Forward Uncertainty by Cutting-Plane Surrogate and Control Variate

RTS-GMLC min-load-shed LP and IEEE-14 AC power flow — TERRA Study Report

Ricardo Marquez, Ph.D.
rmarquez123@gmail.com

September 2026

1. Headline

Table 1: Key results.

Quantity	**Value**
E[shed], control variate, census 3000 + 2,048 residual solves	214.109 ± 0.061 MW
narrower 95% interval than plain Monte Carlo at the same solve budget	22.7×
rms error, control variate (QMC residual) vs QMC over the solve, 7,096 solves	0.0068 vs 0.0403 MW
lines certified below 1% of the variance at zero further solves	4 of 14
AC (IEEE-14, non-convex) residual variance ratio at census 100	2,500×

2. The question

The transmission studies lead with forward questions: the expected load shed under uncertain line ratings, which lines the variance comes from, and what changes under a different rating distribution. Plain Monte Carlo answers them correctly at one solve per draw, $(K + 2)\,N$ solves for attribution, and from scratch under every new distribution. This study asks whether a cutting-plane surrogate harvested from solved draws — value plus gradient at each — with a control-variate estimate of the mean, answers the same questions with fewer solves and a certificate, on the convex RTS-GMLC minimum-load-shed LP (73 buses, 120 branches, $K = 14$ rating-sensitive lines at half their nominal ratings, each rating uniform within ±15%), and whether the construction survives a non-affine, non-convex evaluator, the IEEE-14 AC power flow with $K = 8$ uncertain bus loads.

3. Method

Pipeline, each step one function of `plugins.uncertainty`: (1) harvest a census of solved draws into affine pieces (a_j, c_j, x_j); (2) check the surrogate $L(x) = \max_j[a_j + c_j \cdot (x - x_j)]$ against fresh holdout draws — a certified lower bound only when the evaluator declares convexity with subgradients; (3) estimate $E[G] = E[L] + E[G - L]$, the surrogate mean by scrambled-Sobol quasi-Monte Carlo on the surrogate and the residual by a small fresh sample of solves; (4) compare with plain Monte Carlo and with quasi-Monte Carlo over the solve at equal solve budgets against a 300,000-solve reference; (5) attribute the variance from the surrogate at zero further solves, with the evaluator on the same pick-freeze matrices as the check and a derivative-based bound as the certificate; (6) repeat (1)–(3) and (5) on the AC evaluator with two gradient adapters. Every result object carries its solve ledger.

Table 2: Reference and solve ledger.

Item	LP (RTS-GMLC)	AC (IEEE-14)
reference draws	300,000	10,000 per regime
reference $E[G]$ (MW)	214.114 ± 0.0020	72.32 ± 0.27 / 4.41 ± 0.06
sd$[G]$ (MW)	50.41	27.37 / 6.27
$P(G = 0)$	0.000	0.000 / 0.093
solves in this study	596,704	40,701 (0 fallbacks)
wall clock (s)	1,141	917
LP backend / engine	highs / 0.1.0	engine Newton, warm-started

4. Per-step deep dive

4.1 Census harvest. *What*: one LP solve per draw, its value and its gradient with respect to the K ratings from the bound duals, into a set of affine pieces; prefixes of one 3,000-draw census give the census sizes below. *Inputs*: RTS-GMLC at half nominal ratings; K rating-sensitive lines (branch ids 11, 25, 30, 48, 53, 58, 64, 86, 89, 91, 96, 100, 102, 107); draws uniform on $[0.85, 1.15]\times$ nominal, seed 20260906. *Math*: for the LP $G(x) = \min\{\text{shed}\}$ over dispatch with flow bounds $\pm x_l$, the value is convex in x and the bound dual is a subgradient, so each piece is a supporting plane. *Output*: values, slopes (MW per p.u.) and anchors; ledger 3,000 solves. *Verify*: the 5-bus and RTS-GMLC gradient tests assert the plane inequality on fresh draws on both LP backends (`tests/plugins/power_flow/test_min_shed_rating_gradient.py`).

4.2 Holdout certificate. *What*: $L \le G$ on 2,000 fresh draws, and on the 300,000 reference draws. *Math*: a violation is $L - G > 10\times$ the evaluator's tolerance (10^{-4} MW); the check can refute the certificate, never establish it — convexity plus subgradients is what certifies. *Output*: the table below; on the reference draws 0 violations. *Verify*: `test_case_study_replica.py` recomputes the census-100 row from the pinned pieces and holdout with zero solves.

Table 3: Certificate on 2,000 holdout draws per census size (LP).

census	violations	max $L - G$ (MW)	mean $G - L$ (MW)
100	0/2000	2.9e-11	6.137
300	0/2000	7.0e-11	3.142
1,000	0/2000	7.2e-11	1.323
3,000	0/2000	8.7e-11	0.559

4.3 Control variate. *What*: $E[G] = E[L] + E[G - L]$. *Math*: $E[L]$ by 16 scrambled-Sobol replicates on the surrogate, doubled until its term is at most half the residual term; $E[G - L]$ from 2,048 fresh solves drawn as 8 scrambled-Sobol replicates, its standard error the replicate spread; the 95% half-width is $1.96\sqrt{se_{\text{res}}^2 + se_L^2}$ (both terms, one multiplier). Residual draws are independent of the census by construction. *Output*: the table below, with plain Monte Carlo's half-width at the same solve budget (census + residual). *Verify*: the census-100 row is recomputed from the pinned residual draws, values and QMC replicate means in CI.

$$E[G] = E[L] + E[G - L], \qquad \text{hw}_{95} = 1.96\sqrt{se_{\text{res}}^2 + se_L^2}$$

Table 4: Control variate per census size (LP; reference 214.114 ± 0.0020 MW).

census	budget	$E[G]$ ± hw (MW)	hw_L	var. ratio	plain-MC hw	hw ratio	error
100	2,148	214.241 ± 0.208	0.089	60	2.132	10.2	+0.128
300	2,348	214.153 ± 0.132	0.056	161	2.039	15.4	+0.039
1,000	3,048	214.185 ± 0.114	0.046	626	1.790	15.7	+0.071
3,000	5,048	214.109 ± 0.061	0.023	1,844	1.391	22.7	-0.005

4.4 Equal-budget comparison. *What*: rms error against the reference over 5 replicates, at three solve budgets, for the control variate with a plain and with a scrambled-Sobol residual sample, for scrambled-Sobol QMC over the solve, and for plain Monte Carlo. *Math*: each replicate re-draws the residual (or the whole sample) with a fresh seed on the fixed census. *Output*: the table and figure below. *Verify*: `run.py` asserts the QMC-residual control variate's rms is at most QMC's at the largest budget.

Table 5: rms error vs the reference at equal solve budgets (MW, LP).

census + residual	budget	CV (QMC res.)	CV (plain res.)	QMC	plain MC
300 + 1,024	1,324	0.1266	0.1292	0.1100	0.9656
1,000 + 4,096	5,096	0.0246	0.0282	0.0642	0.7381
3,000 + 4,096	7,096	0.0068	0.0277	0.0403	0.7620

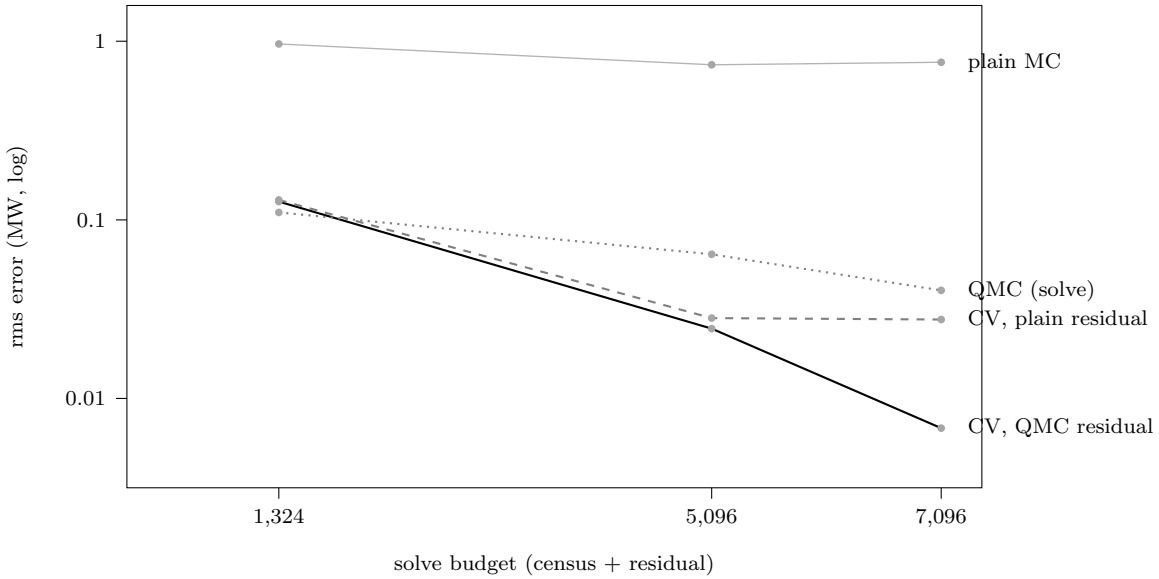


Figure 1: rms error against the 300,000-solve reference at equal solve budgets. The control variate with a quasi-Monte Carlo residual sample compounds the two mechanisms; with a plain residual it is a draw with QMC at the smallest budget and ahead at the larger ones.

4.5 Attribution. *What*: Sobol total indices from the surrogate at zero further solves, checked against the evaluator on the *same* pick-freeze matrices (common random numbers, $n = 500$) so the difference is surrogate error, plus a 1,000-row evaluator reference with bootstrap standard errors,

and the derivative-based bound $S_{T,i} \le (b_i - a_i)^2 \nu_i / (\pi^2 \mathrm{Var}[G])$ with ν_i the mean squared harvested slope. Shapley effects are not run: $K = 14$ exceeds the exact-enumeration cap of 12. *Output*: the tables below; lines certified below 1% of the variance: 30, 89, 91, 107. *Verify*: the 5-bus and additive-toy tests pin the paired comparison, the closed-form total indices and the bound's $12/\pi^2$ tightness on a linear function.

Table 6: Paired Sobol totals, surrogate vs evaluator on common random numbers ($n = 500$).

census	**max** $\lvert\Delta S_T\rvert$	**top-3 set**	**top-3 order**
100	0.0240	3/3	differs
300	0.0111	3/3	differs
1,000	0.0052	3/3	same
3,000	0.0019	3/3	same

Table 7: Total indices per line: evaluator reference (n = 1,000, 15,000 solves), surrogate (census 3,000, 0 solves), DGSM bound.

branch	S_T **reference**	S_T **surrogate**	**DGSM bound**	**certified** $< 1\%$
102	0.229 ± 0.014	0.242	0.289	
58	0.162 ± 0.010	0.161	0.267	
25	0.150 ± 0.010	0.171	0.244	
100	0.131 ± 0.009	0.128	0.277	
48	0.104 ± 0.008	0.103	0.221	
86	0.090 ± 0.006	0.094	0.137	
64	0.086 ± 0.007	0.085	0.160	
96	0.054 ± 0.004	0.053	0.098	
11	0.024 ± 0.001	0.023	0.034	
53	0.021 ± 0.001	0.024	0.030	
91	0.004 ± 0.000	0.005	0.006	yes
30	0.002 ± 0.000	0.002	0.005	yes
89	0.001 ± 0.000	0.001	0.001	yes
107	0.001 ± 0.000	0.001	0.002	yes

4.6 AC evaluator. *What*: the IEEE-14 AC power flow with $K = 8$ load multipliers uniform on $[0.85, 1.15]$, $G = 100\,\mathrm{MVA} \sum_l \max(0, |S_l| - r_l)$, two rating regimes from one set of solves (congested: ratings 0.9× nominal flow, $E[G] = 72.324$ MW; intermittent: 1.05×, $E[G] = 4.410$ MW, atom 0.093 at zero). *Math*: two gradient adapters, both declaring `convex=False`: central differences of G itself (across a kink the average of two slopes, not a subgradient) and the chain rule through the smooth flows with the active set at the centre point (the tangent of G); $2K + 1 = 17$ solves per harvested point, charged to each adapter's ledger though a flow cache shares them. The residual sample is the 2,000-draw holdout. *Output*: the tables below. *Verify*: the census-100 congested row is recomputed from pinned arrays in CI; the kink counterexample is pinned in the adapter tests.

Table 8: AC, congested regime (sd[G] = 27.37 MW): holdout violations (chain rule / FD of G, MW), control variate (chain rule), paired Sobol max $|\Delta S_T|$; charged solves are 17× the census.

census	viol. CR	max CR	viol. FD	max FD	FD/sd	ratio	$E[G] \pm$ hw	$\lvert\Delta S_T\rvert$
100	0	0	0	0	0.000	2,500	72.404 ± 0.025	0.001
300	0	0	14	8.2e-01	0.030	65,822	72.397 ± 0.005	0.000
1,000	0	0	54	8.2e-01	0.030	133,003	72.403 ± 0.004	0.000

Table 9: AC, intermittent regime (sd[G] = 6.27 MW): the same.

census	viol. CR	max CR	viol. FD	max FD	FD/sd	ratio	$E[G] \pm$ hw	$\lvert\Delta S_T\rvert$
100	113	3.7e-03	130	1.4e-01	0.022	93	4.451 ± 0.030	0.051
300	235	3.7e-03	268	4.8e-01	0.076	443	4.457 ± 0.015	0.013
1,000	312	4.6e-03	418	1.1e+00	0.174	2,073	4.454 ± 0.006	0.008

5. Headline in context

The 22.7× narrower interval is against plain Monte Carlo at equal solve budget and carries both Monte Carlo terms. Against quasi-Monte Carlo over the solve, the honest comparator, the control variate with a plain residual is a draw at the smallest budget (0.129 vs 0.110 MW) and ahead by 1.5× at the largest; with a quasi-Monte Carlo residual sample the two mechanisms compound and the advantage is 5.9× at 7,096 solves. The per-sample variance ratios (1,844× on the LP, up to 133,003× on the congested AC regime) measure how affine the consequence is on the box; they are not speed claims. The durable advantages are attribution at zero further solves with a certified exclusion bound, and the certified lower bound on the convex evaluator.

On the AC evaluator the certificate is not claimed. With chain-rule gradients it nevertheless holds on every congested holdout and is violated by at most 0.0046 MW in the intermittent regime (0.07% of sd[G]); finite-differencing the kinked consequence instead produces violations up to 1.09 MW (17%), an artifact of the gradient, not of the physics. The control variate does not care which adapter is used.

6. Limits and known gaps

The upper side of the mean stays statistical: certified upper bounds from function values and convexity were 38% loose (Edmundson–Madansky over 2^{14} vertices) or minutes per query (interpolation) in the spikes and are not shipped. The finite-difference census is 17 solves per point at $K = 8$; at $K \approx 50$–100 it becomes the bottleneck, and an evaluator-supplied gradient is a separate engine-facing follow-up. Shapley effects are capped at $K \leq 12$. The AC reference is 10,000 solves per regime, so the AC control-variate means are compared against a reference an order of magnitude less precise than they are. Re-querying a harvested surrogate under a shifted input distribution costs no solves but its variance reduction under a shift is unmeasured, and a shift outside the census box is uncovered. The evaluator reference for the paired comparison at small census sizes still ranks two close lines in a different order than the surrogate; the totals themselves agree within their standard errors.

7. Reproduction

`PYTHONPATH=src python docs/terra_graph_engine/case_studies/ foundation_power_flow/ rating_uncertainty_surrogate/run.py` regenerates `results.json` (6 significant digits, no wall-clock), `timings.json` and `pinned/`, then checks the plan's numeric targets and exits non-zero if one fails; about 34 minutes with `highspy` installed. Two runs on the same machine and backend give byte-identical outputs. `report_build.py --pdf` renders this report from `results.json`. In CI, `tests/plugins/uncertainty/ test_case_study_replica.py` recomputes the census-100 rows from the pinned arrays with zero solves, re-solves 10 census points on the SciPy backend at 10^{-9}, and asserts targets 1 and 3 from the recomputed values. Expected tail of `run.py`: `all numeric targets met`.

8. Provenance

Table 10: Data and software.

Item	**Value**
RTS-GMLC case	`stochastic_nk_interdiction/references/sundar/RTS_GMLC.m`
network	73 buses, 120 branches, $K = 14$ rating-sensitive lines
IEEE-14 fixture	`benchmarks/pypsa_compare/fixtures/ieee14_ac_pf.json`
engine	terra_graph_engine 0.1.0
LP backend	highs
SciPy / NumPy	1.15.2 / 2.2.4
seeds	20260906 (LP), 20260906 (AC)
plugin	`plugins.uncertainty` v0.1–v0.3 (plan P1–P3)
LP gradient	`power_flow MinLoadShedLp.solve_with_rating_gradient`
report date	September 2026