-as-source-of-truth view is right for evaluation but wrong for optimisation. The framework here is the bridge.

6 Conclusion

We have specified a unified framework for embedding non-linear physics into MILP-based energy-system optimisation via ReLU surrogates with IBP-derived big-M bounds and explicit pruning of always-active and always-inactive neurons. The framework is correct (Proposition 2.3), tractable in practice (the case study converges in under 0.01 seconds on a 24-neuron instance, and production deployments scale to hundreds of neurons), and accompanied by an explicit error decomposition (4) that makes the surrogate's accuracy auditable. The framework is the analytical core of the wake-endogenous wind formulation in CENovaSage, the hosting-capacity surrogate in CEDeris, and the process-surrogate substrate in CENovelFuels.

The piece that most deserves further work is the bound-propagation step. IBP is sound and adequate for the energy-system problem dimensionality we encounter today, but tighter

alternatives (CROWN, β-CROWN, Lirpa) will become operationally meaningful as surrogate networks grow. A v2 of this paper will replace the IBP step with CROWN and report the impact on big-M tightness and solver wall time across a richer benchmark set.

Acknowledgements

This work develops methodology first prototyped within the Compounding Energy Ltd technical blueprint, building on the ReLU-MILP embedding literature (Tjeng et al., 2019; Anderson et al., 2020; Fischetti and Jo, 2018; Grimstad and Andersson, 2019) and the bound-propagation literature originating in adversarial-robustness research (Wang et al., 2018). The Lastrucci 2026 framework (Lastrucci et al., 2026) for energy-system surrogates was a particular inspiration for the integration. The case study is reproducible from the open-source Python implementation accompanying this paper at `compoundingenergy.com/papers/wp03`. Comments are welcome to `christopher@compoundingenergy.com`.

References

Ross Anderson, Joey Huchette, Will Ma, Christian Tjandraatmadja, and Juan Pablo Vielma. Strong mixed-integer programming formulations for trained neural networks. *Mathematical Programming*, 183:3–39, 2020. doi: 10.1007/s10107-020-01474-5.

Math H. J. Bollen and Math Häger. Power system harmonics and harmonic resonance. *IEEE Power and Energy Magazine*, 5(3):81–93, 2007. doi: 10.1109/MPAE.2007.365826.

Fani Boukouvala, Ruth Misener, and Christodoulos A. Floudas. Global optimization advances in mixed-integer nonlinear programming, MINLP, and constrained derivative-free optimization, CDFO. *European Journal of Operational Research*, 252(3):701–727, 2016. doi: 10.1016/j.ejor.2015.12.018.

José A. Caballero and Ignacio E. Grossmann. An algorithm for the use of surrogate models in modular flowsheet optimization. *AIChE Journal*, 54(10):2633–2650, 2008. doi: 10.1002/aic.11579.

Francesco Ceccon, Jordan Jalving, Joshua Haddad, Alexander Thebelt, Calvin Tsay, Carl D. Laird, and Ruth Misener. OMLT: Optimization & machine learning toolkit. *Journal of Machine Learning Research*, 23(349):1–8, 2022.

Alison Cozad, Nikolaos V. Sahinidis, and David C. Miller. Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6):2211–2227, 2014. doi: 10.1002/aic.14418.

Anamika Dubey and Surya Santoso. On estimation and sensitivity analysis of distribution circuit's photovoltaic hosting capacity. *IEEE Transactions on Power Systems*, 32(4):2779–2789, 2017. doi: 10.1109/TPWRS.2016.2622286.

Electric Power Research Institute. OpenDSS: Open distribution system simulator. https://www.epri.com/pages/sa/opendss, 2024.

Matteo Fischetti and Jason Jo. Deep neural networks and mixed integer linear optimization. *Constraints*, 23:296–309, 2018. doi: 10.1007/s10601-018-9285-6.

Bjarne Grimstad and Henrik Andersson. ReLU networks as surrogate models in mixed-integer linear programs. *Computers & Chemical Engineering*, 131:106580, 2019. doi: 10.1016/j.compchemeng.2019.106580.

Carlos A. Henao and Christos T. Maravelias. Surrogate-based superstructure optimization framework. *AIChE Journal*, 57(5):1216–1232, 2011. doi: 10.1002/aic.12341.

Niels Otto Jensen. A note on wind generator interaction. *Risø National Laboratory Report*, (Risø-M-2411), 1983.

I. Katic, J. Højstrup, and Niels Otto Jensen. A simple model for cluster efficiency. *European Wind Energy Association Conference and Exhibition*, pages 407–410, 1986.

G. Lastrucci, T. Karia, V. Schulte, D. Bongartz, and A. M. Schweidtmann. Pruning for efficient deterministic global optimization over trained ReLU neural networks. *arXiv preprint arXiv:2603.23299*, 2026.

Amin Niayifar and Fernando Porté-Agel. Analytical modeling of wind farms: A new approach for power prediction. *Energies*, 9(9):741, 2016. doi: 10.3390/en9090741.

Richard J. A. M. Stevens, Dennice F. Gayme, and Charles Meneveau. Effects of turbine spacing on the power output of extended wind-farms. *Wind Energy*, 19(2):359–370, 2016. doi: 10.1002/we.1835.

Alexander Thebelt, Jan Kronqvist, Miten Mistry, Robert M. Lee, Nathan Sudermann-Merx, and Ruth Misener. ENTMOOT: A framework for optimization over ensemble tree models. *Computers & Chemical Engineering*, 151:107343, 2021. doi: 10.1016/j.compchemeng.2021.107343.

Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. *International Conference on Learning Representations (ICLR)*, 2019.

Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

A Notation glossary

Symbol	Meaning
f^*, $\hat{f}$	True non-linear function and ReLU-MLP surrogate
$\mathcal{X}$	Input domain (closed box in $\mathbb{R}^n$)
$\boldsymbol{W}_\ell, \boldsymbol{b}_\ell$	Weight matrix and bias of layer ℓ
$\boldsymbol{z}_\ell, \boldsymbol{a}_\ell$	Pre- and post-activation of layer ℓ
$\delta_{\ell,j}$	Binary indicator for ReLU node (ℓ, j) in big-M encoding
M	Big-M constant in the ReLU MILP encoding
$[\underline{z}_{\ell,j}, \overline{z}_{\ell,j}]$	IBP-derived pre-activation bounds
W^+, W^-	Positive and negative parts of $\boldsymbol{W}$, componentwise

B Reproducibility

The case study reported in Section 3 is reproducible end-to-end from the Python implementation accompanying this paper. The implementation requires Python 3.11+, NumPy, SciPy (with HiGHS-MILP via `milp`), Pandas, and Matplotlib. Total run time is approximately 8 seconds on a single core for the full pipeline. The random seed is fixed at 20 260 629. Stages `data`, `train`, `prune`, `optimise`, `plots` are individually callable.