GridSFM: A Foundation Model for AC Optimal Power Flow

Weiwei Yang[1], Andrea Britto[1], Thiago Spina[1], Spencer Fowers[1], Baosen Zhang[1,2], Chris M. White[1]

[1] Microsoft Research [2] University of Washington

Technical White Paper, May 2026

Abstract

GridSFM is a foundation model for power systems trained on 200 grids and over half a million scenarios. It predicts AC-OPF solutions in milliseconds: given a grid topology and loading conditions, it produces bus voltages, generator dispatch, branch power flows, and a feasibility classification without running a solver, and when higher confidence is needed its predictions serve as warm starts that accelerate conventional solvers. GridSFM is released in two tiers, **GridSFM-Open** (∼15M parameters, for grids up to a few thousand buses, research and prototyping) and **GridSFM-Premier** (∼100M parameters, for production-scale grids up to tens of thousands of buses), both sharing the same architecture and trained on a broad open-data corpus of transmission topologies and operating scenarios spanning feasible and infeasible regimes via a multi-axis perturbation pipeline. Code is available at https://github.com/microsoft/GridSFM, and released checkpoints are hosted in the Hugging Face collection at https://huggingface.co/collections/microsoft/gridsfm.

1 Overview

We are excited to announce the release of **GridSFM**, a family of small foundation models for AC Optimal Power Flow (AC-OPF). Like ResNet pre-trained on ImageNet [1, 2], GridSFM is a single backbone pre-trained on a broad corpus that downstream tasks consume either zero-shot or by fine-tuning, not as a per-grid specialist. It is a physics-informed neural surrogate that infers a full AC-OPF operating point (V, θ, P_g, Q_g, branch flows, and a feasibility verdict) directly from a grid topology and operating scenario (§4). To our knowledge it is the first openly released model in its class trained jointly across ∼200 base transmission topologies under a multi-axis perturbation pipeline (load, generator outage, line de-rating, voltage bounds, cost orderings, and synthetic infeasibility modes; §5). On the held-out test corpus, the model lands in the same standalone-cost accuracy class as DC-OPF, the industry-standard linearized AC-OPF approximation (§6.4), and accelerates the AC-OPF solver by 1.66× as a warm-start seed (§6.3). Out-of-distribution (OOD) evaluation of the GridSFM-Open checkpoint shows it has learned generalizable physical structure that carries over to unseen grids (§7), and fine-tuning with limited data (∼1,000 scenarios) brings performance on a new grid up to in-sample levels (§8). A few-shot ablation further shows that as few as ∼10 fine-tune scenarios already yield reasonable cost and dispatch estimates on a new grid (§9).

2 The Challenge

The foundation of power system operations is solving an optimization problem: determining the optimal generator dispatch that minimizes cost while satisfying thousands of physical and operational constraints. This *AC Optimal Power Flow* (AC-OPF) problem [3, 4, 5] must be solved repeatedly: every 5–15 minutes for real-time dispatch, hourly and daily for electricity markets, and across thousands of contingencies for security assessment.

Letting $\mathcal{G}$ denote the set of generators and $\mathcal{B}$ the set of buses, the mathematical formulation is

$$\min_{V,\theta,P_g,Q_g} \quad \sum_{g\in\mathcal{G}} c_{2g}\,P_g^2 + c_{1g}\,P_g + c_{0g}, \tag{1}$$

$$\text{s.t.} \quad P_i(V,\theta) = \sum_{g\in\mathcal{G}_i} P_g - P_{d,i}, \;\; Q_i(V,\theta) = \sum_{g\in\mathcal{G}_i} Q_g - Q_{d,i} \quad \forall\, i \in \mathcal{B}, \tag{2}$$

$$P_g^{\min} \le P_g \le P_g^{\max}, \quad Q_g^{\min} \le Q_g \le Q_g^{\max} \quad \forall\, g \in \mathcal{G}, \tag{3}$$

$$V_i^{\min} \le V_i \le V_i^{\max} \quad \forall\, i \in \mathcal{B}, \tag{4}$$

$$|S_{ij}(V,\theta)| \le S_{ij}^{\max} \quad \forall\, (i,j) \in \mathcal{E}, \tag{5}$$

$$\theta_{ij}^{\min} \le \theta_i - \theta_j \le \theta_{ij}^{\max} \quad \forall\, (i,j) \in \mathcal{E}, \tag{6}$$

where Eq. (1) is the generation-cost objective; Eq. (2) are the per-bus active and reactive power-balance equations (Kirchhoff's current law, with $P_i(V,\theta)$ and $Q_i(V,\theta)$ the nonlinear AC injections at bus i); Eq. (3) are generator capacity limits; Eq. (4) are bus voltage-magnitude bounds; Eq. (5) are branch thermal limits on the apparent power $|S_{ij}|$; and Eq. (6) are angle-stability limits across each branch.

The power-balance equations $P_i(V,\theta)$, $Q_i(V,\theta)$ are nonlinear and non-convex, making AC-OPF one of the most important non-convex problems solved in industry. Interior-point solvers like IPOPT [6] handle it reliably but require minutes to hours per solve for large grids, especially when the uncertainties in both load and generation need to be accounted for.

Why this matters now. Grid complexity is increasing: renewable generation introduces variability requiring more frequent re-dispatch, distributed resources add decision variables, and the energy transition demands faster planning tools. The computational cost of AC-OPF is becoming a limiting factor. Although solving a single AC-OPF for a particular load can be done, repeatedly solving AC-OPF for thousands of different conditions is not tractable using conventional solvers.

3 GridSFM at a Glance

GridSFM is designed around four core tenets (Fig. 1):

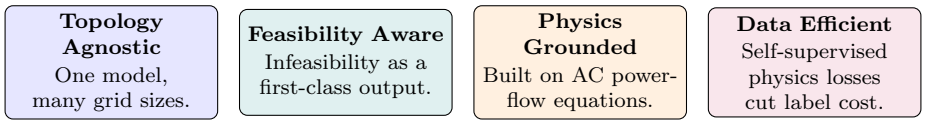

Figure 1: The four core tenets of GridSFM.

- **Topology Agnostic.** GridSFM operates directly on the grid's graph structure (buses as nodes, transmission lines as edges). The same model weights process any grid without topology-specific parameters.

- **Feasibility Aware.** GridSFM classifies every scenario as feasible or infeasible, enabling contingency screening, security assessment, and market-clearing validation.
- **Physics Grounded.** Branch flows come from the AC π-equations on the predicted bus state, and physics penalties regularize training toward feasible operating points.
- **Data Efficient.** Self-supervised physics constraints (power balance, thermal, voltage) supplement supervised solver labels during training, reducing the per-topology label budget needed to fit a useful model.

3.1 Model Tiers

	GridSFM-Open	**GridSFM-Premier**
Parameters	∼15M	∼100M
Grid size	∼4k buses	∼80k buses
License	MIT	TBD

Table 1: GridSFM model tiers. Both share the same architecture; they differ in the scale of supported grids and parameter count.

3.2 What the Model Predicts

Given a grid topology, physical and operating constraints, generation characteristics, and loading scenario, GridSFM produces:

- **Primal variables**: bus voltages V, angles θ, generator dispatch P_g, Q_g, and branch power flows P_{ij}, Q_{ij}: the complete operating point.
- **Feasibility classification**: whether the scenario has a valid AC-OPF solution, with a quantitative confidence score.

4 Architecture

GridSFM is a block-structured discrete neural operator that processes power grids as heterogeneous graphs (Fig. 2). Following discrete exterior calculus (DEC) principles [7], bus quantities (voltage magnitude V, angle θ, nodal injections) are treated as discrete **0-forms** on graph vertices, while branch flows are **1-forms** on oriented edges; the bus–branch incidence acts as the discrete exterior derivative d_0 coupling the two (the angle drop $\theta_i - \theta_j$ across a branch is $d_0\theta$, a 1-form), and KCL appears as its codifferential $\delta = d_0^\top$. This DEC formulation gives the architecture a coordinate-free, topology-agnostic view of power flow that transfers across grids of different size and connectivity.

Type-aware projection embeds the heterogeneous node and edge features into a shared latent space, augmented with a topology-conditioned learned positional encoding. The latent representation is refined by a stack of N blocks, each applying three sub-ops in sequence with residual skips: a per-type global mixer, a topology-aware mixer along the grid's edges, and a per-type MLP. The global mixer is the step where every node attends to every other node of its type at once; the edge-based mixer is the step where node types interact and where the grid's topology enters. Five prediction heads on the resulting latent representation produce bus voltage magnitude V, voltage angle θ, generator active dispatch P_g, generator reactive dispatch Q_g, and a

feasibility head emitting a binary verdict with a continuous margin. Branch active and reactive flows (P_{ij}, Q_{ij}) are not predicted directly: they are computed analytically from the predicted bus state via the standard π-equivalent branch model with off-nominal tap ratio, following the PowerModels.jl polar-AC formulation [8]:

$$P_{ij} = \frac{g}{\tau^2} V_i^2 \; - \; \frac{V_i V_j}{\tau} \big[\, g \cos(\theta_{ij} - \phi) + b \sin(\theta_{ij} - \phi) \,\big], \tag{7}$$

$$Q_{ij} = -\frac{b + b_{fr}}{\tau^2} V_i^2 \; + \; \frac{V_i V_j}{\tau} \big[\, b \cos(\theta_{ij} - \phi) - g \sin(\theta_{ij} - \phi) \,\big], \tag{8}$$

where $\theta_{ij} = \theta_i - \theta_j$, $g + jb \;=\; 1/(r + jx)$ is the series admittance, b_{fr} is the from-side shunt susceptance (half the total line charging in the symmetric pglib convention), and (τ, ϕ) are the transformer tap magnitude and phase shift ($\tau = 1$, $\phi = 0$ for transmission lines). Reverse flows (P_{ji}, Q_{ji}) are computed by the symmetric to-side relations. We follow PowerModels.jl in setting the from/to-side shunt conductances to zero, a standard transmission-network simplification. Training combines supervised regression on solver labels with self-supervised physics constraints (power balance, thermal limits, voltage bounds) and a feasibility classification loss, with an adaptive multi-task wrapper balancing the loss channels.

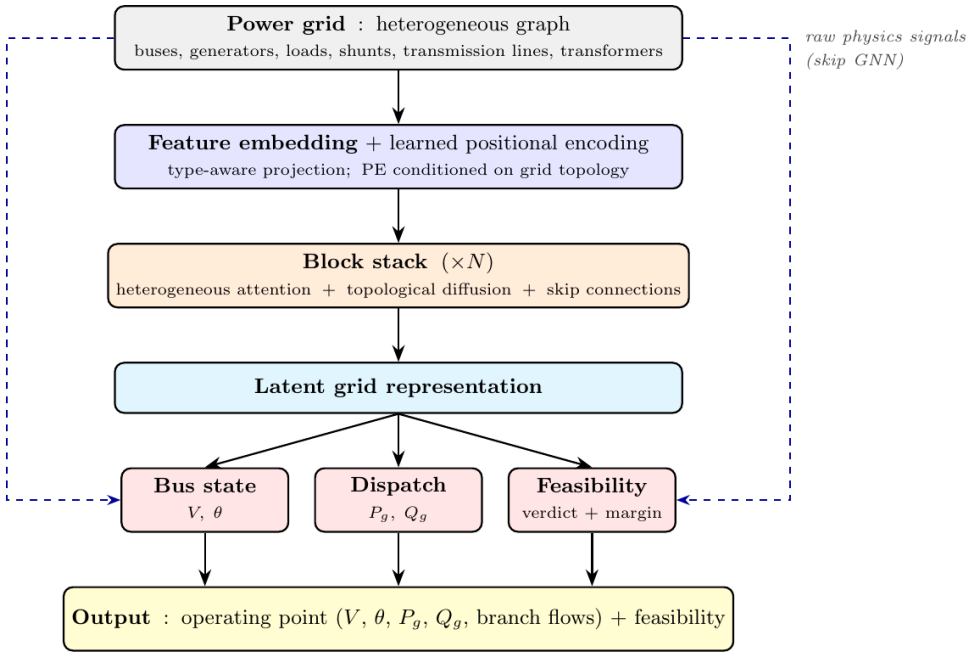

Figure 2: GridSFM architecture. A heterogeneous-graph embedding with learned positional encoding feeds N blocks of attention, topological diffusion, and skips. Five heads predict V, θ, P_g, Q_g, and feasibility; branch flows (P_{ij}, Q_{ij}) are derived analytically from the bus state (Eqs. 7–8).

5 Training Data

GridSFM is trained on a diverse collection of transmission-grid topologies drawn from three open sources: PGLib-OPF [9], the IEEE PES benchmark library covering realistic networks from hundreds to tens of thousands of buses; OPFData [10], a large-scale AC-OPF dataset derived from PGLib-OPF; and the `msr_*` corpus, a 48-state continental US transmission topology set with additional multi-region interconnections, assembled via a companion open-data pipeline (OpenStreetMap infrastructure, EIA generation/demand) and released as part of this work [11].

Perturbation pipeline. Each base topology is expanded into a family of operating scenarios via a perturbation pipeline that varies, independently and in combination, in five axes. Load profiles are sampled with 0.8×–1.5× nominal global scaling and ±10% per-load jitter on top, so that per-load active and reactive demand spans both lightly-loaded and stressed conditions; per-bus weighting captures spatial load heterogeneity, and a separate high-load (`api`) variant samples from 1.1×–1.3× to cover the upper-stress tail. Generator availability is sampled with 30% probability per scenario (conditional on outage, 70% single, 20% double, 10% triple element), with outage selection weighted by $P^{\max}$. Line ratings are sampled with 20% probability per scenario, with 10% of branches de-rated to 70%–95% of nominal rating, covering reduced-capacity equipment, weather-driven derates, and transient transfer-capability constraints. Voltage limits are sampled with 15% probability per scenario, with 10% of buses tightened on $V^{\min}$ and/or $V^{\max}$, covering reactive-support stress and voltage-collapse-adjacent regimes. Generator merit order shuffles cost coefficients on 40% of generators per scenario, so the model learns dispatch from physics and cost structure rather than from per-grid generator orderings. Overall, training spans about 200 grids and over half a million scenarios.

In addition, a synthetic-infeasible pipeline generates targeted failure-mode samples by applying calibrated perturbations to feasible base points across four constraint-violating modes: voltage squeeze, thermal bottleneck, angle tightening, and DC-thermal congestion (verified by a DC-OPF LP). The synth budget is set at a fixed fraction of the per-chunk feasible count and is distributed across the four modes so that no single mode dominates the infeasibility population. For load-scaling-style synth perturbations, the scale is ratcheted up until the AC solver fails to converge, so synth-infeasible scenarios sit clearly outside the feasible distribution rather than near its boundary. Combined with solver-failed natural infeasibles, the per-epoch training mix is approximately 50% feasible and 50% infeasible. Infeasibility is a first-class training signal and a model output (§6.1) rather than an artifact to be discarded.

Feasible scenarios are solved with the AC-OPF solver (IPOPT in PowerModels.jl [8, 6]) to produce labeled ground-truth solutions; infeasible scenarios are labeled by solver non-convergence or by the synthetic generator. The GridSFM-Open training corpus spans a wide range of transmission topologies and large numbers of feasible and infeasible operating scenarios.

Compared with prior data pipelines. OPFData [10] produces a fixed scenario set obtained primarily via per-load scaling in [0.8×, 1.2×] of nominal demand, a relatively narrow band of operating conditions clustered around the base case. The Linux Foundation Energy / IBM Research `gridfm-datakit` pipeline [12] overlaps substantially with what we do: global load scaling up to 4× nominal with per-bus noise, generator and line outages including exhaustive and randomized N-k contingencies, generator cost permutation, and branch admittance (R, X) scaling. GridSFM differs primarily in the explicit voltage-limit perturbation axis and the curated synthetic-infeasibility modes used to drive a balanced feasible/infeasible training mix for the feasibility head.

6 Results

6.1 Feasibility Detection

Feasibility classification is a first-class output of GridSFM. In planning and screening workflows the most important question is rarely *what is the optimal dispatch?* but *given a large number of scenarios, for which of these scenarios would the solver converge?* A fast, accurate feasibility filter lets a planner enumerate large scenario sets (contingencies, demand growth, candidate

siting locations, alternative line builds) and route only the borderline cases to a full AC-OPF solver. The held-out test set we evaluate on combines solver-labeled feasibles, solver-labeled infeasibles, and four synthetic infeasibility modes generated by applying calibrated perturbations to feasible base points (voltage squeeze, thermal bottleneck, angle tightening, and DC-thermal congestion); detection rates are reported as per-class recall so the natural class imbalance does not bias the headline numbers. Table 2 reports the pooled per-mode detection rates and Figure 3 shows the per-grid accuracy distribution broken out by class (real-feas, real-infeas, and the four synthetic modes pooled), which overlap heavily and indicate broadly uniform quality across grids with a small failing tail on the same structurally-hard cases. Pooled across all 54 grids and excluding the broken capacity-aware-spike synthetic mode (§10), the classifier reaches **95.3**% balanced accuracy (mean of TPR and TNR) with $\mathbf{F1 = 0.945}$ on the feasible class at the natural threshold (logit = 0); per-grid balanced accuracy has median 95.0% (range 86.5%–98.3%), per-grid F1 has median 0.945 (range 0.768–0.981), and the per-case values are listed alongside the prediction-accuracy metrics in Table 3.

6.2 Prediction Accuracy on Feasible Scenarios

For feasible scenarios, GridSFM produces a complete operating point: bus voltage magnitude V, voltage angle θ, and generator dispatch P_g, Q_g, with branch flows (P_{ij}, Q_{ij}) derived analytically from the bus state via the π-equivalent branch model (Eqs. 7–8). On the 54-grid test corpus, cost MAPE is 3.35%, V MAE 0.0080 p.u., θ MAE 2.14°, P_g MAE 0.092 p.u., and Q_g MAE 0.129 p.u.; the largest per-grid cost MAPE is 9.97% on `case1803_snem`, with 51 of the 54 grids landing below 5% and a median of 2.85%. Per-grid breakdown is reported in Table 3.

Beyond MAE. MAE and MAPE each hide a different failure mode. MAE is small whenever the underlying quantity has a narrow natural range: bus voltage magnitude V in p.u. only varies between ∼0.95 and ∼1.10, so a flat predictor of $V = 1.0$ on every bus would already post a tiny MAE (a few 10^{-2} p.u.) while carrying no information about the per-scenario voltage profile. MAPE, on the other hand, is dominated by the small-denominator tail: per-generator P_g ranges from small peakers near zero to large baseload units, so a small absolute miss on a small generator becomes a huge percentage error regardless of how well the bulk dispatch is being tracked. Neither metric tells you whether the model is actually following the solver's variation or just collapsing toward a constant. To diagnose that, we run a per-channel pooled linear regression of predicted vs solver-true values on all feasible scenarios in the test split.

Failure mode / population	N (test)	Detection
Real labeled scenarios		
Real feasibles	11,035	96.0%
Real infeasibles	11,118	96.3%
Synthetic infeasibility modes		
Voltage squeeze	1,140	99.7%
Thermal bottleneck	622	77.5%
Angle tightening	849	88.0%
DC-thermal congestion	821	85.4%

Table 2: Feasibility classifier detection rates of the released GridSFM-Open checkpoint, pooled across the GridSFM-Open test split. Real-feasible recall is the fraction of solver-labeled feasibles correctly predicted as feasible; real-infeasible recall is the fraction of solver-labeled infeasibles correctly flagged. The four synthetic infeasibility modes (voltage squeeze, thermal bottleneck, angle tightening, DC-thermal congestion) are detected at 77–99%.

Case	Buses	Cost MAPE (%)	V MAE (p.u.)	θ MAE (deg)	P_g MAE (p.u.)	Q_g MAE (p.u.)	Bal. Acc (%)	F1 (feas)
Aggregate (54-grid test corpus)	–	**3.35**	**0.0080**	**2.14**	**0.092**	**0.129**	**95.3**	**0.945**
case500_goc	500	5.31	0.0086	2.98	0.098	0.137	91.2	0.899
msr_mississippi	528	4.05	0.0066	2.22	0.101	0.092	96.4	0.953
msr_louisiana	571	3.64	0.0048	0.68	0.105	0.135	94.8	0.955
case588_sdet	588	3.84	0.0146	3.71	0.267	0.264	88.5	0.888
msr_south_carolina	588	3.55	0.0032	0.58	0.052	0.070	97.6	0.978
msr_tennessee	603	2.58	0.0051	1.88	0.074	0.125	94.2	0.930
msr_michigan	607	2.76	0.0065	0.99	0.074	0.130	95.0	0.933
msr_new_york	626	7.85	0.0092	3.68	0.084	0.128	92.4	0.963
msr_new_england	640	4.04	0.0029	0.77	0.045	0.070	88.3	0.963
msr_colorado	651	4.96	0.0067	3.13	0.070	0.118	94.4	0.934
msr_kansas	653	2.38	0.0053	0.50	0.064	0.088	94.7	0.944
msr_virginia	661	4.07	0.0126	0.96	0.115	0.165	96.2	0.955
msr_kentucky	701	3.08	0.0067	2.11	0.090	0.146	97.1	0.981
msr_north_carolina	704	4.84	0.0090	0.64	0.069	0.096	96.7	0.973
msr_minnesota	718	3.91	0.0055	0.72	0.043	0.062	97.4	0.980
msr_washington	724	2.71	0.0031	0.23	0.052	0.071	88.8	0.965
msr_arizona	730	3.44	0.0030	0.24	0.045	0.082	95.4	0.963
msr_california	769	3.75	0.0115	3.76	0.083	0.151	94.2	0.882
msr_illinois	770	4.53	0.0355	2.93	0.120	0.269	91.2	0.939
msr_wisconsin	776	4.28	0.0080	0.95	0.066	0.109	96.7	0.973
msr_arkansas	778	4.42	0.0110	1.67	0.091	0.134	96.1	0.954
msr_ohio	784	3.95	0.0075	2.63	0.108	0.150	98.2	0.979
case793_goc	793	2.82	0.0154	2.79	0.228	0.268	92.5	0.913
msr_iowa	832	4.46	0.0080	0.71	0.054	0.068	97.7	0.978
msr_oklahoma	906	2.28	0.0040	0.77	0.048	0.062	96.6	0.979
msr_pennsylvania	910	1.98	0.0056	2.64	0.095	0.136	94.9	0.924
msr_missouri	1033	4.29	0.0054	0.62	0.059	0.083	97.1	0.980
msr_pacific_nw	1106	2.11	0.0020	0.06	0.044	0.041	86.5	0.956
msr_georgia	1240	3.66	0.0065	0.51	0.080	0.114	96.5	0.958
msr_indiana	1271	2.58	0.0036	0.46	0.049	0.064	97.1	0.976
msr_desert_sw	1282	2.07	0.0054	1.41	0.057	0.151	95.7	0.973
case1354_pegase	1354	2.14	0.0104	3.45	0.263	0.401	89.2	0.858
msr_florida	1773	2.50	0.0066	2.68	0.078	0.119	97.9	0.972
case1803_snem	1803	9.97	0.0158	19.84	0.190	0.218	95.5	0.929
case1888_rte	1888	2.14	0.0072	4.82	0.232	0.155	89.8	0.824
case1951_rte	1951	1.35	0.0060	3.10	0.068	0.143	97.3	0.966
case2000_goc	2000	2.27	0.0101	2.89	0.082	0.115	98.3	0.980
case2312_goc	2312	1.96	0.0223	2.89	0.262	0.325	96.7	0.952
case2383wp_k	2383	3.52	0.0099	2.53	0.104	0.097	95.6	0.900
case2736sp_k	2736	1.68	0.0058	1.27	0.063	0.055	97.2	0.939
case2737sop_k	2737	1.99	0.0078	0.92	0.053	0.056	91.9	0.897
case2742_goc	2742	2.03	0.0059	1.89	0.052	0.076	98.1	0.978
case2746wop_k	2746	3.31	0.0059	1.50	0.046	0.049	87.1	0.768
case2746wp_k	2746	1.78	0.0052	1.58	0.056	0.050	89.4	0.835
Texas2k_series25_case1_summerpeak	2751	2.97	0.0084	2.06	0.068	0.098	96.1	0.946
case2848_rte	2848	1.46	0.0090	1.62	0.046	0.081	91.1	0.894
case2853_sdet	2853	2.99	0.0131	2.82	0.134	0.294	92.4	0.884
case2868_rte	2868	1.47	0.0065	2.94	0.052	0.105	93.4	0.908
case2869_pegase	2869	1.78	0.0066	10.56	0.223	0.288	95.0	0.932
case3012wp_k	3012	1.66	0.0063	1.57	0.077	0.072	92.6	0.882
case3022_goc	3022	2.62	0.0142	15.98	0.460	0.491	93.2	0.890
case3120sp_k	3120	1.40	0.0072	1.01	0.080	0.066	96.1	0.932
case3375wp_k	3374	0.72	0.0088	2.15	0.080	0.176	91.1	0.783
msr_texas	3889	2.87	0.0061	3.44	0.080	0.149	88.1	0.859

Table 3: Per-grid prediction accuracy and feasibility metrics for the released GridSFM-Open checkpoint on the 54-grid test corpus. Channels are pooled mean absolute errors. *Bal. Acc* (mean of TPR and TNR) and *F1* are computed at the natural decision threshold (logit > 0) with the broken capacity-aware-spike synthetic mode excluded.

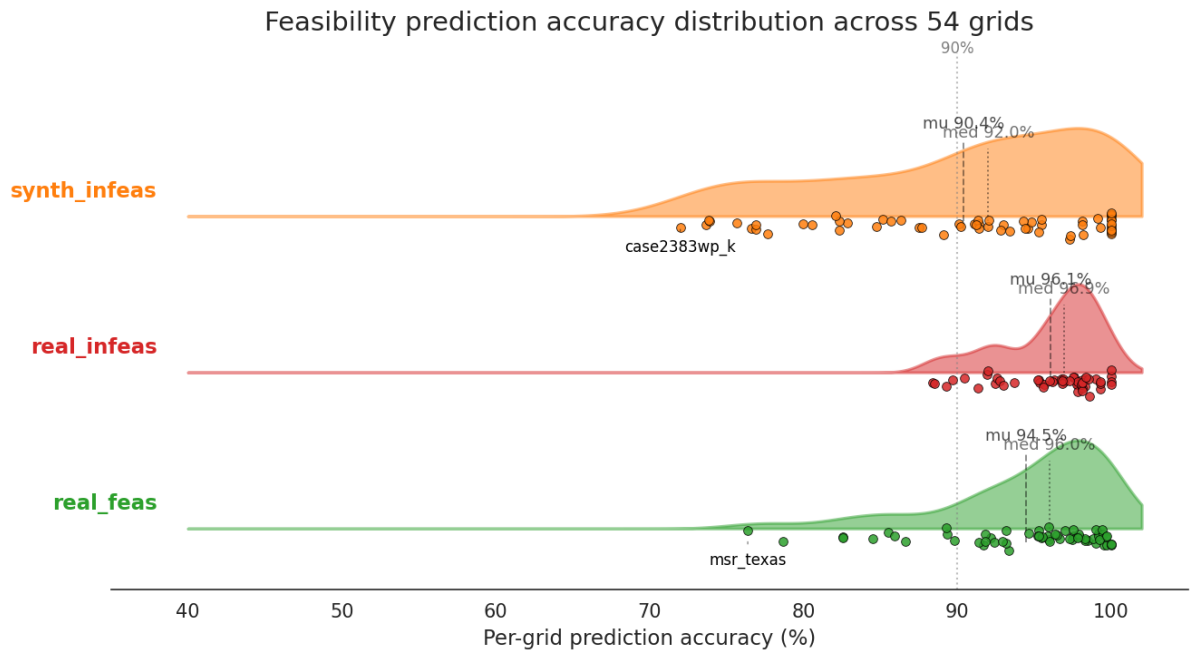

Figure 3: Per-grid feasibility prediction accuracy of the released GridSFM-Open checkpoint, broken out by class: real feasibles (green), real infeasibles (red), and synthetic constraint-violating modes pooled (orange; voltage squeeze, thermal bottleneck, angle tightening, DC-thermal congestion). Filled KDE plus per-grid dots, with class-mean (μ) and median lines.

The *slope* measures whether the model captures the right amplitude: a slope of 1 tracks the solver's spread one-for-one, while a slope below 1 indicates systematic under-prediction of the spread (the model collapsing toward the mean). The *Pearson correlation* R, and its square R^2, measures whether the model captures the right ordering: a high R means scenarios the solver maps to higher values are also predicted higher by the model.

Table 4 reports the per-channel numbers. Cost, P_g, and θ are essentially perfect on all three metrics (slope ≈ 1, $R^2 \geq 0.88$): the model tracks both the amplitude and the ordering of the solver's solution on these channels. V has recovered substantially compared with earlier checkpoints (slope $= 0.87$, $R^2 = 0.89$), so the constant-prediction failure mode the small-MAE caveat above warns about is no longer the dominant story; voltage spread is now mostly being captured. Q_g is the remaining weak channel (slope $= 0.75$, $R^2 = 0.74$): the predicted reactive spread covers about three-quarters of the solver's, with mild amplitude collapse and a meaningful share of variance unexplained.

Channel	Slope	R	R^2
θ (bus angle, rad)	1.01	0.94	0.88
V (bus voltage magnitude, p.u.)	0.87	0.94	0.89
P_g (generator active power)	0.99	0.99	0.98
Q_g (generator reactive power)	0.75	0.86	0.74
Per-graph cost ($)	0.98	1.00	0.99

Table 4: Per-channel pooled regression of predicted vs solver-true values for the released GridSFM-Open checkpoint on its test corpus. Slope $= 1$, $R = 1$ is perfect prediction.

6.3 Warm-Start Acceleration

For situations that need an exact AC-OPF answer, GridSFM's prediction is exported as a warm-start seed for the PowerModels.jl [8] AC-OPF solver and used to accelerate the solve rather than replace it. The AC-OPF formulation in PowerModels.jl is handled by an interior-point algorithm (IPOPT [6]) that starts from an initial guess of the operating point and iteratively refines it until the optimality conditions are satisfied. The number of refinement iterations depends directly on how close the starting guess is to the optimum: a poor starting point can require dozens of iterations, while a near-optimal one converges in a handful. A cold solve uses a generic, typically far-from-optimal default ($V = 1.0$, $\theta = 0$, P_g at the midpoint of its capacity envelope), and a warm-start replaces that default with a closer estimate so the solver reaches the same tolerance in fewer iterations and less wall time.

The standard fast approximation in industry is the DC-OPF linearization [13, 5], a linear relaxation of AC-OPF that fixes voltage magnitudes at $V_i \equiv 1.0$, drops reactive power, assumes lossless branches, and reduces the power-flow equations to a sparse linear system in θ and P_g. The resulting LP is fast and produces a θ, P_g pair that can seed an AC warm-start, so the reference baseline in this section is not only cold AC but also DC-warmed AC.

For each grid in our test corpus we evaluate 10 scenarios from the held-out test split, running three AC-OPF solves through PowerModels.jl on each (case, scenario) pair: cold (default initialization), DC-warm (seed with the DC-OPF solution), and GridSFM-warm (seed with the model's forward-pass prediction). We also include a fourth, idealized comparison point that we call *ground-truth warm-start* (GT-warm): we seed the solver with the converged AC-OPF solution itself, taken from a prior tight solve on the same scenario. By definition no warm-start strategy can seed with a starting point closer to the optimum than the optimum itself, so GT-warm is the practical ceiling on how much speedup any warm-start can deliver. Both approximate warm-start modes (DC and GridSFM) use a wide-barrier config (`mu_init`= 1.0, `warm_start_bound_push`= 1.0), since at narrower `mu_init` the barrier function is too steep at the start of the warm-start solve, the algorithm takes vanishing steps from any non-central-path starting point, and the approximate warm-start loses to cold on every grid. All wall times are the solver's reported `solve_time`, with the DC-warm time including the DC-OPF preprocess. We aggregate per-grid speedup ratios by the *geometric mean*, defined for n per-grid speedup ratios $r_1, \ldots, r_n$ (with $r_i = t_{\text{cold,i}}/t_{\text{warm,i}}$) as

$$\text{GeoMean}(r_1, \ldots, r_n) \;=\; \left(\prod_{i=1}^{n} r_i\right)^{1/n}. \tag{9}$$

This is the right aggregate for multiplicative ratios for two reasons. First, ratios live on a log-scale: a 2× speedup on grid A and a 0.5× slowdown on grid B (their product is 1) should average to 1× "no net change", which the geometric mean gives; the arithmetic mean of {2, 0.5} would falsely report 1.25×. Second, the geometric mean is invariant under inverting the ratio: reporting cold-time-divided-by-warm-time gives the reciprocal of warm-time-divided-by-cold-time, as expected for a multiplicative summary. (The arithmetic mean does not have this property.)

On the release corpus, GridSFM warm-start achieves **1.66**× geometric-mean wall speedup vs cold (winning on 41/54 grids) and delivers **1.59**× the speedup of DC-warm alone (beating DC head-to-head on 39/54 grids). The GT ceiling is 2.72×, so GridSFM is currently capturing ∼61% of the cold-to-GT headroom. DC-warm by itself is essentially tied with cold (1.04× geometric mean) because its linearization helps on lightly-loaded radial grids and hurts on meshed transmission ones, leaving the population roughly cancelled. Figure 4 shows the per-grid speedup distribution across the three approximate methods plus the GT ceiling; per-grid timings and speedup ratios are reported in Table 5.

Table 5: Per-grid AC-OPF wall-clock timing and warm-start speedup over cold AC-OPF, paired in a single Julia run (54-way parallel, one core per case, BLAS pinned to 1 thread). *DC-warm* and *Model-warm* are AC-OPF solve times when seeded by DC-OPF and the GridSFM partial-warm payload (P, θ) respectively; speedup ratios use the same per-case cold time, with bolded entries marking grids where the seed beat cold ($\geq 1\times$). Aggregate (geomean across 54 grids): DC-warm 1.04×, Model-warm 1.66×, GT-ceiling oracle 2.72×.

Case	Buses	Cold (s)	DC-warm (s)	Model-warm (s)	DC ×	Model ×
Aggregate (54-grid geomean)	–	–	–	–	**1.04×**	**1.66×**
case500_goc	500	2.38	2.39	3.76	1.00×	0.63×
msr_mississippi	528	4.19	3.90	6.98	**1.07×**	0.60×
msr_louisiana	571	3.62	3.94	6.50	0.92×	0.56×
case588_sdet	588	2.43	3.04	5.02	0.80×	0.48×
msr_south_carolina	588	7.36	6.89	8.39	**1.07×**	0.88×
msr_tennessee	603	6.57	4.72	7.12	**1.39×**	0.92×
msr_michigan	607	19.40	10.03	6.78	**1.93×**	**2.86×**
msr_new_york	626	14.37	9.53	8.45	**1.51×**	**1.70×**
msr_new_england	640	4.35	4.34	6.63	**1.00×**	0.66×
msr_colorado	651	6.36	6.37	7.56	1.00×	0.84×
msr_kansas	653	4.02	4.33	6.05	0.93×	0.66×
msr_virginia	661	7.28	8.01	7.26	0.91×	**1.00×**
msr_kentucky	701	5.71	6.22	7.35	0.92×	0.78×
msr_north_carolina	704	12.73	12.11	7.82	**1.05×**	**1.63×**
msr_minnesota	718	9.28	8.85	8.24	**1.05×**	**1.13×**
msr_washington	724	4.18	4.39	7.13	0.95×	0.59×
msr_arizona	730	7.45	5.38	6.93	**1.38×**	**1.08×**
msr_california	769	14.34	14.17	7.30	**1.01×**	**1.96×**
msr_illinois	770	14.29	11.12	9.30	**1.29×**	**1.54×**
msr_wisconsin	776	6.16	6.34	7.02	0.97×	0.88×
msr_arkansas	778	5.96	5.99	7.50	1.00×	0.80×
msr_ohio	784	14.13	16.87	6.62	0.84×	**2.14×**
case793_goc	793	6.54	4.35	5.37	**1.50×**	**1.22×**
msr_iowa	832	9.34	8.96	7.96	**1.04×**	**1.17×**
msr_oklahoma	906	9.63	9.50	8.98	**1.01×**	**1.07×**
msr_pennsylvania	910	16.51	18.45	6.60	0.89×	**2.50×**
msr_missouri	1033	14.83	14.48	8.02	**1.02×**	**1.85×**
msr_pacific_nw	1106	15.47	16.31	8.52	0.95×	**1.81×**
msr_georgia	1240	15.11	14.37	8.37	**1.05×**	**1.81×**
msr_indiana	1271	18.14	18.00	8.66	**1.01×**	**2.09×**
msr_desert_sw	1282	19.19	22.33	9.31	0.86×	**2.06×**
case1354_pegase	1354	14.46	11.74	7.65	**1.23×**	**1.89×**
msr_florida	1773	32.28	28.46	11.05	**1.13×**	**2.92×**
case1803_snem	1803	19.45	84.59	9.27	0.23×	**2.10×**
case1888_rte	1888	33.15	28.74	11.57	**1.15×**	**2.87×**
case1951_rte	1951	43.92	28.03	10.93	**1.57×**	**4.02×**
case2000_goc	2000	23.66	23.21	9.27	**1.02×**	**2.55×**
case2312_goc	2312	22.18	21.48	10.37	**1.03×**	**2.14×**
case2383wp_k	2383	29.70	24.76	10.54	**1.20×**	**2.82×**
case2736sp_k	2736	31.10	28.18	12.26	**1.10×**	**2.54×**
case2737sop_k	2737	32.76	25.85	12.79	**1.27×**	**2.56×**
case2742_goc	2742	69.09	78.47	9.31	0.88×	**7.42×**
case2746wop_k	2746	31.22	30.76	14.12	**1.02×**	**2.21×**
case2746wp_k	2746	30.15	29.63	12.29	**1.02×**	**2.45×**
Texas2k_series25_case1_summerpeak	2751	74.51	27.79	10.90	**2.68×**	**6.83×**
case2848_rte	2848	47.38	38.71	12.24	**1.22×**	**3.87×**
case2853_sdet	2853	29.74	39.15	18.21	0.76×	**1.63×**

Continued on next page

Table 5 – *Continued from previous page*

Case	Buses	Cold (s)	DC-warm (s)	Model-warm (s)	DC ×	Model ×
case2868_rte	2868	58.60	47.75	14.48	**1.23**×	**4.05**×
case2869_pegase	2869	31.61	45.35	14.49	0.70×	**2.18**×
case3012wp_k	3012	33.29	25.98	14.05	**1.28**×	**2.37**×
case3022_goc	3022	28.08	47.18	14.49	0.60×	**1.94**×
case3120sp_k	3120	33.94	31.41	14.35	**1.08**×	**2.37**×
case3375wp_k	3374	35.64	37.68	15.07	0.95×	**2.37**×
msr_texas	3889	45.82	45.53	21.08	**1.01**×	**2.17**×

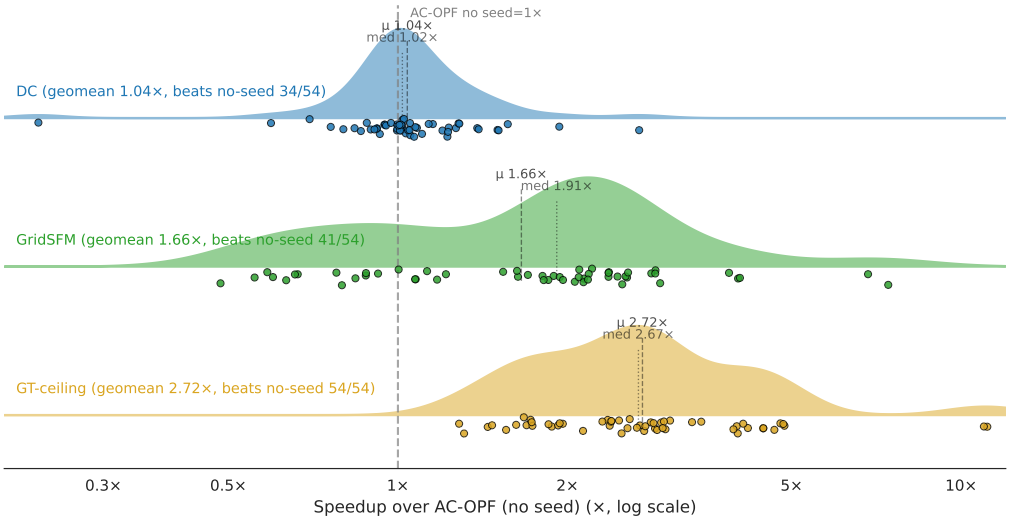

Figure 4: Per-grid speedup distribution over cold AC-OPF for the released GridSFM-Open checkpoint (log-x axis): GridSFM warm-start (green, geomean 1.66×), DC warm-start (blue, 1.04×), and ground-truth warm-start (gold dashed, 2.72×, the practical ceiling). Filled KDE plus per-grid dots; light dashed and dotted lines mark each method's geometric mean and median.

6.4 Cost-Prediction Accuracy: GridSFM vs DC-OPF

We also profiled the standalone cost-prediction accuracy of GridSFM against DC-OPF on the same scenarios used for the warm-start benchmark, with the AC-OPF cold-solve cost as ground truth. Unlike the speedup ratios in §6.3, cost gaps are absolute percentage errors ($\text{gap}_i = |c_{\text{pred},i} - c_{\text{true},i}|/c_{\text{true},i} \cdot 100\%$) rather than multiplicative ratios, so we aggregate them by the *arithmetic mean* (the standard MAPE convention used in §6.2) rather than the geometric mean. The arithmetic-mean cost gap is **2.80**% for DC vs 3.41% for GridSFM (medians **1.81**% vs 2.23%): on standalone cost prediction, DC-OPF is slightly more accurate than GridSFM in aggregate. Figure 5 shows the full per-scenario cost-gap distribution. These aggregate cost-gap numbers should be read as a starting point rather than a ceiling: as a foundation model, GridSFM further specializes via per-grid fine-tuning (§8, §9), and even a small number of labeled scenarios on a target grid is sufficient to drop its standalone cost gap by an order of magnitude, well below the DC reference.

6.5 Comparison against similar models

We compare GridSFM with the closest existing model we have come across, but a direct head-to-head is impossible, since GridSFM is in its own class as a multi-grid generalist trained jointly

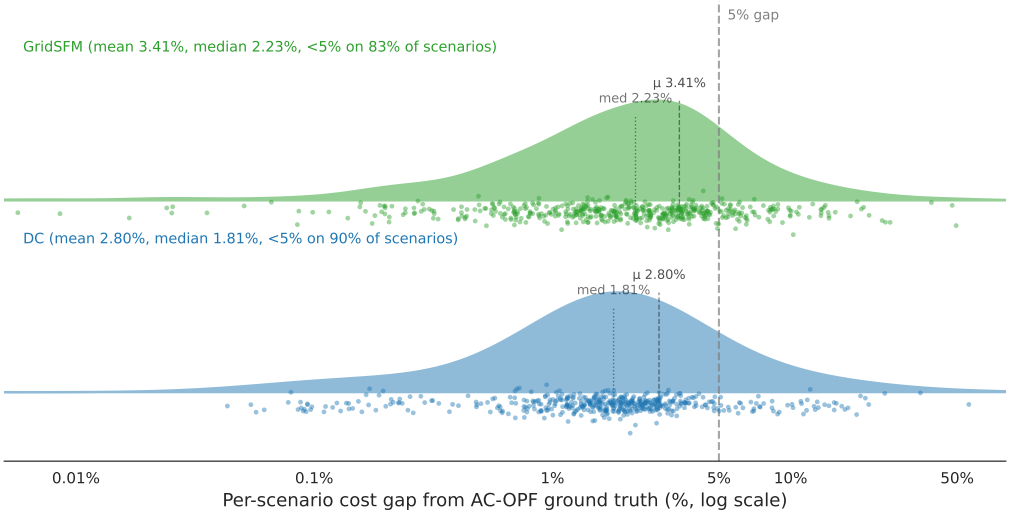

Figure 5: Per-scenario cost gap from AC-OPF ground truth on test dataset, log-x axis: DC-OPF (blue) and GridSFM (green). Filled KDE plus per-scenario dots; dashed lines mark mean (μ) and median; the vertical line marks the 5% gap.

on a corpus of distinct base topologies, while published learned AC-OPF surrogates are almost universally per-grid specialists, with different training scenarios, perturbation envelopes, and held-out test sets. The numbers below should therefore be read as "what accuracy class is each model operating in on this channel?" rather than as strict values.

The Linux Foundation Energy / IBM Research `gridfm-graphkit` [14] is a heterogeneous Transformer-based GNN with bus and generator node types and three message-passing relations (bus–bus along transmission edges, plus bus–generator in both directions). The architecture stacks 12 `HeteroConv` layers, each running a `TransformerConv` per relation with 8 attention heads and hidden dim 48, and at every layer a small MLP consumes per-node active/reactive power-balance residuals computed from the current latent state and feeds a correction back into the embedding. The bus head outputs V and θ per bus; the generator head outputs P_g per generator; Q_g is recovered from the reactive bus balance rather than predicted per generator, which is why the Q_g column in Table 6 is reported at the bus level (PV/REF buses) for `gridfm-graphkit` and per generator for GridSFM. We trained `gridfm-graphkit` from scratch by following the upstream `gridfm-datakit [12] + gridfm_graphkit train` pipeline on `case500_goc` and `case2000_goc` (∼30k feasible scenarios per grid, 200 epochs, the upstream `LayeredWeightedPhysics + MaskedBusMSE + MaskedGenMSE` loss configuration), then ran the trained checkpoint through a custom evaluator that emits MAE in per-unit on the same channels we report for GridSFM.

Grid	Model	Params	V MAPE (%)	θ MAE (deg)	P_g MAE (p.u.)	Q_g MAE (p.u.)	Cost MAPE (%)
`case500_goc`	`gridfm-graphkit`	20.1M	1.53	2.81	0.049	0.518‡	2.67
	GridSFM-Open	15.1M	0.86	2.98	0.098	0.137	5.31
`case2000_goc`	`gridfm-graphkit`	20.1M	2.16	6.72	0.067	0.430‡	3.05
	GridSFM-Open	15.1M	1.01	2.89	0.082	0.115	2.27

Table 6: Each model on the full test split of its own pipeline. ‡ `gridfm-graphkit` reports Q_g aggregated per bus (PV/REF) rather than per generator (the two are equal when there is one generator per bus, but strictly not the same metric).

A note on the θ column: `gridfm-graphkit`'s per-bus θ MAE is inflated by an uncontrolled angle-

gauge drift in their model, per-edge $\Delta\theta_{ij}$ MAE on the same checkpoint is 0.146° (`case500_goc`) / 0.136° (`case2000_goc`), which is the quantity branch flows depend on. Closing the per-bus gap would require their network to learn a scenario-specific transform tying every bus back to the slack reference.

7 Out-of-Distribution Generalization

To probe out-of-distribution (OOD) generalization we evaluate the released GridSFM-Open checkpoint on `OPFData_pglib_opf_case6470_rte`, a 6470-bus network derived from `pglib-opf`. GridSFM-Open was trained on grids of up to 4661 buses (`case4661_sdet`), so `case6470` is roughly 1.4× larger than the largest grid the model has ever seen. Both the full-topology test split (`fulltop`) and the single-line-contingency split (`n1`) are used as held-out probes. Table 7 reports the headline regression and feasibility metrics; Table 8 reports per-channel calibration.

metric	Pre-train in-sample	`case6470 fulltop` (OOD, zero-shot)	`case6470 n1` (OOD, zero-shot)	Change vs in-sample
`cost_mape`	3.35%	13.99%	14.24%	4.2× worse
`feas_f1`	0.945	0.000	0.000	classifier collapses
P_g MAE	0.092	0.257	0.257	2.8×
Q_g MAE	0.129	0.289	0.286	2.2×
V MAE	0.0080	0.0314	0.0314	3.9×
θ MAE	0.0374	0.1727	0.1744	4.6×

Table 7: Zero-shot out-of-distribution evaluation of the released GridSFM-Open checkpoint on `case6470_rte`. *Pre-train in-sample* is the checkpoint's aggregate accuracy on the test splits of the grids it was trained on (Table 3). The single-line-contingency split (`n1`) tracks the intact topology (`fulltop`) within 0.3 pp on cost, so the bottleneck is grid scale and topology, not contingency reasoning.

channel	OOD slope	OOD R	OOD R^2	Pre-train slope (in-sample)	calibration shift
θ	0.843	0.902	0.813	1.007	mild under-prediction
V	0.273	0.333	0.111	0.874	near-collapse
P_g	0.911	0.976	0.952	0.985	minor regression
Q_g	0.573	0.610	0.372	0.747	meaningful regression
cost	1.165	0.999	0.997	0.978	overshoots by 17%

Table 8: Per-channel calibration on the out-of-distribution evaluation pool. Slope = 1 means predictions match true magnitudes; R is the linear correlation with truth; R^2 is the fraction of variance explained. The *Pre-train slope (in-sample)* column is the calibration of the released checkpoint on the test splits of the grids it was trained on (Table 4), for context.

As Tables 7 and 8 show, θ and cost generalize to the new grid (angle slope 1.01 → 0.84, cost $R^2 = 0.997$ with a 17% magnitude overshoot, so the relative cost ordering is essentially perfect); V calibration nearly collapses (slope 0.27, $R^2 = 0.11$); Q_g degrades meaningfully (slope 0.57); and the feasibility classifier collapses to "always infeasible" on both splits ($F_1 = 0$ despite 355/750 scenarios actually being feasible).

8 Fine-Tuning to Operator Topologies

We fine-tuned the released GridSFM-Open checkpoint on `OPFData_case6470_rte__fulltop` for 10 epochs with 1,000 training graphs; the N-1 contingency split (`n1`) is held out entirely from

fine-tuning. Train and validation loss decrease monotonically over the 10 epochs (Figure 6). Held-out test metrics are reported in Table 9 (regression and feasibility) and Table 10 (per-channel calibration). Compared with the zero-shot OOD baseline in Tables 7 and 8, the fine-tuned checkpoint reduces `cost_mape` by about 12×, brings the V slope from near-collapse to near-ideal, and recovers the feasibility classifier from $F_1 = 0$ to $F_1 \approx 0.99$; the held-out `n1` column tracks `fulltop` on every channel (see Table 9), so fine-tuning on the full topology transfers cleanly to the contingency variant. This result demonstrates the foundation-model claim concretely: GridSFM has already absorbed the AC-OPF physics during pre-training, so adapting to an unseen grid only requires re-aligning the model's calibration to that grid's specific scale and operating envelope rather than re-learning the physics from scratch, which is why ∼1,000 scenarios suffice. The same effect should also apply within the in-sample corpus: based on these results, we suspect that further per-grid fine-tuning of the released checkpoint on each grid in Table 3 would dramatically improve its per-case performance by allowing the model to specialize.

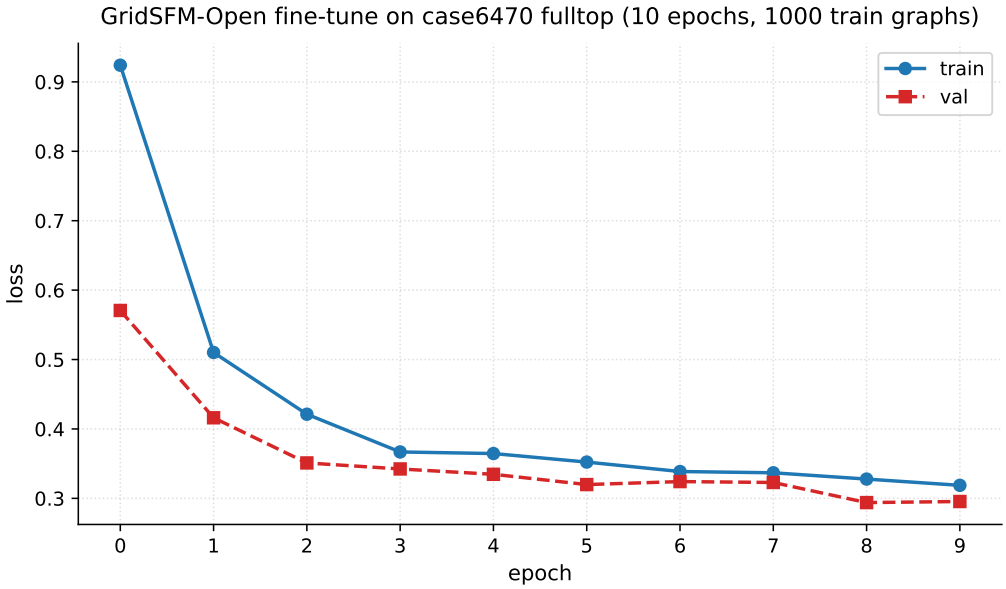

Figure 6: Train (solid) and val (dashed) loss curves over 10 epochs of fine-tuning on `case6470_rte__fulltop` (1,000 train graphs).

metric	Pre-train in-sample	Pre-finetune (OOD)	Post-finetune (`fulltop`)	Post-finetune (`n1`, held out)
`cost_mape`	3.35%	13.99%	**1.12%**	**0.93%**
`feas_f1`	0.945	0.000	**0.988**	**0.997**
P_g MAE	0.092	0.2573	0.0562	0.0507
Q_g MAE	0.129	0.2887	0.0993	0.0957
V MAE	0.0080	0.0314	0.0059	0.0057
θ MAE	0.0374	0.1727	0.0432	0.0447

Table 9: Test evaluation of the released GridSFM-Open checkpoint before and after fine-tuning. *Pre-train in-sample*: aggregate over the in-distribution test corpus reported in Table 3. *Pre-finetune*: zero-shot evaluation of the released checkpoint on `case6470_rte__fulltop` (out-of-distribution; reproduced from Table 7). *Post-finetune* (green text): same checkpoint after a 1,000-scenario fine-tune on the `case6470_rte__fulltop` train split, evaluated on the held-out `fulltop` test split (in-domain after the fine-tune) and on `n1` (still held out from the fine-tune; the model never saw a single N-1 contingency scenario during fine-tuning).

channel	Pre-FT slope	Post-FT slope	Pre-FT R	Post-FT R	Pre-FT R^2	Post-FT R^2
θ	0.843	**0.996**	0.902	**0.956**	0.813	**0.914**
V	0.273	**0.914**	0.333	**0.961**	0.111	**0.924**
P_g	0.911	**0.999**	0.976	**0.996**	0.952	**0.992**
Q_g	0.573	**0.845**	0.610	**0.932**	0.372	**0.868**
cost	1.165	**0.982**	0.999	0.999	0.997	0.998

Table 10: Per-channel calibration on `case6470` before and after fine-tuning. Pre-FT columns are the zero-shot OOD calibration (reproduced from Table 8); Post-FT columns (green text) are the same checkpoint after a 1,000-scenario fine-tune on `case6470_rte__fulltop`.

9 Few-shot Ablation

In §8 we showed that 1,000 fine-tune scenarios are enough to recover baseline accuracy on `case6470`. To map out the data-efficiency frontier, we sweep $n_{\text{train}} \in \{10, 100, 500, 1{,}000\}$ while holding the rest of the recipe in §8 fixed: same released checkpoint, 10 epochs, AdamW at 10^{-4}, the `case6470__fulltop` train and val pools, and `n1` held out throughout. Table 11 reports per-channel regression and feasibility errors on the `fulltop` test split; Table 12 reports per-channel calibration aggregated over `fulltop` and `n1`.

n_{train}	`cost_mape`	`feas_f1`	V MAE	θ MAE	P_g MAE	Q_g MAE
0 (zero-shot)	13.99%	0.000	0.0314	0.1727	0.2573	0.2887
10	1.76%	0.919	0.0184	0.0570	0.1053	0.2050
100	0.88%	0.970	0.0120	0.0483	0.0841	0.1254
500	0.75%	0.996	0.0069	0.0423	0.0623	0.1018
1,000	1.12%	0.988	0.0059	0.0432	0.0562	0.0993

Table 11: Few-shot fine-tune ablation on `case6470_rte`: regression and feasibility errors on the `fulltop` test split as a function of fine-tune training-set size n_{train}. `feas_f1` excludes the broken capacity-aware-spike synthetic mode, matching the convention of §6.1.

	θ		V		P_g		Q_g		cost	
n_{train}	slope	R^2	slope	R^2	slope	R^2	slope	R^2	slope	R^2
0 (zero-shot)	0.843	0.813	0.273	0.111	0.911	0.952	0.573	0.372	1.165	0.997
10	0.900	0.885	0.266	0.289	0.989	0.986	0.639	0.613	1.012	0.997
100	0.938	0.897	0.657	0.701	0.999	0.989	0.801	0.838	0.975	0.997
500	0.978	0.913	0.890	0.905	0.997	0.992	0.851	0.865	0.975	0.998
1,000	0.996	0.914	0.914	0.924	0.999	0.992	0.845	0.868	0.982	0.998

Table 12: Few-shot fine-tune ablation on `case6470_rte`: per-channel calibration (slope and R^2) aggregated over `fulltop` and `n1`, matching the convention of Tables 8 and 10.

The channels reach good prediction at markedly different data budgets. Cost and the values of P_g are essentially recovered with as few as ∼10 fine-tune scenarios; θ and Q_g reach their best operating point with 100–500 scenarios; and V is the slowest channel, with calibration only fully recovering near $n_{\text{train}} = 1{,}000$. The practical implication is to match the fine-tune budget to the downstream application: warm-starting a solver needs at least accurate θ and P_g, whereas rough cost estimation only needs accurate P_g. We also note that OPFData [10] restricts load variation to the narrow [0.8, 1.2] range of nominal demand, which makes the underlying scenario distribution relatively easy to fit and likely understates the data budget that a wider operating envelope would require.

10 Limitations

GridSFM has several known weaknesses that frame the next round of work.

The lead direction is the topology-conditioned positional encoding (§4). The released checkpoint's positional embedding (PE) is only shallowly learned: it has not yet seen enough diverse-topology training signal to fully internalize the relative relationships between grid components. Training a richer PE end-to-end is data-gated: each labeled scenario requires an AC-OPF solve, and solver runs are the dominant cost in our data pipeline. Looking ahead, we plan to leverage the released checkpoint itself, both as a warm-start seed for the solver (§6.3) and as a stress score for active-learning scenario selection, to drive down labeling cost.

A second limitation is the synthetic infeasibility coverage in pre-training. Most of our current synthetic modes (§5) work by tightening an operational limit on top of a feasible base point, and do not yet cover failure modes that arise from structural changes such as topology cuts that island a load pocket, or generator outages that strand reactive demand without a nearby reactive source. A richer synthetic-infeasibility generator that probes these structural failure regimes is needed to broaden the feasibility classifier's coverage.

A related limitation is the broader perturbation applied in our data generation (§5): it does not currently include N-k branch contingencies (which the `gridfm-datakit` pipeline does [12]) or branch admittance (R, X) perturbations, and our load-scaling envelope tops out at 1.5× nominal on the feasible side. These are complementary axes that future training rounds can incorporate to broaden coverage of structurally stressed operating regimes.

Another limitation is in the prediction scope: the current model predicts only primal AC-OPF variables. The current model backbone has no inherent restriction to primal outputs, and a natural next step is to extend GridSFM to directly predict dual variables (locational marginal prices, branch shadow prices) and per-branch congestion cost [15], which would unlock market-clearing and congestion-management workflows alongside the dispatch and feasibility outputs the current release already provides.

11 Released Artifacts

The GridSFM source code, training pipeline, evaluation harness, and warm-start integration with PowerModels.jl are released under `https://github.com/microsoft/GridSFM`. The released model checkpoints, including the GridSFM-Open checkpoint used throughout the evaluation in §6.1–§6.3, are hosted in the Hugging Face collection at `https://huggingface.co/collections/microsoft/gridsfm`. The continental US transmission topology corpus introduced in §5 is described in detail in the companion paper [11].

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[3] J. Carpentier, "Contribution à l'étude du dispatching économique," *Bulletin de la Société Française des Électriciens*, vol. 3, pp. 431–447, 1962.

[4] S. Frank, I. Steponavice, and S. Rebennack, "Optimal power flow: a bibliographic survey I — formulations and deterministic methods," *Energy Systems*, vol. 3, no. 3, pp. 221–258, 2012.

[5] D. K. Molzahn and I. A. Hiskens, "A survey of relaxations and approximations of the power flow equations," *Foundations and Trends in Electric Energy Systems*, vol. 4, no. 1–2, pp. 1–221, 2019.

[6] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006.

[7] M. Desbrun, A. N. Hirani, M. Leok, and J. E. Marsden, "Discrete exterior calculus," arXiv:math/0508341, 2005.

[8] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "PowerModels.jl: An open-source framework for exploring power flow formulations," in *Proceedings of the Power Systems Computation Conference (PSCC)*, 2018.

[9] S. Babaeinejadsarookolaee *et al.*, "The power grid library for benchmarking AC optimal power flow algorithms," arXiv:1908.02788, 2019.

[10] S. Lovett, M. Zgubic, S. Liguori, S. Madjiheurem, H. Tomlinson, S. Elster, C. Apps, S. Witherspoon *et al.*, "OPFData: Large-scale datasets for AC optimal power flow with topological perturbations," arXiv:2406.07234, 2024.

[11] A. Britto, T. Spina, W. Yang, S. Fowers, B. Zhang, and C. White, "Building power grid models from open data: A complete pipeline from OpenStreetMap to optimal power flow," *arXiv preprint arXiv:2605.04289*, 2026. [Online]. Available: https://arxiv.org/abs/2605.04289

[12] Linux Foundation Energy and IBM Research, "GridFM DataKit: A data-generation pipeline for power-systems foundation models," https://github.com/gridfm/gridfm-datakit, 2025.

[13] B. Stott, J. Jardim, and O. Alsaç, "DC power flow revisited," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290–1300, 2009.

[14] Linux Foundation Energy and IBM Research, "GridFM GraphKit: Graph neural network for power systems," https://github.com/gridfm/gridfm-graphkit, 2025.

[15] F. C. Schweppe, M. C. Caramanis, R. D. Tabors, and R. E. Bohn, *Spot Pricing of Electricity*. Springer (Kluwer Academic Publishers), 1988.