MYCELIUM: A GENERALIZABLE CROSS-GRID MULTI-TASK MODEL FOR ELECTRICAL DISTRIBUTION SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Electrical distribution grid operations require inference across heterogeneous networks from sparse, noisy, and incomplete time series measurements. In this work, we identify challenges and explore solutions towards a unified model that can perform diverse tasks grounded in the physics of the electric grid and generalize to unseen distribution networks. We define a unified grid ontology that represents variable sized distribution networks as heterogeneous graphs while preserving native topology, asset types, and electrical relationships across networks. We develop a physics based data simulation pipeline, that combines reference and procedurally generated distribution networks with network reconfigurations, fault scenarios, and configurable sensing conditions. We present *Mycelium*, a heterogeneous graph transformer with structure aware communication edges and electrical reference features that encode network position and nominal phase orientation, together with task specific temporal readouts which generate per task outputs. We train Mycelium on reference as well as synthetic grids, and study its generalization on benchmark networks completely excluded from training and validation. Mycelium is observed to outperform task specific neural baselines on most reported benchmark metrics. Architectural ablations and the aforementioned studies reveal Mycelium's capability to learn representations of the underlying physics which serves to enhance cross-task performance, thereby addressing a significant challenge in unified grid models.

1 INTRODUCTION

Distribution networks supply homes, businesses, and critical services while integrating rooftop solar, batteries, electric vehicles, and electrified heating. Reliable operation amid increasingly variable demand and generation requires better monitoring and inference (Cleenwerck et al., 2026). Machine learning offers a promising approach by learning physics based invariant patterns across networks and operating conditions (Hamann et al., 2024). However, early efforts toward general purpose grid models have emphasized transmission systems, where measurements and power flow data are more standardized (Hamann et al., 2024; Puech et al., 2026).

Extending these capabilities to distribution networks presents distinct challenges. Transmission networks are generally better instrumented and commonly modeled using balanced single-phase equivalents (Hamann et al., 2024). Distribution networks combine radial or weakly meshed topologies with diverse equipment, voltage levels, and phase configurations, and distribution measurements are often sparse, asynchronous, noisy, or restricted by privacy and ownership. Evolving connectivity and operating conditions can also render network records outdated. Limited central monitoring of low voltage switches (für Elektrizität & Gas, 2022) illustrates why switch states and customer phase assignments may require inference alongside electrical states. Thus, abundant power sector data do not necessarily provide consistently labeled training examples. In developing countries, limited data infrastructure, access, and analytical capacity further constrain their use (Energy Sector Management Assistance Program, 2017). Existing learning based approaches address important distribution grid tasks, including state estimation, phase attribution, and fault diagnosis (Duan & Stewart, 2019; Dande et al., 2025; Chanda & Soltani, 2024). However, a model developed for a particular network can rely on patterns specific to its topology, measurement config-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

uration, and operating distribution. Maintaining separate models for every network and task would multiply data, computation, and engineering requirements as networks evolve. This motivates the use of shared models that remain useful across heterogeneous networks and support multiple inference tasks. The central question is whether a single model can generalize to unseen distribution networks under sparse, noisy, and incomplete measurements.

We introduce *Mycelium*, a shared heterogeneous graph model for state estimation, phase attribution, switch state inference, and fault diagnosis. Mycelium uses heterogeneous equipment representations, a domain specific graph representation with virtual communication edges, and inductive biases constraining electrical quantity outputs to support learning across topologically and electrically diverse networks. We combine this architecture with a procedure for synthetically generating diverse yet representative distribution grids to study the intersection of topological generalization and task generalization under a single model. Our main contributions are:

- **Domain informed architecture.** We introduce a heterogeneous graph architecture combining equipment aware representations, communication structures beyond the grid's own topology, and reference setpoints for quantities such as voltage angles. Ablation studies are performed to quantify their effects across tasks.
- **Multi-task capability with sparse measurements.** Mycelium supports the tasks of state estimation, phase attribution, switch state inference, and fault diagnosis from sparse, noisy, and incomplete measurements. Comparisons with single task training and finetuning characterize the benefits and tradeoffs of sharing.
- **Out-of-distribution generalization.** We evaluate Mycelium on 52 unseen test grids and a frozen benchmark of five grids from source families excluded from training and validation. The results show promising cross-grid generalization, which we further characterize through ablations among architectural choices, task outputs, and grid size.
- **Physics grounded data generation and augmentation.** We develop a simulation pipeline combining curated reference networks and procedurally generated synthetic networks under network reconfigurations, faults, and varied sensing conditions. Scaling experiments quantify the benefits and task dependent tradeoffs of synthetic augmentation on unseen grids.

2 Related Work

Topology aware and cross-grid graph learning. GridFM (Hamann et al., 2024) propose initial steps towards a grid foundation model, introducing the idea of masked reconstruction over diverse grid topologies and operating conditions, followed by finetuned adaptation. Zhu et al. (2025) propose a physics informed self supervised graph transformer with reusable initialization across power flow (PF), optimal PF (OPF), contingency analysis, and outage prediction, but demonstrate neither cross-grid transfer nor a single shared model across grids. MxGPS (Papaioannou et al., 2026) show that joint PF and state estimation training can improve cross-grid robustness, but at the cost of in-distribution accuracy and with only two related regression tasks on small, balanced transmission networks. Finally, Huang et al. (2026) show that a frozen encoder can provide a reusable representation interface across unseen transmission grids and heterogeneous computational tasks. Their downstream computations nevertheless rely on decoders trained independently per task.

Learning for distribution systems. Zamzam & Sidiropoulos (2020) exploit known grid topology to sparsify neural estimators. DSS^2 (Habib et al., 2023) combines heterogeneous graphs with weighted least squares (WLS) residuals and power flow constraints to learn the full state without clean state labels, while F-PINN (Azam et al., 2026) uses factor graphs and augmented Lagrangian training to enforce physical constraints. For phase identification, Dande et al. (2025) train GNNs on network specific power flow simulations, assuming known customer mapping to upstream transformers. Switch state inference has progressed from CNN classification of simulated micro phasor measurement unit (μPMU) time series (Duan & Stewart, 2019) to topology aware GNNs (Madbhavi et al., 2023). Alternatively DGS-GNN (Ebtia et al., 2024) uses GraphSAGE (Hamilton et al., 2017) embeddings of measurements alongside physical topology to recover switch states. For fault diagnosis, Chanda & Soltani (2024) jointly predict various attributes of a given fault event using graph convolutional networks (GCNs), while Karabulut et al. (2026) study

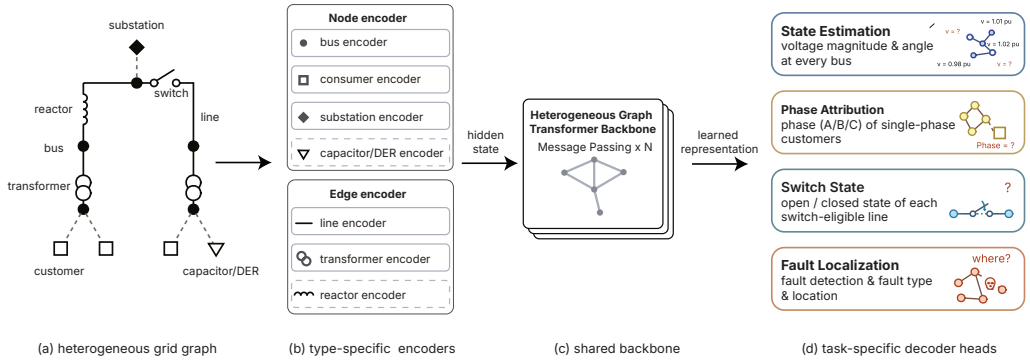

Figure 1: Mycelium architecture. (a) A heterogeneous graph models network components and electrical relations. (b) Type specific encoders embed attributes and temporal measurements. (c) A shared heterogeneous graph transformer performs spatial message passing. (d) Task specific temporal heads support SE, PA, SW, and fault diagnosis.

spatiotemporal fault localization under distributed energy resources (DER) shifts, finding asymmetric generalization across penetration regimes. These studies suffer from specificity to grid topologies, leaving the problem of cross-grid generalization understudied.

Generalist learning for transmission systems. Recent work extends task specific surrogates toward cross-grid and generalist models. CANOS (Piloto et al., 2024) approximates AC OPF on networks of up to 10,000 buses under $N - 1$ outages, but relies on per grid training. PFΔ (Rivera Him et al., 2026) benchmarks power flow across six large scale transmission systems, but demonstrates limitations in electrical feasibility and cross-grid transfer. LUMINA (Li et al., 2026) performs a study in zero-shot ac OPF cross-grid generalization through pretraining across multiple topologies, but only demonstrates generalization across up to two unseen grids. GridSFM (Yang et al., 2026) pretrains across approximately 200 networks for ac OPF and feasibility screening, although cross-grid generalization is shown to require finetuning. GENCO (Puech et al., 2026) provides a common architecture for power flow, OPF, and state estimation, but uses separate task/grid checkpoints and evaluates cross-grid pretraining only for power flow. The aforementioned works show a lack of cross-grid generalization across multiple tasks, especially under temporally varying sparse measurements.

3 Task Overview and Mycelium Architecture

3.1 The Four Tasks

We infer electrical states, connection attributes, and fault events from a network graph and partial measurements over $T = 24$ hourly timesteps. Table 7 summarizes the four tasks: state estimation (SE) predicts continuous voltage trajectories. Phase attribution (PA) and switch state inference (SW) classify consumer connections and branch states, respectively, which remain fixed within the window. Switch locations are known, while their open/closed states are predicted via edge classification. Fault diagnosis (F) predicts normal operation or one of five fault classes and, for faulted samples, the affected equipment under the assumption that the fault onset occurs at the final observed timestep. We perform inference from available observations without assuming control of generation or grid equipment. Formal definitions are given in Appendix B.

3.2 Mycelium's Graph Representation and Inputs

Mycelium encodes the structural graph and masked measurements, applies shared spatial message passing at each timestep, and uses task specific temporal readouts to produce the four outputs. Figure 1 summarizes the architecture. Each network is defined as a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with bus, consumer, and substation nodes, together with capacitor and DER nodes

when present. Lines, transformers, and reactors form distinct bus–bus relations. Service and source links attach consumers and substations to their buses. All switch eligible line edges remain present regardless of operating state. The network components satisfy the AC power flow equations (Grainger & Stevenson, 1994), given as $\mathbf{f}(\mathbf{V}_t, \mathbf{\Theta}_t, \mathbf{p}_t, \mathbf{q}_t; \mathcal{G}, \mathbf{c}_t) = \mathbf{0}$, where $\mathbf{V}_t$ and $\mathbf{\Theta}_t$ are per phase bus voltage magnitudes and angles, $\mathbf{p}_t$ and $\mathbf{q}_t$ are net active and reactive power injections, and $\mathbf{c}_t$ collects equipment parameters and operating configurations. Equipment nameplate ratings and static electrical parameters are assumed available, while switch open/closed states are excluded from edge inputs. Inputs combine equipments' static attributes with available voltage, power, current, and feeder head measurements. Observability masks distinguish zero filled missing readings from observed zeros. Separate supervision masks identify valid prediction targets. Type specific node and edge encoders map these inputs into a common hidden dimension while retaining the temporal axis. Their parameters are shared across entities of the same type and across networks. Appendix C.1 describes the input representation and masks.

3.3 Distribution Grid Inductive Biases

Communication Edges. Long radial paths can require many message passing layers to exchange information between customers and the substation. We shorten these paths with two typed bidirectional communication edges, illustrated in Figure 2: *uplinks* connect the substation to each distinct distribution transformer primary bus, and *downlinks* connect customers to their serving transformer's secondary bus. Together with the physical transformer relation, these bidirectional shortcuts place each customer with a valid transformer association within three message passing hops of the substation. The connections are derived from the available topology and customer registry and introduce no additional measurement features. Relation specific parameters distinguish the shortcuts from physical electrical connections.

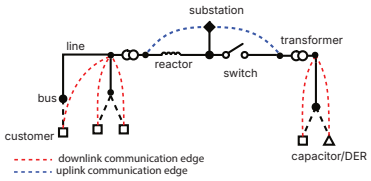

Figure 2: Mycelium's message passing graph inherits the base grid topology and adds communication edges. Blue dashed edges connect the substation to transformer primary buses, and red dashed edges connect customers to their serving transformer's secondary bus.

We analyze voltage prediction with a linear learner using power-of-two ancestor links. This idealized communication graph differs from Mycelium's equipment-based uplinks and downlinks. On a radial grid of physical depth $h \geq 1$, let $\mathbf{V}$ contain bus voltage magnitudes and $\mathbf{p}, \mathbf{q}$ the active and reactive injections. Assume the linear distribution flow (LinDistFlow) model (Deka et al., 2024),

$$\mathbf{V}^{\odot 2} = 2\mathbf{R}\mathbf{p} + 2\mathbf{X}\mathbf{q} + V_0^2\mathbf{1}. \tag{1}$$

Here squaring is elementwise, V_0 is fixed, and $\mathbf{R}, \mathbf{X}$ encode grid topology and powerline resistances and reactances. Suppose the true injections have joint covariance $\mathbf{\Sigma} \succ 0$. Each bus receives its own injection measurements, with additive zero-mean noise independent of the injections and joint noise covariance $\tau\mathbf{\Sigma}$, where $\tau \geq 0$ is the ratio of sensing variance to ground truth variance. With a slight abuse of notation, $\mathbf{p}, \mathbf{q}$ also denote the sensed learner inputs. We study the prediction task $(\mathbf{p}, \mathbf{q}) \mapsto \mathbf{V}^{\odot 2}$. Let $\mathcal{D}$ denote the joint probability distribution of injections and measurement noise described above. For snapshots $s \sim \mathcal{D}$, denote predicted squared magnitudes by $\widehat{\mathbf{V}^{\odot 2}}_{\mathbf{w}}(s)$ and train weights $\mathbf{w}$ on

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}\,\mathbb{E}_{s\sim\mathcal{D}}\left[\left\|\widehat{\mathbf{V}^{\odot 2}}_{\mathbf{w}}(s) - \mathbf{V}(s)^{\odot 2}\right\|_2^2\right]. \tag{2}$$

Proposition 1 (Exact communication and voltage prediction)**.** *Under the preceding assumptions, there exists a linear learner H for this task that computes each prediction in $2\lceil \log_2 h \rceil$ message passing rounds using physical grid edges and added bidirectional ancestor edges. Let $\lambda_j > 0$ be its loss-Hessian eigenvalues and c_j its initial parameter errors relative to the optimum in an orthonormal eigenbasis. Use the constant gradient descent (GD) step size $\eta' = \eta/(1+\tau)$, where η is the step size for noiseless inputs. After $n \geq 1$ full GD steps from zero initialization with $0 < \eta' < 2/\max_j \lambda_j$, its error $\mathcal{E}_{\mathrm{H}}(n) = 2\mathcal{L}(\mathbf{w}^{(n)})$ satisfies, for every integer $\ell \geq 2\lceil \log_2 h \rceil$,*

$$\mathcal{E}^{\star}_{\mathrm{P}}(\ell) - \mathcal{E}_{\mathrm{H}}(n) = \frac{C_\ell}{1+\tau} - \sum_j \lambda_j (1-\eta'\lambda_j)^{2n} c_j^2. \tag{3}$$

Here $\mathcal{E}^{\star}_{\mathrm{P}}(\ell)$ is the minimum error among linear predictors using only physical grid edges and at most ℓ message passing rounds per prediction.

Here C_ℓ is the squared voltage variation not linearly explained by injections within ℓ physical hops. When $C_\ell > 0$, the gap becomes positive after finitely many steps and tends to $C_\ell/(1+\tau)$. Appendix C.3 gives the Schur-complement expression, learner construction, and proof. The result assumes exact LinDistFlow, injection coverage at every bus, clean voltage targets, and population GD. Mycelium's nonlinear model, partial measurements, and AdamW training are evaluated empirically.

Nominal Angle Reference. Voltage angles contain predictable offsets from source phase sequences and transformer connections. We derive a static nominal angle $\theta^{\mathrm{nom}}_{i,p}$ for each (bus, phase) pair from equipment metadata, without solved voltage trajectories. Observed angles and targets are expressed relative to this reference as $\Delta\theta_{i,p,t} = \mathrm{wrap}(\theta_{i,p,t} - \theta^{\mathrm{nom}}_{i,p})$, where wrap maps angles to $[-\pi, \pi)$, and predictions are converted back as $\widehat{\theta}_{i,p,t} = \mathrm{wrap}(\theta^{\mathrm{nom}}_{i,p} + \widehat{\Delta\theta}_{i,p,t})$. This parameterization focuses regression on deviations from nominal phase structure. Appendix C.4 describes the reference construction.

3.4 Shared Backbone and Task Readouts

We use a heterogeneous graph Transformer (Hu et al., 2020) with relation specific attention over physical, attachment, and communication edges. At layer ℓ, each node aggregates messages within each relation and sums their contributions:

$$\mathbf{m}^{(\ell)}_{i,t} = \sum_{r} \Big\Vert_{k=1}^{H} \sum_{j \in \mathcal{N}_r(i)} \alpha^{(\ell,r,k)}_{ji,t} \mathbf{v}^{(\ell,r,k)}_{ji,t}. \tag{4}$$

Here, $\mathcal{N}_r(i)$ contains incoming neighbors under relation r, k indexes the H attention heads, and $\|$ denotes concatenation. The coefficient $\alpha^{(\ell,r,k)}_{ji,t}$ weights the message $\mathbf{v}^{(\ell,r,k)}_{ji,t}$ from j to i. Head outputs are concatenated and relation outputs summed. Physical edge embeddings condition both attention scores and message values. Each layer combines attention with pre normalization and an FFN specific to the node type, with residual connections around both. Spatial attention operates independently at each timestep with shared parameters, preserving $\mathbf{H}^{(\ell)} \in \mathbb{R}^{|\mathcal{V}|\times T\times d}$ without temporal pooling. The model uses eight layers, hidden dimension 128, four attention heads, and approximately 12.2M parameters. Appendix C.6 provides further details. For state estimation, a causal dilated temporal convolutional network (Bai et al., 2018) processes each bus sequence. A shared linear head predicts per phase voltage magnitudes and nominal angle residuals, with absolute angles recovered as described in Appendix C.4. For phase attribution, mask aware temporal attention pools consumer, attached bus, and network representations for three-way classification. For switch inference, the readout combines attention weighted, mean, and maximum temporal summaries of endpoint and branch representations. For fault diagnosis, a temporal readout compares the final representation with an attended history using signed and absolute differences. Network and candidate readouts produce $\mathbf{z}^{\mathrm{F}}_{\mathcal{G}}$ and $\mathbf{z}^{\mathrm{F}}_{c}$. A shared scorer evaluates the network's bus, line, and transformer candidates, $s_c = f_{\mathrm{loc}}([\mathbf{z}^{\mathrm{F}}_{c} \| \mathbf{z}^{\mathrm{F}}_{\mathcal{G}}])$ for $c \in \mathcal{C}(\mathcal{G})$. Scoring the available candidates accommodates different network sizes without a fixed location vocabulary. A separate classifier combines network context and maximum pooled candidate evidence to distinguish normal operation and five fault classes. Appendix C.7 describes all readouts.

3.5 Multi-task Training

We jointly optimize the backbone and task heads with unit coefficients for the four task losses. State estimation uses scaled Smooth-L_1 errors for voltage magnitude and wrapped angle. Phase attribution and switch inference use cross entropy. The fault loss combines class weighted six-class classification and equipment localization. For entity level losses, we average over valid targets within each network window and then over applicable windows, preventing target count alone from giving larger networks greater weight. Appendix C.8 describes the detailed losses and masking.

Table 1: Network level assignments.

Split	Networks
Train	SMART-DS AUS/SFO (70% each); registered IEEE feeder; EPRI `ckt5`, `ckt24`; D-Suite `spd_r/s/u`
Validation	SMART-DS AUS/SFO (15% each); EPRI `ckt7`; D-Suite `spm_s`
Test	SMART-DS AUS/SFO (15% each) and all GSO; D-Suite `spm_r/u`
Benchmark	Iowa 240- and 296-bus; European industrial, rural, and urban

4 Data and Evaluation Protocol

4.1 Data

Reference and Synthetic Network Corpus The reference corpus combines publicly available models from NREL SMART-DS (Palmintier et al., 2020), EPRI, the IEEE test feeder suite (Schneider et al., 2017), D-Suite (Deakin, 2025), Iowa State (Bu et al., 2019), and European utilities (Taye et al., 2024). Appendix A.1 describes these sources. We simulate the network models in OpenDSS (Dugan & McDermott, 2011) using convergence checked three-phase unbalanced power flow. Operating cases include transformer reassignment, consumer phase changes, coordinated switching, and faults at buses, lines, and transformer terminals. Seven sensor policies vary measurement coverage, noise, and missingness (Appendix D). The reference corpus contains 98,799 electrical cases and 478,891 retained sensor policy samples after filtering. We also generate diverse feeders from street and building geometry, grouping customers into transformer service areas and routing primary and secondary networks. Equipment parameters and planning conventions are sampled from the training reference networks, with 24-hour demand, reactive power, and PV profiles defining operating conditions. Geography, equipment conventions, load climate, and solar region are sampled separately. OpenDSS (Dugan & McDermott, 2011) checks daily operation and high load and high PV stress cases. Bounded repairs adjust phases, taps, reactive support, and equipment sizing. Networks still violating convergence or operating limits are rejected. Appendix A.3 details generation and acceptance criteria.

Data Split We partition reference networks before generating variants, temporal windows, or sensor policy realizations. All derived samples retain their source network's assignment. Each split includes networks exhibiting three task relevant characteristics: meshed topology, single-phase customers served by three-phase distribution transformers, and multiple distribution transformers. Table 1 summarizes the network disjoint assignments. The test split contains 52 unseen networks and 10,038 samples, largely from source families represented in training. The frozen benchmark contains 1,043 simulated samples on five utility derived network models from source families excluded from training and validation. The two splits assess generalization across networks and source families, respectively. Checkpoint selection uses validation loss. Because benchmark scenarios and sensor policies repeatedly sample the same five networks, the sample count does not represent 1,043 independent network structures.

4.2 Metrics

State estimation. Over valid (bus, timestep, phase) entries, we report voltage magnitude MAE in per unit (p.u.) and volts (v_{mag}), angle MAE in degrees (v_{ang}), and magnitude MAPE. Per unit errors use each bus's voltage base, so the two units can rank models differently across networks. **Phase attribution.** We report three-class top-1 accuracy for single-phase customers on multiphase transformers with at least one observed reading. **Switch state.** We report precision, recall, and F1 over applicable switchable lines, excluding switches whose two endpoints are marked de-energized. OPEN is the positive class, and predictions use a fixed zero logit threshold. **Fault diagnosis.** We report detection F1 (Det. F1) over normal and faulted samples in each split. On faulted samples, we report five-class type accuracy (class_5), exact equipment localization accuracy (Loc.), and localization within k graph hops (Hop@k).

Table 2: Mycelium and task-specific neural baselines on the held-out test split and frozen benchmark. Dashes indicate tasks a baseline does not model. Lower is better for state estimation errors and higher is better otherwise. Best values within each split are bold.

			State estimation				Phase	Switch			Fault					
Split	Baseline	Task	p.u. ↓	v_{mag} (V) ↓	v_{ang} (deg) ↓	v_{mag} MAPE (%) ↓	acc ↑	prec ↑	rec ↑	F1 ↑	det F1 ↑	class_5 ↑	loc ↑	hop@1 ↑	hop@2 ↑	hop@3 ↑
Test	CANOS(Piloto et al., 2024)	State est.	**0.00221**	**2.86**	**0.576**	**0.136**	-	-	-	-	-	-	-	-	-	-
	GraphSAGE-CPI(Dande et al., 2025)	Phase	-	-	-	-	0.3276	-	-	-	-	-	-	-	-	-
	DGS-GNN(Ebtia et al., 2024)	Switch	-	-	-	-	-	0.0807	0.2399	0.1207	-	-	-	-	-	-
	STGATv2(Karabulut et al., 2026)	Fault	-	-	-	-	-	-	-	-	-	-	0.1349	0.2162	0.2683	0.3107
	Ours (10k FINAL)	All four	0.00345	5.99	1.107	0.234	**0.5713**	**0.9960**	**0.8137**	**0.8957**	**1.0000**	**0.4840**	**0.1710**	**0.2369**	**0.2999**	**0.3314**
Benchmark	CANOS(Piloto et al., 2024)	State est.	0.01754	**9.50**	3.254	5.591	-	-	-	-	-	-	-	-	-	-
	GraphSAGE-CPI(Dande et al., 2025)	Phase	-	-	-	-	0.3369	-	-	-	-	-	-	-	-	-
	DGS-GNN(Ebtia et al., 2024)	Switch	-	-	-	-	-	0.4737	0.4114	0.4404	-	-	-	-	-	-
	STGATv2(Karabulut et al., 2026)	Fault	-	-	-	-	-	-	-	-	-	-	0.1177	0.1772	0.1825	0.2262
	Mycelium (10k FINAL)	All four	**0.01267**	81.28	**1.635**	**5.425**	**0.4001**	**0.9553**	**0.7198**	**0.8210**	**1.0000**	**0.7209**	**0.1958**	**0.3175**	**0.3558**	**0.4114**

4.3 Implementation Details

We train Mycelium on eight 80 GB NVIDIA A100 GPUs using FP32, activation checkpointing, and a global batch size of 64. The backbone has eight Transformer layers with pre normalization, hidden width 128, four attention heads, feedforward width 256, and dropout 0.1. Training uses 12,800 samples per epoch for up to 100 epochs, with AdamW (Loshchilov & Hutter, 2017), learning rate 10^{-3}, cosine decay, weight decay 10^{-2}, and gradient clipping at 1.0. We retain the checkpoint with the lowest 4 loss and stop after ten epochs without improvement. The reference configuration combines reference and synthetic networks with a 10,000-bus training cap. Appendix F details memory optimization.

5 Experiments

5.1 Baseline Comparisons

We compare Mycelium with task specific baselines: WLS and a CANOS based estimator (Piloto et al., 2024) adapted to the same inputs for state estimation, correlation based matching and GraphSAGE-CPI (Dande et al., 2025) for phase attribution, DGS-GNN (Ebtia et al., 2024) for switch state inference, and STGATv2 (Karabulut et al., 2026) for fault diagnosis.

Table 2 compares task-specific neural baselines with Mycelium 10k, our reference configuration trained on the full reference training set and synthetic network corpus with a maximum training network size of 10,000 buses. This size cap applies only during training. Evaluation networks are not truncated. CANOS achieves lower errors on all four state estimation metrics on the test split. On the frozen benchmark, Mycelium has lower per-unit voltage error, angle error, and magnitude MAPE, but higher voltage error in volts: 81.28V versus 9.50V. Across both splits, Mycelium improves phase accuracy over GraphSAGE-CPI, switch state F1 over DGS-GNN, and exact fault localization over STGATv2. We report WLS state estimation and correlation-based phase matching in Appendix E. Together, these comparisons show that one Mycelium checkpoint supports all four tasks with competitive accuracy on feeders from unseen source families, without the privileged operating information supplied to WLS. Appendix E.4 reports per-network and network macro results, revealing substantial variation across benchmark networks.

5.2 Ablation Studies

Communication graph. Table 3 distinguishes the effects of shortcut connectivity and placement. Removing shortcuts reduces test switch F1 from 0.8957 to 0.3253, but random rewiring outperforms the proposed placement on switch F1 on both splits: 0.9334 versus 0.8957 on test and 0.9278 versus 0.8210 on the benchmark. The proposed placement achieves lower benchmark per-unit voltage error and higher fault type accuracy than random rewiring. These results support the utility of added connectivity for test-set switch inference, but do not establish a general advantage of the proposed placement.

Nominal angle reference. Removing the nominal angle reference while keeping the architecture and training protocol fixed increases angle error from 1.107 to 16.461 degrees on the test split and from 1.635 to 24.896 degrees on the frozen benchmark (Table 3). Thus, the reference reduces angle error approximately fifteenfold on both splits.

Table 3: Architectural ablations on the test split and benchmark: (a) communication edges and (b) nominal angle references. Random rewiring preserves edge counts and endpoint types per relation and network. Training data and optimization remain fixed. Best values are bold.

		State estimation				Phase	Switch			Fault					
Split	Configuration	p.u. ↓	v_{mag} (V) ↓	v_{ang} (deg) ↓	MAPE (%) ↓	Acc. ↑	Prec. ↑	Rec. ↑	F1 ↑	Det. F1 ↑	class_5 ↑	Loc. ↑	Hop@1 ↑	Hop@2 ↑	Hop@3 ↑
(a) Test	No communication	0.00370	**4.99**	1.286	0.240	**0.6611**	0.1950	**0.9800**	0.3253	**1.0000**	0.4540	0.1819	**0.2743**	**0.3464**	**0.3822**
	Random communication	**0.00316**	5.24	**1.083**	**0.230**	0.6328	0.9631	0.9056	**0.9334**	**1.0000**	0.4241	**0.1881**	0.2628	0.3267	0.3583
	Mycelium 10k	0.00345	5.99	1.107	0.234	0.5713	**0.9960**	0.8137	0.8957	**1.0000**	**0.4840**	0.1710	0.2369	0.2999	0.3314
Benchmark	No communication	0.01866	**77.46**	1.639	5.955	**0.4256**	0.7945	**0.9968**	0.8842	**1.0000**	0.7143	0.1878	0.2950	0.3492	0.4034
	Random communication	0.01805	78.24	**1.543**	5.773	0.4231	0.8721	0.9910	**0.9278**	0.9993	0.6468	**0.1971**	0.2857	0.3135	0.3810
	Mycelium 10k	**0.01267**	81.28	1.635	**5.425**	0.4001	**0.9553**	0.7198	0.8210	**1.0000**	**0.7209**	0.1958	**0.3175**	**0.3558**	**0.4114**
(b) Test	No angle reference	**0.00333**	**5.02**	16.461	0.239	**0.6272**	0.9914	**0.8939**	**0.9401**	**1.0000**	**0.4970**	**0.1774**	**0.2484**	**0.3190**	**0.3497**
	Mycelium 10k	0.00345	5.99	**1.107**	**0.234**	0.5713	**0.9960**	0.8137	0.8957	**1.0000**	0.4840	0.1710	0.2369	0.2999	0.3314
Benchmark	No angle reference	**0.01054**	**73.97**	24.896	**5.265**	**0.4203**	0.8193	**0.9738**	**0.8899**	**1.0000**	0.7011	0.1786	0.2804	0.3360	0.3968
	Mycelium 10k	0.01267	81.28	**1.635**	5.425	0.4001	**0.9553**	0.7198	0.8210	**1.0000**	**0.7209**	**0.1958**	**0.3175**	**0.3558**	**0.4114**

Table 4: Data scaling on the test split and benchmark: (a) synthetic corpus fractions added to the full reference set with a 10,000-bus cap, and (b) training network size caps with full synthetic augmentation. Mycelium 10k is the shared reference, and evaluation sets remain fixed. Best values are bold.

		State estimation				Phase	Switch			Fault					
Split	Configuration	p.u. ↓	v_{mag} (V) ↓	v_{ang} (deg) ↓	MAPE (%) ↓	Acc. ↑	Prec. ↑	Rec. ↑	F1 ↑	Det. F1 ↑	class_5 ↑	Loc. ↑	Hop@1 ↑	Hop@2 ↑	Hop@3 ↑
(a) Test	Reference only	0.00881	7.63	**0.933**	0.320	0.5846	**0.9992**	0.9178	0.9568	**1.0000**	0.3455	0.1526	0.2305	0.2899	0.3249
	+25% synthetic	**0.00302**	**4.45**	1.116	**0.197**	**0.5922**	0.9806	**0.9542**	**0.9672**	**1.0000**	0.4708	**0.1839**	**0.2820**	**0.3437**	**0.3721**
	Mycelium 10k	0.00345	5.99	1.107	0.234	0.5713	0.9960	0.8137	0.8957	**1.0000**	**0.4840**	0.1710	0.2369	0.2999	0.3314
Benchmark	Reference only	0.04025	148.88	2.388	7.714	0.3539	0.9445	0.9946	0.9689	**1.0000**	0.2937	0.1323	0.1799	0.2249	0.3016
	+25% synthetic	0.01991	141.51	**1.134**	5.669	0.3886	**0.9585**	**0.9958**	**0.9768**	**1.0000**	0.6111	0.1429	0.2156	0.2500	0.3056
	Mycelium 10k	**0.01267**	**81.28**	1.635	**5.425**	**0.4001**	0.9553	0.7198	0.8210	**1.0000**	**0.7209**	**0.1958**	**0.3175**	**0.3558**	**0.4114**
(b) Test	5,000 buses	**0.00322**	**5.21**	1.198	**0.224**	0.5776	0.9716	**0.9042**	**0.9367**	**1.0000**	0.4479	0.1676	0.2501	0.3034	0.3315
	20,000 buses	0.00393	6.96	**0.979**	0.289	**0.6150**	0.9800	0.8848	0.9299	0.9996	**0.5837**	**0.1918**	**0.2793**	**0.3524**	**0.3947**
	Mycelium 10k	0.00345	5.99	1.107	0.234	0.5713	**0.9960**	0.8137	0.8957	**1.0000**	0.4840	0.1710	0.2369	0.2999	0.3314
Benchmark	5,000 buses	**0.00963**	**73.11**	**1.071**	**5.177**	0.4234	0.9503	**0.9934**	**0.9713**	**1.0000**	0.6852	0.1799	0.3016	0.3267	0.3955
	20,000 buses	0.01063	75.72	1.456	5.332	**0.4387**	0.9258	0.5574	0.6959	0.9980	0.7063	**0.1971**	0.2897	0.3399	0.4008
	Mycelium 10k	0.01267	81.28	1.635	5.425	0.4001	**0.9553**	0.7198	0.8210	**1.0000**	**0.7209**	0.1958	**0.3175**	**0.3558**	**0.4114**

5.3 Data Scaling and Grid Size Generalization

Synthetic network scaling. All runs include the full reference training set. We add either no synthetic networks, a nested 25% subset matched to the full corpus's network size distribution, or the full synthetic corpus. Architecture, optimization, and training budget are held fixed, and all runs are evaluated on the same unseen reference networks. Synthetic augmentation improves several cross-family metrics, with nonmonotonic gains across tasks. Table 4(a) shows that the full synthetic corpus reduces benchmark per-unit voltage error by 69% relative to reference-only training and raises fault type accuracy from 0.2937 to 0.7209. The 25% subset instead achieves the best test-set voltage magnitude errors, switch F1, and exact localization. On the benchmark, switch F1 falls from 0.9768 at 25% to 0.8210 with the full corpus. Additional synthetic coverage therefore benefits several tasks without uniformly improving performance.

Maximum training network size. We train models with network size caps of 5,000, 10,000, and 20,000 buses, keeping the evaluation sets unchanged. The 20,000-bus run requires a batch size of four rather than eight, so this comparison varies effective batch size. Larger training networks do not uniformly improve generalization. Table 4(b) shows that the 20,000-bus run achieves the highest test-set phase accuracy, fault type accuracy, and exact fault localization accuracy, but has the highest test-set voltage magnitude error. On the frozen benchmark, the 5,000-bus run has the lowest state estimation errors and highest switch F1, whereas the 20,000-bus run has the highest phase accuracy. Exposure to larger networks therefore does not improve every task, and the batch size difference prevents attributing these changes solely to network size.

5.4 Multi-task vs. Single-task

We compare the cross-task Mycelium model with four models that use the same backbone but receive supervision for only one task. Each single-task model is evaluated on its trained task. Joint training provides selective cross-family benefits. Table 5 reveals a reversal in state estimation perfor-

Table 5: Joint versus single-task training with the same backbone on the test split and frozen benchmark. Dashes mark untrained tasks. Lower is better for state estimation errors and higher otherwise. Best values per split are bold.

Split	Run	State estimation				Phase	Switch			Fault					
		p.u. ↓	v_{mag} (V) ↓	v_{ang} (deg) ↓	v_{mag} MAPE (%) ↓	acc ↑	prec ↑	rec ↑	F1 ↑	det F1 ↑	$class_5$ ↑	exact ↑	hop@1 ↑	hop@2 ↑	hop@3 ↑
Test	SE only	**0.00148**	**3.33**	**1.057**	**0.148**	-	-	-	-	-	-	-	-	-	-
	Phase only	-	-	-	-	0.5363	-	-	-	-	-	-	-	-	-
	Switch only	-	-	-	-	-	0.8325	**0.9374**	0.8818	-	-	-	-	-	-
	Fault only	-	-	-	-	-	-	-	-	0.7990	**0.5322**	**0.1927**	**0.2731**	**0.3441**	**0.3768**
	Ours (10k FINAL)	0.00345	5.99	1.107	0.234	**0.5713**	**0.9960**	0.8137	**0.8957**	**1.0000**	0.4840	0.1710	0.2369	0.2999	0.3314
Benchmark	SE only	0.07072	**21.51**	7.038	12.067	-	-	-	-	-	-	-	-	-	-
	Phase only	-	-	-	-	0.3683	-	-	-	-	-	-	-	-	-
	Switch only	-	-	-	-	-	0.7870	**0.9961**	**0.8793**	-	-	-	-	-	-
	Fault only	-	-	-	-	-	-	-	-	0.8405	0.6746	**0.2169**	**0.3188**	0.3320	0.3889
	Ours (10k FINAL)	**0.01267**	81.28	**1.635**	**5.425**	**0.4001**	**0.9553**	0.7198	0.8210	**1.0000**	**0.7209**	0.1958	0.3175	**0.3558**	**0.4114**

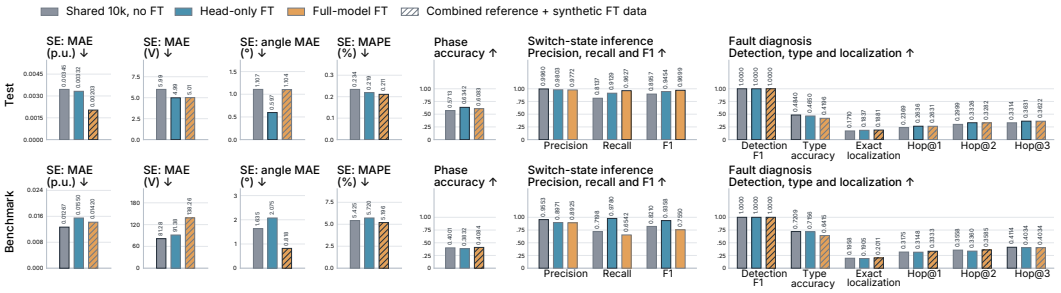

Figure 3: Single-task finetuning on the test split and frozen benchmark, shown in the top and bottom rows. Hatching denotes combined reference and synthetic finetuning data. Lower is better for state estimation errors and higher is better otherwise.

mance across splits. The single-task model has lower test per-unit error than joint training, 0.00148 versus 0.00345, but substantially higher benchmark error, 0.07072 versus 0.01267. Joint training also improves phase accuracy on both splits and benchmark fault type accuracy. Its benefits remain selective: single-task training retains lower benchmark voltage error in volts, higher benchmark switch F1, and better exact fault localization on both splits. Shared supervision therefore improves some aspects of cross-family generalization while leaving advantages for specialization.

5.5 Single-task finetuning

We finetune Mycelium 10k using only the target task loss. Head-only runs freeze the encoders and backbone and use reference networks. Full-model runs update all parameters using reference and synthetic networks, except for switch inference, which uses reference networks in both settings. For the other tasks, parameter scope and data composition therefore vary together. Figure 3 shows that finetuning improves test-set state estimation, phase accuracy, switch F1, and fault localization but reduces fault type accuracy. These gains do not consistently transfer to the benchmark. Head-only switch adaptation raises benchmark F1 from 0.8210 to 0.9358, whereas full-model adaptation reaches a higher test F1 (0.9699) but lowers benchmark F1 to 0.7550. This suggests the pretrained backbone encodes a powerful transferable representation that a frozen feature head preserves and full finetuning overwrites. Yet head-only phase adaptation also trades benchmark accuracy for test accuracy, so task specialization can improve in-family performance without improving cross-family generalization, even with a frozen backbone.

6 Conclusion

We introduced Mycelium, a shared model for four inference tasks across distribution grids. Synthetic augmentation and joint supervision improve selected cross-family metrics, while nominal angle references substantially reduce angle error. Communication edge placement and task-specific finetuning introduce tradeoffs, and gains on familiar source families do not consistently extend to excluded families. These results identify both benefits and limits of shared learning across grids.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code implementation, synthetic data generation, figure preparation, proving mathematical claims, manuscript feedback, and writing and polishing. Anthropic Claude Code contributed to the implementation of the multi-task model, synthetic network generator, data processing and evaluation pipelines, reimplemented baselines, and analysis and plotting scripts. AI tools helped identify potentially relevant literature, which the authors independently reviewed. The AI agent did not write the literature review. They also helped draft portions of the manuscript, edit language, and format LaTeX tables from measured results. AI tools were not used to formulate the research questions or design the experiments.

The authors reviewed all AI assisted work. AI generated code for data processing, model components, and metrics was covered by author reviewed unit tests, and bug fixes followed the repository's test first policy. Checks against the source publications revealed and corrected two substantive errors in AI assisted baseline implementations before training. We regenerated reported numbers from raw evaluation counters and checked every table against the stored results. The authors reviewed AI drafted prose and removed or qualified unsupported claims. We take full responsibility for the final text, results, and artifacts.

ETHICS STATEMENT

Our experiments use simulated measurements on reference and procedurally generated distribution network models. Because distribution grids are critical infrastructure, deployment would require independent validation under realistic sensing conditions and appropriate human oversight. The reported results do not establish operational reliability. Public data releases will respect source licenses and protect sensitive customer and infrastructure information.

REPRODUCIBILITY STATEMENT

The model and experimental sections describe the architecture, training configuration, baselines, and evaluation metrics. We specify the network disjoint partitioning, and Appendix details input features and observability masks. Additional formulations and experimental results are provided in the appendices. Upon publication, we will publicly release our code and data, including generation and evaluation scripts, configurations, and split manifests. For third party datasets with redistribution restrictions, we will provide source references and preprocessing instructions.

REFERENCES

M Furqan Azam, Marta Vanin, Chris Hermans, Reinhilde D'hulst, and Geert Deconinck. Physics-informed factor graph neural networks for constrained state estimation in low observable distribution networks. *Sustainable Energy, Grids and Networks*, pp. 102291, 2026.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Fankun Bu, Yuxuan Yuan, Zhaoyu Wang, Kaveh Dehghanpour, and Anne Kimber. A time-series distribution test system based on real utility data. In *2019 North American Power Symposium (NAPS)*, pp. 1–6. IEEE, 2019.

Dibaloke Chanda and Nasim Yahya Soltani. A heterogeneous graph-based multi-task learning for fault event diagnosis in smart grid. *IEEE Transactions on Power Systems*, 40(2):1427–1438, 2024.

Rémy Cleenwerck, Robbert Claeys, Thierry Coosemans, and Jan Desmet. Distribution networks in transition: A european survey on challenges and opportunities. *Renewable and Sustainable Energy Reviews*, 234:116885, 2026.

Chandra Sekhar Charan Dande, Nikolaos A Efkarpidis, Matthias Christen, Mirko Ginocchi, and Antonello Monti. Consumer phase identification in distribution grids using graph neural networks based on synthetic and measured power profiles. *Energy and AI*, pp. 100607, 2025.

Matthew Deakin. Six UK-based Low Voltage Distribution Network Models. 1 2025. doi: 10.25405/data.ncl.27175317.v1. URL https://data.ncl.ac.uk/articles/model/Six_UK-based_Low_Voltage_Distribution_Network_Models/27175317.

Deepjyoti Deka, Vassilis Kekatos, and Guido Cavraro. Learning distribution grid topologies: A tutorial. *IEEE Transactions on Smart Grid*, 15(1):999–1013, 2024. doi: 10.1109/TSG.2023.3271902. URL https://engineering.purdue.edu/~kekatos/papers/TSG2022-tutorial.pdf.

Nan Duan and Emma M Stewart. Deep-learning-based power distribution network switch action identification leveraging dynamic features of distributed energy resources. *IET Generation, Transmission & Distribution*, 13(14):3139–3147, 2019.

Roger C Dugan and Thomas E McDermott. An open source platform for collaborating on smart grid research. In *2011 IEEE power and energy society general meeting*, pp. 1–7. IEEE, 2011.

Afshin Ebtia, Mohsen Ghafouri, Mourad Debbabi, Marthe Kassouf, and Arash Mohammadi. Power distribution network topology detection using dual-graph structure graph neural network model. *IEEE Transactions on Smart Grid*, 16(2):1833–1850, 2024.

Energy Sector Management Assistance Program. Energy analytics for development: Big data for energy access, energy efficiency, and renewable energy. ESMAP Knowledge Series 027/17, World Bank, Washington, DC, 2017. URL https://openknowledge.worldbank.org/entities/publication/f588bc57-e143-5211-b275-fa4e71674fb8.

Bundesnetzagentur für Elektrizität and Telekommunikation Gas. Post und eisenbahnen„bericht zum zustand und ausbau der verteilernetze 2021 “. *Berichte der Verteilernetzbetreiber gem*, 14, 2022.

John J. Grainger and William D. Stevenson. *Power System Analysis*. McGraw-Hill, 1994. ISBN 9780070612938.

Benjamin Habib, Elvin Isufi, Ward van Breda, Arjen Jongepier, and Jochen L Cremer. Deep statistical solver for distribution system state estimation. *IEEE Transactions on Power Systems*, 39(2): 4039–4050, 2023.

Hendrik F Hamann, Blazhe Gjorgiev, Thomas Brunschwiler, Leonardo SA Martins, Alban Puech, Anna Varbella, Jonas Weiss, Juan Bernabe-Moreno, Alexandre Blondin Massé, Seong Lok Choi, et al. Foundation models for the electric power grid. *Joule*, 8(12):3245–3258, 2024.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pp. 2704–2710, 2020.

Yanhao Huang, Changqing Liu, Zichang Wang, and Qianhong Wu. A unified power-grid representation for reuse across network structures and computational tasks. *arXiv preprint arXiv:2609.07650*, 2026.

Burak Karabulut, Olayiwola Arowolo, Carlo Manna, Chris Develder, and Jochen L Cremer. Assessing the generalization of graph neural networks for fault location across increasing distributed energy resource penetration levels. *arXiv preprint arXiv:2607.29293*, 2026.

Vassilis Kekatos, Liang Zhang, Georgios B. Giannakis, and Ross Baldick. Fast localized voltage regulation in single-phase distribution grids. In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 725–730, 2015. doi: 10.1109/SmartGridComm.2015.7436387. URL https://engineering.purdue.edu/~kekatos/papers/SGC2015.pdf.

Yijiang Li, Zeeshan Memon, Hongwei Jin, Stefano Fenu, Keunju Song, Sunash B Sharma, Parfait Gasana, Hongseok Kim, Liang Zhao, and Kibaek Kim. Lumina: Foundation models for topology transferable acopf. *arXiv preprint arXiv:2603.04300*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Rahul Madbhavi, Balasubramaniam Natarajan, and Babji Srinivasan. Distribution system topology identification using graph neural networks. In *2023 IEEE PES GTD international conference and exposition (GTD)*, pp. 151–155. IEEE, 2023.

Kurt Mehlhorn. A faster approximation algorithm for the Steiner problem in graphs. *Information Processing Letters*, 27(3):125–128, 1988. doi: 10.1016/0020-0190(88)90066-X.

Overture Maps Foundation. Overture maps documentation: Transportation and buildings, 2026. URL `https://docs.overturemaps.org/`. Accessed September 22, 2026.

Bryan Palmintier, Carlos Mateo Domingo, Fernando Emilio Postigo Marcos, Tomas Gomez San Roman, Fernando de Cuadra, Nicolas Gensollen, Tarek Elgindy, and Pablo Duenas. Smart-ds synthetic electrical network data opendss models for sfo, gso, and aus. Open Energy Data Initiative (OEDI), National Renewable Energy Laboratory (NREL), https://data.openei.org/submissions/2981, 2020. URL `https://data.openei.org/submissions/2981`. Accessed: 2026-09-15.

Charilaos Papaioannou, Ioannis Tsantilas, Dimitris Giannakakos, Vasilis Michalakopoulos, Sotiris Pelekis, Vangelis Marinakis, Arsam Aryandoust, Antonello Monti, Ricardo J Bessa, Perdo P Vergara, et al. Mxgps: Multiplex graph transformers for a power grid foundation model. *arXiv preprint arXiv:2607.13763*, 2026.

Luis Piloto, Sofia Liguori, Sephora Madjiheurem, Miha Zgubic, Sean Lovett, Hamish Tomlinson, Sophie Elster, Chris Apps, and Sims Witherspoon. Canos: A fast and scalable neural ac-opf solver robust to n-1 perturbations. *arXiv preprint arXiv:2403.17660*, 2024.

Alban Puech, Matteo Mazzonelli, Tamara R Govindasamy, Mangaliso Mngomezulu, Héctor Maeso-García, Thomas Tolhurst, Javad Bayazi, Ali Moeini, Naomi Simumba, Celia Cintas, et al. Genco-a unified neural solver embedded in a development framework for steady-state grid analysis. *arXiv preprint arXiv:2608.09921*, 2026.

Ana Rivera Him, Anvita Bhagavathula, Alvaro Carbonero, and Priya Donti. PFΔ: A benchmark dataset for power flow under load, generation, and topology variations. *Advances in Neural Information Processing Systems*, 38, 2026.

Shammya Shananda Saha, Eran Schweitzer, Anna Scaglione, and Nathan G. Johnson. A framework for generating synthetic distribution feeders using OpenStreetMap. In *North American Power Symposium (NAPS)*, pp. 1–6, 2019. doi: 10.1109/NAPS46351.2019.9000187.

Kevin P Schneider, BA Mather, Bikash Chandra Pal, C-W Ten, Greg J Shirek, Hao Zhu, Jason C Fuller, José Luiz Rezende Pereira, Luis F Ochoa, Leandro Ramos de Araujo, et al. Analytic considerations and design basis for the ieee distribution test feeders. *IEEE Transactions on power systems*, 33(3):3181–3188, 2017.

Tarikua Taye, Bassam Mohamed, Lućıa Suarez-Ramon, and Pablo Arboleyá. A set of non-synthetic test systems of european lv rural, lv urban and hybrid mv/lv industrial distribution networks. *International Journal of Electrical Power & Energy Systems*, 158:109941, 2024.

Eric J. H. Wilson et al. End-use load profiles for the U.S. building stock: Methodology and results of model calibration, validation, and uncertainty quantification. Technical Report NREL/TP-5500-80889, National Renewable Energy Laboratory, 2022.

Weiwei Yang, Andrea Britto Mattos Lima, Thiago V Spina, Spencer Fowers, Baosen Zhang, and Chris White. Gridsfm: A foundation model for ac optimal power flow, 2026.

Ahmed Samir Zamzam and Nicholas D Sidiropoulos. Physics-aware neural networks for distribution system state estimation. *IEEE Transactions on Power Systems*, 35(6):4347–4356, 2020.

Yuhong Zhu, Yongzhi Zhou, Wei Wei, Peng Li, and Wenqi Huang. Gnns' generalization improvement for large-scale power system analysis based on physics-informed self-supervised pretraining. *IEEE Transactions on Power Systems*, 40(5):4145–4157, 2025.

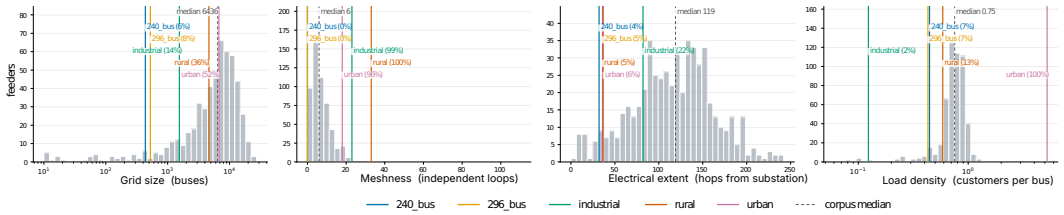

Figure 4: Structural characteristics of the reference corpus and the five frozen benchmark networks: bus count (a), independent loop count (b), electrical extent in hops from the substation (c), and customer density (d). Histograms show corpus distributions, colored vertical lines mark benchmark networks, and dashed lines indicate corpus medians. The table reports benchmark percentile ranks. Bus count and customer density use logarithmic horizontal axes.

A Data

A.1 Reference Data Sources

The reference arm of the corpus comprises publicly available distribution network models from four source families: (a) **NREL SMART-DS:** networks covering the Austin, San Francisco, and Greensboro regions, representing realistic synthetic U.S. distribution systems with multiple medium voltage classes and detailed secondary networks (Palmintier et al., 2020). (b) **Standard benchmark networks:** EPRI circuits [1] (`ckt5`, `ckt7`, and `ckt24`), the IEEE test feeder suite, including the IEEE European low voltage network (Schneider et al., 2017) and D-Suite networks (Deakin, 2025). (c) **Iowa State test systems:** 240- and 296-bus real distribution systems released as OpenDSS models with one year of smart meter data (Bu et al., 2019).[2] (d) **Nonsynthetic European systems:** utility derived networks based on GIS and smart meter data, including a hybrid MV/LV industrial network and large rural and urban LV networks, the largest containing 6,738 buses (Taye et al., 2024).

A.2 Structural Diversity of Reference Networks

Figure 4 characterizes the reference corpus by bus count, independent loop count, electrical extent, and customer density. The benchmark networks occupy distinct parts of these distributions: the Iowa networks have relatively small bus counts and electrical extents, whereas the European networks lie at the 96th-100th percentiles in independent loop count. The urban network is near the median in bus count but at the 100th percentile in customer density. These contrasts highlight structural differences that network size alone does not capture.

A.3 Synthetic Network Generation

Algorithm 1 summarizes the procedure described below. We define (D_{ref}) as the training subset of the reference network corpus. The component library, customer load records, and source specific or pooled planning statistics used by the generator are derived exclusively from this subset. No frozen benchmark network contributes to synthetic network generation.

We construct synthetic distribution networks and operating profiles by sampling geographic layouts, equipment conventions, and customer demand, then checking the resulting electrical behavior in OpenDSS (Dugan & McDermott, 2011). The procedure combines geographic feeder synthesis (Saha et al., 2019) with empirical component sampling and simulation guided repair. Component parameters, customer load records, and planning statistics are obtained from a reference corpus $\mathcal{D}_{\text{ref}}$. Operating profiles are drawn from the datasets described below. Each accepted case contains an electrical network and one 24-hour operating window.

[1] EPRI OpenDSS Test Circuits: `https://sourceforge.net/p/electricdss/code/HEAD/tree/trunk/Distrib/EPRITestCircuits/`

[2] Iowa Distribution Test Systems: `https://wzy.ece.iastate.edu/testsystem.html`

Geography and component sampling. Street segments and building footprints from Overture Maps (Overture Maps Foundation, 2026) provide candidate cable routes and customer locations. We sample a territory, choose a feeder head location, and snap it to the nearest street node. We keep the streets reachable from that node and select nearby buildings to serve. A component library built from $\mathcal{D}_{\text{ref}}$ supplies conductor impedances and ratings, transformer parameters, customer load records, and control settings. A sampled source dataset convention specifies customers per transformer, transformer utilization, secondary voltage classes, and device prevalence. When dataset specific statistics are unavailable, we use pooled values from $\mathcal{D}_{\text{ref}}$. Load records are sampled within single- and three-phase classes, with source weights tempered to reduce domination by larger datasets. Groups identified as placeholder or aggregate spot loads are excluded. Geography and the source dataset convention are sampled separately.

Network construction. We cluster customers spatially into service areas, each supplied by one distribution transformer. The initial cluster count is determined by the sampled number of customers per transformer from $\mathcal{D}_{\text{ref}}$. We increase this count when capacity, reach, or estimated voltage drop checks fail. Transformers are placed near each cluster's demand weighted centroid. Eligible clusters containing only single-phase customers receive center-tapped transformers with individual service drops. The remaining clusters use three-phase units and shortest path street routing. Approximate Steiner trees (Mehlhorn, 1988) connect transformer and substation locations at higher voltage levels. Downstream per phase demand determines conductor and transformer sizing. The construction also samples sectionalizing switches, normally open ties, capacitors, voltage controls, and rooftop PV. These choices vary network size, branching, phase composition, and equipment. At this stage, each customer has a nominal demand. Time series are assigned next.

Operating profiles and load coincidence. Nominal active and reactive demands are sampled from the component library and rescaled to a sampled network demand level. We assign residential ResStock profiles to single-phase loads and commercial ComStock profiles to three-phase loads (Wilson et al., 2022). Profiles are drawn from a sampled climate group, with annual peak ranks matched to nominal load sizes. Let $\bar{p}_i$ be customer i's fixed nominal active demand and $a_i(h)$ its assigned reference building demand profile in kW, where $h \in \{1, \ldots, 8760\}$ indexes the hours of the year. We define the synthetic customer's nominal demand schedule as

$$p_i(h) = \bar{p}_i \frac{a_i(h)}{\max_{h'} a_i(h')}, \qquad \eta_c = \frac{\max_h \sum_{i \in c} p_i(h)}{\sum_{i \in c} \bar{p}_i}, \tag{5}$$

where c is the set of customers served by one transformer. The reference profile supplies the temporal pattern, while $\bar{p}_i$ sets its annual peak. The coincidence factor η_c accounts for customers reaching their individual peaks at different times. A common date from a sampled season defines the 24-hour window. Reactive power and PV output profiles are drawn from SMART-DS (Palmintier et al., 2020). Reactive profiles are normalized by their annual peaks and scaled by nominal reactive demand. PV profiles combine a common solar trajectory with site level variation. Geography, load climate, and solar region are sampled separately rather than jointly matched.

Snapshot simulation. We first construct a static OpenDSS model from the sampled network and nominal customer demands. At this stage, power flow is solved in *snapshot mode*: one operating point is evaluated at a time. Annual load profiles are used to compute the coincidence factors η_c, but are not yet applied as a time stepped simulation.

An operating input $\mathbf{U} = (\mathbf{p}, \mathbf{q}, \boldsymbol{\alpha})$ specifies the nominal active and reactive demands of all customers and the availability of DER generation. A DER multiplier $\alpha_j = 0$ disables generation at device j, while $\alpha_j = 1$ specifies full availability under its configured device model. With $c(i)$ denoting the transformer service area containing customer i, the two stress inputs are

$$\begin{aligned} \mathbf{U}_{\text{A}} &= \left((\eta_{c(i)} \bar{p}_i)_i,\ (\eta_{c(i)} \bar{q}_i)_i,\ \mathbf{0} \right), \\ \mathbf{U}_{\text{B}} &= \left(\lambda \bar{\mathbf{p}},\ \lambda \bar{\mathbf{q}},\ \mathbf{1} \right), \qquad \lambda \sim \text{Uniform}(0.2, 0.4), \end{aligned} \tag{6}$$

where $\bar{q}_i$ is nominal reactive demand. Thus, $\mathbf{U}_{\text{A}}$ represents coincident peak demand with DER generation disabled, while $\mathbf{U}_{\text{B}}$ represents reduced demand with full DER availability. The multiplier λ is drawn once per proposed network and retained during repair.

The call $Y = \mathrm{SnapshotPF}(\mathcal{G}, \mathbf{c}; \mathbf{U})$ applies one such input to the network and runs an unbalanced OpenDSS snapshot solve. Here $\mathbf{c}$ contains equipment parameters and configurations. The returned diagnostics Y include convergence status, bus voltages, line currents, and line and transformer loading ratios. We solve the two operating points separately to obtain Y_{A} and Y_{B}.

Repairs to the proposed network. Repairs are offline edits to the synthetic simulator model during data generation. The editable variables are customer phase assignments, transformer taps and ratings, conductor multiplicities, capacitor and regulator placement and ratings, and DER inverter modes and output limits. Let $\mathcal{R}_{\mathrm{C}}$ denote the ordered snapshot repair schedule and $\mathcal{R}_{\mathrm{D}}$ the daily repair schedule in Table 6. Each scheduled operation r has a trigger based on the current diagnostics and a repetition limit. When triggered, it edits the affected variables in $(\mathcal{G}, \mathbf{c})$. For example, a line overload can trigger an additional parallel conductor, whereas overvoltage can trigger a tap change or DER curtailment. Repairs for overloading are enabled only when the thermal enforcement flag b is set. After every edit, the relevant operating points are solved again: improving one can worsen another.

Table 6: Repair actions used during synthetic network generation. Actions apply to the affected service areas or branches, subject to the sampled design's equipment options. Snapshot repairs also seek voltage headroom beyond the final acceptance limits.

Trigger	Change to the proposed model	Repetition limit
Low snapshot voltage	Rebalance single-phase loads where connections permit; adjust transformer taps; add capacitors or regulators; add parallel conductors along affected paths; select a larger transformer.	Ordered passes; up to 4 conductor reinforcement and 2 transformer upgrade rounds.
High snapshot voltage	Enable inverter volt-var control, back off voltage boosting taps where the low voltage condition permits, reinforce conductors, or reduce affected DER output limits by 20%.	Up to 5 rounds.
Snapshot overload, with thermal enforcement	Add parallel conductor circuits, choose a larger transformer, add parallel substation units, or increase regulator capacity.	Up to 3 rounds.
Daily voltage violation	Adjust taps or reinforce affected paths for undervoltage; use inverter control, tap backoff, phase rebalancing, or DER curtailment for overvoltage.	Up to 2 undervoltage and 8 overvoltage rounds.

Daily simulation and acceptance. After snapshot repair, we attach the selected 24-hour load and DER generation profiles. The call $Y_{1:24} = \mathrm{DailyPF}(\mathcal{G}, \mathbf{c}; \mathbf{U}_{1:24})$ switches OpenDSS to daily mode and performs 24 sequential one-hour solves, returning diagnostics at each step. The daily repairs in Table 6 address violations identified along this trajectory. After each change, both snapshot conditions and the complete day are checked again. The load profiles and sampled multipliers remain fixed during these repairs.

Acceptance requires power flow convergence, a minimum voltage of at least 0.95 p.u. in snapshot A, and voltages within $[0.95, 1.05]$ p.u. in snapshot B and throughout the sampled day. Daily voltage checks omit values at or below 0.05 p.u. to exclude floating tie nodes. An entirely collapsed step is rejected. For each proposed network, thermal enforcement is enabled with probability 0.7 and remains fixed during repair. When enabled, line and transformer ratings are checked in both snapshots, and line current ratings are checked at every timestep of the day. Cases that still fail the applicable checks after the allowed repairs are rejected.

The predicate Accept applies the convergence, voltage, and conditional rating checks described above to the final snapshot and daily diagnostics. The flag b controls thermal enforcement. The variables λ and b remain fixed throughout repair.

Outputs and scope. The generator saves self contained OpenDSS models, load and solar time series, sampled parameters, repair histories, and acceptance results. Observation noise and missingness are specified separately by the sensor policies in Appendix D. Repair and rejection change the distribution of accepted cases relative to the sampling priors.

Algorithm 1: Synthetic network generation

Input: Reference corpus $\mathcal{D}_{\mathrm{ref}}$; street and building maps; load and DER generation profiles $\mathcal{P}$; seeds $\mathcal{S}$; repair schedules $\mathcal{R}_{\mathrm{C}}, \mathcal{R}_{\mathrm{D}}$ defined in the text and Table 6.
Output: Accepted networks with 24 validated hourly operating points.

Generate component library $\mathcal{L}$ and occurrence statistics from $\mathcal{D}_{\mathrm{ref}}$ using OpenDSS
// An unrecovered error or timeout records failure and skips the current seed.
foreach $s \in \mathcal{S}$ **do**
  Initialize the random generator with seed s
  Sample territory, feeder head location, and equipment nameplate parameters
  Draw $\lambda \sim \mathrm{Uniform}(0.2, 0.4)$ and $b \sim \mathrm{Bernoulli}(0.7)$
  // Stage 1: construct and repair a snapshot model.
  Keep streets reachable from the feeder head and select nearby customer locations
  Assign customer phase classes, then sample and rescale nominal demands $(\bar{p}_i, \bar{q}_i)$ from $\mathcal{L}$
  Sample DER presence, capacities, and inverter settings from $\mathcal{D}_{\mathrm{ref}}$
  Cluster customers, site transformers, and route service and upstream connections
  Assign phases and size equipment
  Add sectionalizing switches and normally open ties to obtain $(\mathcal{G}, \mathbf{c})$
  Build the OpenDSS model in snapshot mode with nominal demands $(\bar{\mathbf{p}}, \bar{\mathbf{q}})$
  Use annual profiles from $\mathcal{P}$ to compute η_c in Eq. equation 5
  Form the snapshot inputs $\mathbf{U}_{\mathrm{A}}, \mathbf{U}_{\mathrm{B}}$ in Eq. equation 6
  $Y_{\mathrm{A}} \leftarrow \mathrm{SnapshotPF}(\mathcal{G}, \mathbf{c}; \mathbf{U}_{\mathrm{A}})$; $Y_{\mathrm{B}} \leftarrow \mathrm{SnapshotPF}(\mathcal{G}, \mathbf{c}; \mathbf{U}_{\mathrm{B}})$
  foreach *triggered operation r in $\mathcal{R}_{\mathrm{C}}$, within its repetition limit* **do**
    $(\mathcal{G}, \mathbf{c}) \leftarrow r(\mathcal{G}, \mathbf{c}; Y_{\mathrm{A}}, Y_{\mathrm{B}}, b)$
    Solve both snapshots again to update $Y_{\mathrm{A}}, Y_{\mathrm{B}}$
  // Stage 2: attach time series and validate one day.
  Select a common date and construct load and DER generation inputs $\mathbf{U}_{1:24}$
  $Y_{1:24} \leftarrow \mathrm{DailyPF}(\mathcal{G}, \mathbf{c}; \mathbf{U}_{1:24})$
  foreach *triggered operation r in $\mathcal{R}_{\mathrm{D}}$, within its repetition limit* **do**
    $(\mathcal{G}, \mathbf{c}) \leftarrow r(\mathcal{G}, \mathbf{c}; Y_{\mathrm{A}}, Y_{\mathrm{B}}, Y_{1:24}, b)$
    Solve both snapshots and all 24 hourly steps again to update $Y_{\mathrm{A}}, Y_{\mathrm{B}}, Y_{1:24}$
  if $\mathrm{Accept}(Y_{\mathrm{A}}, Y_{\mathrm{B}}, Y_{1:24}; b)$ **then**
    Save $(\mathcal{G}, \mathbf{c}, \mathbf{U}_{1:24})$ as an accepted case
  else
    Record rejection
  Record seed, sampled parameters, repair history, and final diagnostics

B Task Overview

We jointly infer electrical states, connectivity attributes, and fault events from a network's equipment graph and partial, noisy measurements. All four tasks share the same inputs. Labels and target validity masks are used only for supervision and evaluation.

B.1 Network Graph and Partial Observations

We represent each network as a typed graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \tau_{\mathcal{V}}, \tau_{\mathcal{E}}), \tag{7}$$

with bus, consumer, substation, capacitor, and DER nodes. Relations distinguish lines, transformers, reactors, and device to bus attachments. All switchable branches remain in the graph regardless of their operating state: the equipment inventory is known, while switch states must be inferred.

Each sample contains $T = 24$ consecutive hourly observations, indexed by $\mathcal{T} = \{0, \ldots, T-1\}$. Entity $u \in \mathcal{V} \cup \mathcal{E}$ has static attributes $\mathbf{s}_u$, temporal measurements $\mathbf{X}_u \in \mathbb{R}^{T \times d_{\tau(u)}}$, and an observation mask $\mathbf{M}_u \in \{0,1\}^{T \times d_{\tau(u)}}$. Static attributes describe equipment properties. Measurements include available voltage, power, current, and feeder head telemetry. Missing readings are zero filled and masked to distinguish them from physical zeros.

The model receives

$$\mathcal{O} = \{(\mathbf{M}_u \odot \mathbf{X}_u, \mathbf{M}_u, \mathbf{s}_u) : u \in \mathcal{V} \cup \mathcal{E}\}, \tag{8}$$

and jointly predicts

$$f_{\boldsymbol{\theta}}(\mathcal{G}, \mathcal{O}) = \big(\widehat{\mathbf{V}}, \widehat{\boldsymbol{\Theta}}, \widehat{\mathbf{P}}^{\mathrm{phase}}, \widehat{\mathbf{p}}^{\mathrm{sw}}, \widehat{\mathbf{p}}^{\mathrm{fault}}, \widehat{\mathbf{p}}^{\mathrm{loc}}\big). \tag{9}$$

Sensor policies vary measurement placement, density, noise, and missingness without changing the task definitions.

B.2 State Estimation

For each bus, timestep, and phase in $\mathcal{P} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$, the model predicts voltage magnitude and angle:

$$\widehat{\mathbf{V}}, \widehat{\boldsymbol{\Theta}} \in \mathbb{R}^{|\mathcal{V}_{\mathrm{bus}}| \times T \times 3}. \tag{10}$$

A validity mask $\mathbf{M}^{\mathrm{state}}$ of the same shape excludes absent phases and invalid targets. The task reconstructs electrical states over the input window, with supervision and evaluation restricted to valid (bus, timestep, phase) entries.

B.3 Phase Attribution

Phase attribution applies to eligible single-phase consumers whose connection phase is not determined trivially by the local structure. For each eligible consumer j, the model predicts a distribution $\widehat{\mathbf{p}}_j^{\mathrm{phase}} \in [0,1]^3$ over the canonical A-B-C ordering and assigns the phase with the highest probability. The physical phase label remains constant throughout the window.

Separate masks track structural eligibility and the availability of at least one customer voltage observation, supporting evaluation over either all eligible consumers or the observed subset.

B.4 switch state Inference

For each switchable line $e \in \mathcal{E}_{\mathrm{sw}}$, the target is $y_e^{\mathrm{sw}} = 1$ for open and 0 for closed. Equipment metadata identifies switch locations. The model predicts open state probabilities from network structure and available electrical measurements, with operating state labels withheld. Switch states remain constant within each window.

B.5 Fault Classification and Conditional Localization

Fault diagnosis combines graph level classification with equipment localization. The model predicts a distribution over

$$\mathcal{C}_{\mathrm{fault}} = \{\mathrm{normal}, \mathrm{LG}, \mathrm{LL}, \mathrm{LLG}, \mathrm{LLL}, \mathrm{LLLG}\}, \tag{11}$$

where normal denotes no active fault.

For faulted samples, the location target identifies an element of the joint candidate set

$$\mathcal{A}_{\mathrm{fault}} = \mathcal{V}_{\mathrm{bus}} \,\dot\cup\, \mathcal{E}_{\mathrm{line}} \,\dot\cup\, \mathcal{E}_{\mathrm{xfmr}}. \tag{12}$$

The model scores all candidate buses, lines, and transformers together and predicts $\widehat{y}^{\mathrm{loc}} = \arg\max_{a \in \mathcal{A}_{\mathrm{fault}}} \widehat{p}_a^{\mathrm{loc}}$. Normal samples have no location target and contribute no localization loss.

The current benchmark injects one fault at the final observed timestep, $t = T - 1$. Diagnosis therefore uses the pre event history and initial fault response, without post fault recovery measurements. Arbitrary onset times and sustained faults are outside this evaluation.

B.6 Joint Task Applicability

A task applicability vector $\mathbf{a}_n \in \{0,1\}^4$ specifies which tasks have valid targets for sample n. Combined with entity level validity masks, it allows one window to supervise multiple tasks. These supervision masks remain separate from the observation masks in $\mathcal{O}$ and are never model inputs. Every sample uses the same $(\mathcal{G}, \mathcal{O})$ interface without an explicit task identifier.

Table 7: Prediction problems and temporal scope.

Task	ML formulation	Prediction scope
State estimation (SE)	Continuous regression	Magnitude and angle per (bus, phase, timestep)
Phase attribution	Three-class classification	One A/B/C assignment per eligible consumer and window
switch state inference	Binary classification	One open/closed state per switch and window
Fault diagnosis	Six-class classification; conditional localization	Normal/fault class per window; equipment location for faulted windows

C Model Details

C.1 Input Schema and Supervision Masks

Tables 8 summarize the node, relation, and auxiliary inputs. Static attributes are shared across a sample's $T = 24$ hourly steps, while temporal measurements retain their time dimension. Phase resolved measurements follow the fixed (A, B, C) ordering. Consumer measurements are scalar channels without an explicit consumer phase label. Capacitor and distributed energy resource (DER) nodes are included when present in the network. Features marked as optional are controlled by model configuration.

Switch eligible lines remain in the structural graph regardless of their open/closed state. Customer assignments to transformers identify serving equipment. They are distinct from the customer phase labels being predicted. Transformer winding mappings likewise describe equipment metadata.

For entity u, let $\mathbf{x}_{u,t}$ denote its measurement vector and $\mathbf{m}_{u,t} \in \{0,1\}^{d_u}$ its observability mask. The model receives transformed measurements together with their masks:

$$\widetilde{\mathbf{x}}_{u,t} = [\text{where}\,(\mathbf{m}_{u,t}, \psi_u(\mathbf{x}_{u,t}), \mathbf{0}) \parallel \mathbf{m}_{u,t}], \qquad t = 1, \ldots, T, \tag{13}$$

where ψ_u denotes the configured channelwise transformation and $\parallel$ denotes concatenation. Masks reflect sensor placement and measurement availability. Channels sharing a measurement source may share a mask. Unavailable values are zero filled after transformation. These observation masks are separate from target validity and task applicability masks used for supervision.

Optional electrical distance encodings supplement the listed attributes with topology derived position information. The nominal angle transformation is described in Appendix C.4, and the positional encoding in Appendix C.5.

For static attributes $\mathbf{s}_u$ and transformed, masked inputs $\widetilde{\mathbf{x}}_{u,t}$, type specific node encoders produce

$$\mathbf{H}_u^{(0)} = E_{\tau(u)}^{\text{node}}(\mathbf{s}_u, \widetilde{\mathbf{x}}_{u,1:T}) \in \mathbb{R}^{T \times d}. \tag{14}$$

Analogous edge encoders process physical branch inputs. Parameters are shared across entities of the same type and across networks. Connectivity only relations have no separate attribute encoder. Static features are broadcast across the window, while measurements and masks retain their time indices.

Observation masks identify available inputs. Supervision masks identify valid targets and applicable tasks. An unobserved voltage can therefore remain a supervised target when its ground truth is available during training.

C.2 Communication Edge Construction

The communication graph augments the electrical graph with two relation types. Uplinks connect the substation to each distinct distribution transformer primary bus. Downlinks connect a customer to the secondary bus of its serving transformer. Both relations support bidirectional message passing and have parameters distinct from the physical relations.

For a customer with a valid transformer association, the augmented graph contains the path

$$\text{substation} \leftrightarrow \text{primary bus} \leftrightarrow \text{secondary bus} \leftrightarrow \text{customer}. \tag{15}$$

The middle connection is the physical transformer relation. The first and last connections are shortcuts. Thus, the customer lies within three message passing hops of the feeder head, irrespective of the length of the corresponding electrical route. The shortcuts change information exchange within the model while preserving the physical graph. Their construction uses topology and registry information without adding measurement channels.

Table 8: Model inputs for nodes (a), relations (b), and auxiliary stores (c). Every temporal measurement has an observability mask. Connectivity only relations provide endpoint indices and relation identity without separate attribute vectors. DT denotes distribution transformer. Dashes indicate no direct temporal input. Auxiliary entries describe their inputs and role across the last two columns.

Type	Static inputs / connectivity	Temporal inputs
(a) Nodes		
Bus	Base voltage; source bus indicator; normalized coordinates and coordinate validity flag.	Per phase voltage magnitude and angle; active power P and reactive power Q.
Consumer	One-hot metering tier encoding.	Customer voltage channels for phase attribution and voltage derived from the metering tier; active and reactive power.
Substation	Positive- and zero-sequence source resistance and reactance; three-phase and single-phase short-circuit strengths; base voltage; voltage setpoint; source angle; frequency.	Feeder head voltage and active/reactive power when feeder head fusion is enabled.
Capacitor	Voltage and reactive power ratings; switching step count; delta connection and controller indicators; active step fraction.	–
DER	Device kind encoding for PV, generator, or storage; active/reactive power ratings; delta connection indicator.	–
(b) Relations		
Line bus–bus	Switch, fuse, and recloser indicators; length; normal/emergency current ratings; sequence capacitances and impedances; phase count. Full phase impedance matrices are optional.	Per phase current magnitude; active/reactive power flows with extended measurements.
Transformer bus–bus	Regulator, fuse, and recloser indicators; apparent power rating; turns ratio; winding count; no-load loss, magnetizing current, and interwinding reactance parameters; phase counts. Winding connections and phase presence indicators are optional.	Terminal voltage, angle, current, and power embeddings with extended measurements.
Reactor bus–bus	Switch, fuse, and recloser indicators; phase count; optional resistance and reactance.	–
Device attachments	Links to buses from consumers (service), substations (source), capacitors, and DERs.	–
Communication (optional)	Links from substations to DT primary buses and from consumers to DT secondary buses.	–
Phase head links (optional)	Links from consumers to their serving transformer's secondary bus for the phase prediction head.	–
Terminal attachments	Links from terminals to buses and transformer objects.	–
(c) Auxiliary stores: inputs and role		
Transformer terminal	Winding index; per phase voltage magnitude and angle, current magnitude, active/reactive power, and observability masks.	
Transformer object	Aggregates encoded terminal measurements for fusion into the corresponding transformer edge representation.	
Grid / feeder head	Reference voltage, feeder head active/reactive power, and masks; observed tap summaries and sampling cadence features.	
Power flow	Feeder head current magnitude and mask in the extended measurement path.	

C.3 Learning with Communication Edges

Model and observations. Let $N \geq 1$ and let buses $0, 1, \ldots, N$ form a tree rooted at the fixed voltage source. Number parents before children and index each branch by its child. Let $d(i, j)$ count physical branches between buses i and j. All expectations are over $s \sim \mathcal{D}$, including operating conditions and measurement noise. Snapshot arguments are suppressed below. In this appendix, $\mathbf{p}, \mathbf{q}$ denote the ground truth injections in (1), using a net-injection sign convention. At each bus, active power supplied to the grid contributes positively to p_i, while active power consumption contributes negatively. Likewise, reactive power injection contributes positively to q_i and absorption negatively. Set $\delta\mathbf{p} = \mathbf{p} - \mathbb{E}[\mathbf{p}]$, $\delta\mathbf{q} = \mathbf{q} - \mathbb{E}[\mathbf{q}]$, and $\mathbf{y} = \mathbf{V}^{\odot 2} - \mathbb{E}[\mathbf{V}^{\odot 2}]$. Since V_0 is fixed, subtracting the expectation of (1) gives $\mathbf{y} = 2\mathbf{R}\delta\mathbf{p} + 2\mathbf{X}\delta\mathbf{q}$. For $\mathbf{z} = (\delta\mathbf{p}^\mathsf{T}, \delta\mathbf{q}^\mathsf{T})^\mathsf{T}$, assume $\mathbf{\Sigma} = \mathbb{E}[\mathbf{z}\mathbf{z}^\mathsf{T}] \succ 0$. The centered measurements are $\widetilde{\mathbf{z}} = \mathbf{z} + \boldsymbol{\varepsilon} = (\widetilde{\mathbf{z}}_p^\mathsf{T}, \widetilde{\mathbf{z}}_q^\mathsf{T})^\mathsf{T}$, where $\widetilde{\mathbf{z}}_p, \widetilde{\mathbf{z}}_q \in \mathbb{R}^N$ are its active and reactive measurement blocks. The sensor error is independent of $\mathbf{z}$ and satisfies $\mathbb{E}\boldsymbol{\varepsilon} = \mathbf{0}$ and $\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\mathsf{T}] = \tau\mathbf{\Sigma}$, so $\mathbb{E}[\widetilde{\mathbf{z}}\widetilde{\mathbf{z}}^\mathsf{T}] = (1+\tau)\mathbf{\Sigma}$. Each bus receives only its own noisy injection components, with no global measurement aggregates. Voltage targets are noiseless. Both injections and sensor errors may be correlated across buses.

Let $J_{ij} = 1$ when j is the nonroot parent of i, and zero otherwise. The reduced branch–bus incidence matrix, oriented positively at the parent, is $\mathbf{A} = \mathbf{J} - \mathbf{I}$. Let the physical depth be $h = \max_i d(0, i)$. Then $\mathbf{J}^h = \mathbf{0}$ and

$$\mathbf{T} = -\mathbf{A}^{-1} = \sum_{k=0}^{h-1} \mathbf{J}^k, \qquad \mathbf{R} = \mathbf{T}\,\mathrm{diag}(\mathbf{r})\mathbf{T}^\mathsf{T}, \qquad \mathbf{X} = \mathbf{T}\,\mathrm{diag}(\mathbf{x})\mathbf{T}^\mathsf{T}. \tag{16}$$

The inverse identity follows by multiplying the sum by $\mathbf{I} - \mathbf{J}$. These factorizations yield the radial LinDistFlow relation (1) (Kekatos et al., 2015; Deka et al., 2024), assumed exact here. The vectors $\mathbf{r}, \mathbf{x}$ collect branch resistances $r_e > 0$ and reactances $x_e \geq 0$. Direct multiplication gives $R_{ij} = \sum_{e \in \mathcal{P}_i \cap \mathcal{P}_j} r_e$ and $X_{ij} = \sum_{e \in \mathcal{P}_i \cap \mathcal{P}_j} x_e$, where $\mathcal{P}_i$ is the root path of bus i. An injection at j affects squared voltage at i when R_{ij} or X_{ij} is nonzero.

Learner and communication. The model learns unconstrained branch coefficients $\mathbf{w} = (\mathbf{w}_r^\mathsf{T}, \mathbf{w}_x^\mathsf{T})^\mathsf{T}$:

$$\widehat{\mathbf{y}}_\mathbf{w} = \mathbf{T}\big[\mathrm{diag}(\mathbf{w}_r)\mathbf{T}^\mathsf{T}\widetilde{\mathbf{z}}_p + \mathrm{diag}(\mathbf{w}_x)\mathbf{T}^\mathsf{T}\widetilde{\mathbf{z}}_q\big] = \mathbf{\Phi}(\widetilde{\mathbf{z}})\mathbf{w}, \tag{17}$$

For any $\mathbf{a} = (\mathbf{a}_p^\mathsf{T}, \mathbf{a}_q^\mathsf{T})^\mathsf{T}$, $\mathbf{\Phi}(\mathbf{a}) = \mathbf{T}[\,\mathrm{diag}(\mathbf{T}^\mathsf{T}\mathbf{a}_p) \quad \mathrm{diag}(\mathbf{T}^\mathsf{T}\mathbf{a}_q)\,]$, which is linear in its argument. The physical coefficients $\mathbf{w}_* = (2\mathbf{r}^\mathsf{T}, 2\mathbf{x}^\mathsf{T})^\mathsf{T}$ satisfy $\mathbf{y} = \mathbf{\Phi}(\mathbf{z})\mathbf{w}_*$. The factor two comes from squared voltage magnitudes. Coefficients are shared across snapshots and affected outputs, but remain branch specific. The main text prediction is $\widehat{\mathbf{V}^{\odot 2}}_\mathbf{w} = \mathbb{E}[\mathbf{V}^{\odot 2}] + \widehat{\mathbf{y}}_\mathbf{w}$, an affine map of the raw injections for fixed weights.

Set $K = \lceil \log_2 h \rceil$. Include bidirectional links to every nonroot ancestor at physical distances 2^k, $0 \leq k < K$. Then

$$\prod_{k=0}^{K-1} (\mathbf{I} + \mathbf{J}^{2^k}) = \sum_{j=0}^{2^K - 1} \mathbf{J}^j = \mathbf{T}. \tag{18}$$

Each exponent occurs once by binary expansion, and powers at least h vanish. The factors commute. Applying their transposes to $[\,\widetilde{\mathbf{z}}_p \quad \widetilde{\mathbf{z}}_q\,]$ computes observed subtree injection sums in K upward rounds. Bus e multiplies the two sums by $w_{r,e}, w_{x,e}$ and adds them. Applying the original factors accumulates these contributions along root paths in K downward rounds, giving (17). Messages have two coordinates upward and one downward, and all aggregation weights are fixed at one. For $h = 1$, the products are empty and communication is unnecessary.

Proof of Proposition 1. Centering (2) gives $\mathcal{L}(\mathbf{w}) = \frac{1}{2}\mathbb{E}\,\|\widehat{\mathbf{y}}_\mathbf{w} - \mathbf{y}\|^2$. Write $\mathbf{M}_* = [\,2\mathbf{R} \quad 2\mathbf{X}\,]$, so $\mathbf{y} = \mathbf{M}_*\mathbf{z}$, and put $\alpha = (1+\tau)^{-1}$. Define the residual $\boldsymbol{\xi} = \mathbf{y} - \alpha\mathbf{M}_*\widetilde{\mathbf{z}} = \mathbf{M}_*[(1-\alpha)\mathbf{z} - \alpha\boldsymbol{\varepsilon}]$. The noise assumptions give

$$\mathbb{E}[\boldsymbol{\xi}\widetilde{\mathbf{z}}^\mathsf{T}] = \mathbf{M}_*[(1-\alpha)\mathbf{\Sigma} - \alpha\tau\mathbf{\Sigma}] = \mathbf{0}, \tag{19}$$

$$\nu_\tau := \mathbb{E}\,\|\boldsymbol{\xi}\|^2 = \big[(1-\alpha)^2 + \alpha^2\tau\big]\,\mathrm{tr}(\mathbf{M}_*\mathbf{\Sigma}\mathbf{M}_*^\mathsf{T}) = (1-\alpha)\mathcal{E}_\mathrm{H}(0). \tag{20}$$

The last equality uses $\mathcal{E}_{\mathrm{H}}(0) = \mathbb{E}\,\|\mathbf{y}\|^2 = \mathrm{tr}(\mathbf{M}_*\boldsymbol{\Sigma}\mathbf{M}_*^{\mathsf{T}})$. For any linear predictor $\mathbf{F}\widetilde{\mathbf{z}}$, expand its error as $(\mathbf{F} - \alpha\mathbf{M}_*)\widetilde{\mathbf{z}} - \boldsymbol{\xi}$. The cross term vanishes by (19), yielding

$$\mathbb{E}\,\|\mathbf{F}\widetilde{\mathbf{z}} - \mathbf{y}\|^2 = \nu_\tau + (1+\tau)\,\mathrm{tr}\big[(\mathbf{F} - \alpha\mathbf{M}_*)\boldsymbol{\Sigma}(\mathbf{F} - \alpha\mathbf{M}_*)^{\mathsf{T}}\big]. \tag{21}$$

Thus ν_τ is the minimum error among unrestricted linear predictors. Since $\alpha\mathbf{M}_*$ is realized by $\mathbf{w}_\tau = \alpha\mathbf{w}_*$, the same minimum is attainable by our learner. No Gaussian assumption is used.

Let $\mathbf{H}_0 = \mathbb{E}[\boldsymbol{\Phi}(\mathbf{z})^{\mathsf{T}}\boldsymbol{\Phi}(\mathbf{z})]$ and $\mathbf{H}_\tau = \mathbb{E}[\boldsymbol{\Phi}(\widetilde{\mathbf{z}})^{\mathsf{T}}\boldsymbol{\Phi}(\widetilde{\mathbf{z}})]$. Each entry of $\boldsymbol{\Phi}$ is linear in its argument, so zero cross covariance between $\mathbf{z}, \boldsymbol{\varepsilon}$ and the proportional noise covariance give $\mathbf{H}_\tau = (1+\tau)\mathbf{H}_0$. Equation (21) implies

$$\mathcal{L}(\mathbf{w}) = \frac{\nu_\tau}{2} + \frac{1}{2}(\mathbf{w} - \mathbf{w}_\tau)^{\mathsf{T}}\mathbf{H}_\tau(\mathbf{w} - \mathbf{w}_\tau). \tag{22}$$

For $\mathbf{u} = (\mathbf{u}_r^{\mathsf{T}}, \mathbf{u}_x^{\mathsf{T}})^{\mathsf{T}}$, set $\mathbf{M}(\mathbf{u}) = [\,\mathbf{T}\,\mathrm{diag}(\mathbf{u}_r)\mathbf{T}^{\mathsf{T}} \quad \mathbf{T}\,\mathrm{diag}(\mathbf{u}_x)\mathbf{T}^{\mathsf{T}}\,]$. Then $\mathbf{u}^{\mathsf{T}}\mathbf{H}_\tau\mathbf{u} = (1+\tau)\,\mathrm{tr}(\mathbf{M}(\mathbf{u})\boldsymbol{\Sigma}\mathbf{M}(\mathbf{u})^{\mathsf{T}})$. Since $\mathbf{T}$ is invertible, $\mathbf{M}(\mathbf{u}) \neq \mathbf{0}$ for $\mathbf{u} \neq \mathbf{0}$. Positive definiteness of $\boldsymbol{\Sigma}$ makes this trace strictly positive. Hence $\mathbf{H}_\tau \succ 0$ and all its eigenvalues are positive.

For $\mathbf{e}_n = \mathbf{w}^{(n)} - \mathbf{w}_\tau$, gradient descent gives $\mathbf{e}_n = (\mathbf{I} - \eta'\mathbf{H}_\tau)^n\mathbf{e}_0$, with $\mathbf{e}_0 = -\mathbf{w}_\tau$. Diagonalize $\mathbf{H}_\tau = \mathbf{U}\,\mathrm{diag}(\lambda_j)\mathbf{U}^{\mathsf{T}}$, where $\mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{I}$, and set $\mathbf{c} = \mathbf{U}^{\mathsf{T}}\mathbf{e}_0$. Then

$$\mathcal{E}_{\mathrm{H}}(n) - \nu_\tau = \sum_j \lambda_j(1 - \eta'\lambda_j)^{2n}c_j^2 =: Q_n. \tag{23}$$

This is an equality for the population loss, with no bound on the Hessian spectrum substituted. At initialization, $Q_0 = \mathbf{e}_0^{\mathsf{T}}\mathbf{H}_\tau\mathbf{e}_0 = \mathcal{E}_{\mathrm{H}}(0) - \nu_\tau = \alpha\mathcal{E}_{\mathrm{H}}(0)$. For $0 < \eta' < 2/\max_j \lambda_j$, every factor $|1 - \eta'\lambda_j|$ is less than one, so Q_n is nonincreasing and tends to zero.

For bus i and integer budget $\ell \geq 0$, define the accessible and inaccessible injection coordinates

$$\mathcal{A}_i = \{j, N+j : 1 \leq j \leq N,\ d(i,j) \leq \ell\}, \qquad \mathcal{U}_i = \{1, \ldots, 2N\} \setminus \mathcal{A}_i. \tag{24}$$

The two coordinates for each bus correspond to active and reactive injections. Subscripts on $\boldsymbol{\Sigma}$ select the indicated rows and columns, in a fixed order. Write $\mathbf{m}_i = (\mathbf{M}_*)_{i,:} \in \mathbb{R}^{1\times 2N}$ for row i. Induction over communication rounds shows that every physical linear predictor has $\mathbf{F}_{i,\mathcal{U}_i} = \mathbf{0}$. For nonempty $\mathcal{U}_i$, set

$$\mathbf{C}_i = \boldsymbol{\Sigma}_{\mathcal{U}_i\mathcal{U}_i} - \boldsymbol{\Sigma}_{\mathcal{U}_i\mathcal{A}_i}\boldsymbol{\Sigma}_{\mathcal{A}_i\mathcal{A}_i}^{-1}\boldsymbol{\Sigma}_{\mathcal{A}_i\mathcal{U}_i}, \qquad C_\ell = \sum_{i:\mathcal{U}_i\neq\emptyset} \mathbf{m}_{i,\mathcal{U}_i}\mathbf{C}_i\mathbf{m}_{i,\mathcal{U}_i}^{\mathsf{T}}. \tag{25}$$

The principal block being inverted is positive definite because $\boldsymbol{\Sigma} \succ 0$ and $\mathcal{A}_i$ includes bus i. An empty inaccessible set contributes zero.

To minimize (21) subject to locality, let $\mathbf{d}_i = \mathbf{F}_{i,\mathcal{A}_i} - \alpha\mathbf{m}_{i,\mathcal{A}_i}$. The corresponding row of $\mathbf{F} - \alpha\mathbf{M}_*$ is $[\,\mathbf{d}_i, -\alpha\mathbf{m}_{i,\mathcal{U}_i}\,]$ in the $(\mathcal{A}_i, \mathcal{U}_i)$ ordering. Its quadratic contribution is

$$\begin{aligned}&\mathbf{d}_i\boldsymbol{\Sigma}_{\mathcal{A}_i\mathcal{A}_i}\mathbf{d}_i^{\mathsf{T}} - 2\alpha\mathbf{d}_i\boldsymbol{\Sigma}_{\mathcal{A}_i\mathcal{U}_i}\mathbf{m}_{i,\mathcal{U}_i}^{\mathsf{T}} + \alpha^2\mathbf{m}_{i,\mathcal{U}_i}\boldsymbol{\Sigma}_{\mathcal{U}_i\mathcal{U}_i}\mathbf{m}_{i,\mathcal{U}_i}^{\mathsf{T}}\\ &\quad= \boldsymbol{\delta}_i\boldsymbol{\Sigma}_{\mathcal{A}_i\mathcal{A}_i}\boldsymbol{\delta}_i^{\mathsf{T}} + \alpha^2\mathbf{m}_{i,\mathcal{U}_i}\mathbf{C}_i\mathbf{m}_{i,\mathcal{U}_i}^{\mathsf{T}}, \qquad \boldsymbol{\delta}_i = \mathbf{F}_{i,\mathcal{A}_i} - \mathbf{F}^\star_{i,\mathcal{A}_i},\end{aligned} \tag{26}$$

where

$$\mathbf{F}^\star_{i,\mathcal{A}_i} = \alpha\left(\mathbf{m}_{i,\mathcal{A}_i} + \mathbf{m}_{i,\mathcal{U}_i}\boldsymbol{\Sigma}_{\mathcal{U}_i\mathcal{A}_i}\boldsymbol{\Sigma}_{\mathcal{A}_i\mathcal{A}_i}^{-1}\right), \qquad \mathbf{F}^\star_{i,\mathcal{U}_i} = \mathbf{0}. \tag{27}$$

If $\mathcal{U}_i$ is empty, the second term in parentheses is absent and $\mathbf{F}^\star_{i,:} = \alpha\mathbf{m}_i$. Using $(1+\tau)\alpha^2 = \alpha$ and summing the completed squares gives

$$\mathcal{E}_{\mathrm{P}}(\mathbf{F}) = \nu_\tau + \alpha C_\ell + (1+\tau)\sum_i \boldsymbol{\delta}_i\boldsymbol{\Sigma}_{\mathcal{A}_i\mathcal{A}_i}\boldsymbol{\delta}_i^{\mathsf{T}}. \tag{28}$$

Thus the minimum over all local linear maps is $\mathcal{E}^\star_{\mathrm{P}}(\ell) = \nu_\tau + \alpha C_\ell$, attained uniquely by $\mathbf{F}^\star$ on the allowed coordinates. This minimum is also attainable with ℓ physical communication rounds when message width and node specific coefficients are unrestricted. To see this, place each node's two measurements in their own coordinates of a $2N$-vector. In the first round, send that vector to each neighbor. In later rounds, forward the preceding round's vectors to all neighbors except the edge from which each arrived. On a tree, a measurement reaches another bus exactly once, after its physical distance in rounds. Summing the received vectors over rounds and retaining the bus's own

coordinates therefore supplies exactly $\widetilde{\mathbf{z}}_{\mathcal{A}_i}$. All routing and aggregation are linear. Bus i then applies row $\mathbf{F}^\star_{i,\mathcal{A}_i}$. The root can relay messages with zero initial coordinates. This argument establishes attainability in the unrestricted comparison class, not in a prescribed fixed width or shared weight neural architecture.

Combining (23) and (28) yields the exact decomposition

$$\boxed{\mathcal{E}_{\mathrm{P}}(\mathbf{F}) - \mathcal{E}_{\mathrm{H}}(n) = \frac{C_\ell}{1+\tau} - Q_n + (1+\tau)\sum_i \boldsymbol{\delta}_i \boldsymbol{\Sigma}_{\mathcal{A}_i\mathcal{A}_i} \boldsymbol{\delta}_i^{\mathsf{T}}.} \tag{29}$$

The three terms describe inaccessible information, remaining GD error, and the physical predictor's excess error above its local optimum. Dropping only the last, nonnegative term gives a sharp lower bound for each fixed grid, covariance, and iterate: equality holds exactly when $\mathbf{F} = \mathbf{F}^\star$. Setting $\mathbf{F} = \mathbf{F}^\star$ and substituting the exact expression for Q_n proves Proposition 1. Every other physical linear predictor has at least this error gap. Linearity is stated after centering. Restoring the true output mean gives affine predictors of the raw observations. Any additional constant bias contributes its squared norm to the risk. □

Why the Schur complement is the relevant covariance. The residual after the best linear prediction of inaccessible injections is

$$\boldsymbol{\zeta}_i = \mathbf{z}_{\mathcal{U}_i} - \boldsymbol{\Sigma}_{\mathcal{U}_i\mathcal{A}_i} \boldsymbol{\Sigma}^{-1}_{\mathcal{A}_i\mathcal{A}_i} \mathbf{z}_{\mathcal{A}_i}. \tag{30}$$

Direct expansion gives $\mathbb{E}[\boldsymbol{\zeta}_i \mathbf{z}_{\mathcal{A}_i}^{\mathsf{T}}] = \mathbf{0}$ and $\mathbb{E}[\boldsymbol{\zeta}_i \boldsymbol{\zeta}_i^{\mathsf{T}}] = \mathbf{C}_i$. Orthogonality makes every other linear reconstruction add a nonnegative quadratic error. Thus C_ℓ sums the residual variances after weighting by the corresponding voltage sensitivities. No Gaussian assumption is needed. For a general distribution, $\mathbf{C}_i$ is a linear residual covariance, not necessarily a conditional covariance.

Positivity and the exact step threshold. For a nonzero column vector $\mathbf{b}$ on a nonempty $\mathcal{U}_i$, set $\mathbf{a}_* = -\boldsymbol{\Sigma}^{-1}_{\mathcal{A}_i\mathcal{A}_i} \boldsymbol{\Sigma}_{\mathcal{A}_i\mathcal{U}_i} \mathbf{b}$. Expanding the quadratic form gives

$$\mathbf{b}^{\mathsf{T}} \mathbf{C}_i \mathbf{b} = \begin{bmatrix} \mathbf{a}_* \\ \mathbf{b} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \boldsymbol{\Sigma}_{\mathcal{A}_i\mathcal{A}_i} & \boldsymbol{\Sigma}_{\mathcal{A}_i\mathcal{U}_i} \\ \boldsymbol{\Sigma}_{\mathcal{U}_i\mathcal{A}_i} & \boldsymbol{\Sigma}_{\mathcal{U}_i\mathcal{U}_i} \end{bmatrix} \begin{bmatrix} \mathbf{a}_* \\ \mathbf{b} \end{bmatrix} > 0. \tag{31}$$

Thus $\mathbf{C}_i \succ 0$, so $C_\ell > 0$ exactly when some buses i, j more than ℓ physical hops apart satisfy $R_{ij}^2 + X_{ij}^2 > 0$. Increasing ℓ enlarges the feasible predictor class, so C_ℓ cannot increase. Choosing the zero predictor in the noiseless problem gives $0 \le C_\ell \le \mathcal{E}_{\mathrm{H}}(0)$.

For $C_\ell > 0$, the exact first GD step that beats every physical linear predictor is

$$n_\sharp = \min\{n \in \mathbb{N}_0 : Q_n < C_\ell/(1+\tau)\}. \tag{32}$$

Since Q_n is nonincreasing and tends to zero under the stated step size condition, $n_\sharp$ is finite, and every later iterate also satisfies the strict comparison. If $C_\ell = 0$, an optimal physical predictor already attains ν_τ, so no hierarchical linear predictor can strictly outperform it.

Effect of the noise level. The base step size η is constant across GD iterations. Since $\mathbf{H}_\tau = (1+\tau)\mathbf{H}_0$ and $\mathbf{w}_\tau = \alpha \mathbf{w}_*$, the choice $\eta' = \eta/(1+\tau)$ gives $\mathbf{I} - \eta' \mathbf{H}_\tau = \mathbf{I} - \eta \mathbf{H}_0$. With zero initialization, the noisy parameter error is therefore α times the noiseless error at every iteration. Its squared Hessian norm satisfies $Q_n(\tau) = \alpha Q_n(0)$. Because C_ℓ is independent of τ, the exact optimal comparator error gap in (3) equals its noiseless value divided by $1+\tau$. Thus the exact crossing time is unchanged by τ under this matched step scaling. The common noise floor still increases with τ.

Resistance uncertainty. Keep topology, injection covariance, and noise level fixed. Let $\boldsymbol{\kappa} = (\mathbf{r}^{\mathsf{T}}, \mathbf{x}^{\mathsf{T}})^{\mathsf{T}}$ range over the compact box $\mathcal{B}$ defined by $r_e \in [\underline{r}_e, \overline{r}_e]$ and $x_e \in [\underline{x}_e, \overline{x}_e]$, with $\underline{r}_e > 0$ and $\underline{x}_e \ge 0$. The Hessian $\mathbf{H}_\tau$ and its eigenbasis remain fixed, while C_ℓ and the initial error coordinates depend on $\boldsymbol{\kappa}$. For a fixed admissible step size, Proposition 1 gives the exact worst-case gap

$$\Gamma_{\ell,n} = \min_{\boldsymbol{\kappa} \in \mathcal{B}} \left\{ \frac{C_\ell(\boldsymbol{\kappa})}{1+\tau} - Q_n(\boldsymbol{\kappa}) \right\}. \tag{33}$$

Continuity and compactness ensure that the minimum is attained. Thus $\Gamma_{\ell,n} > 0$ holds exactly when the hierarchical learner strictly outperforms every physical linear predictor at every impedance assignment in the box. The learners are retrained from zero for each assignment. Both terms are quadratic in $\boldsymbol{\kappa}$: the voltage sensitivities and initial parameter error are linear in the impedances, while the residual covariances and Hessian are fixed. Their difference need not be convex, so this exact uniform comparison is a potentially nonconvex quadratic minimization over the box. Substituting only the lower impedance endpoints is not generally valid.

Scope. The result requires the prescribed links in both directions and noisy injection coverage at every bus. It concerns population loss under exact LinDistFlow, with clean voltage targets. For a general noise covariance $\boldsymbol{\Omega}$, the best linear response is $\mathbf{M}_*\boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\boldsymbol{\Omega})^{-1}$, obtained from the normal equation $\mathbf{F}(\boldsymbol{\Sigma}+\boldsymbol{\Omega}) = \mathbf{M}_*\boldsymbol{\Sigma}$. It need not retain the learner's path factorization. Thus the proportional covariance assumption is substantive. The floor ν_τ is optimal among linear predictors, not necessarily among nonlinear predictors. Gaussianity is not assumed. For finite samples the same proof requires the corresponding empirical moment identities, not merely a positive definite empirical covariance. It does not by itself establish a test error guarantee. The exact physical optimum uses population covariances and unrestricted local linear coefficients. If physical communication already implements (17), the two implementations have identical losses and gradient iterates.

C.4 Nominal Angle Reference

We separate static phase conventions from operating angle deviations using a nominal reference $\theta^{\mathrm{nom}}i, p$ for each (bus, phase) pair. The reference is derived from network topology, source configuration, and transformer metadata, without using simulated voltage trajectories. A validity mask ri, p identifies pairs with a defined reference. Source phase angles initialize propagation through the network. Lines and reactors preserve phase identity with zero nominal shift, leaving angle drops that depend on operating conditions to the residual prediction. Transformer mappings follow winding conductor order and connection metadata:

- Matching winding connection types assign zero shift to corresponding conductor positions.
- Mixed delta-wye connections with three matched conductor positions introduce a shift of $\pm\pi/6$, determined by the configured lead/lag convention.
- Single-conductor winding mappings assign zero shift to matching positions and π to opposite positions, accounting for polarity and center-tapped secondary legs.

Mappings include only phases present on the branch and its endpoint buses. Unsupported mappings are omitted, and conflicting mappings for the same phase pair are discarded. Reverse traversal inverts the phase mapping and negates its shift. All angles below are in radians, with

$$\mathrm{wrap}(x) = ((x+\pi) \bmod 2\pi) - \pi. \tag{34}$$

For valid (bus, phase) pairs, observed angles and targets use the same residual coordinates. The state head predicts $\widehat{\Delta\theta}i, p, t$ and recovers absolute angles through

$$\Delta\theta i, p, t = \mathrm{wrap}\left(\theta_{i,p,t} - \theta^{\mathrm{nom}}i, p\right), \ \widehat{\theta}i, p, t \quad = \mathrm{wrap}\left(\theta^{\mathrm{nom}}i, p + \widehat{\Delta\theta}i, p, t\right). \tag{35}$$

Because prediction and target share the reference,

$$\mathrm{wrap}(\widehat{\theta}i, p, t - \theta i, p, t) = \mathrm{wrap}(\widehat{\Delta\theta}i, p, t - \Delta\theta i, p, t). \tag{36}$$

This parameterization removes static phase offsets from the regression target while preserving circular angle error. Figure 5 illustrates the nominal references and corresponding operating angle residuals. Appendix C.7.1 specifies the state decoder.

C.5 Electrical Distance Positional Encoding

We construct positional descriptors on the full structural graph of lines, transformers, and reactors. All switches are treated as closed, independently of their operational state. The descriptors therefore do not use switch state labels and are computed once per network and physical topology variant.

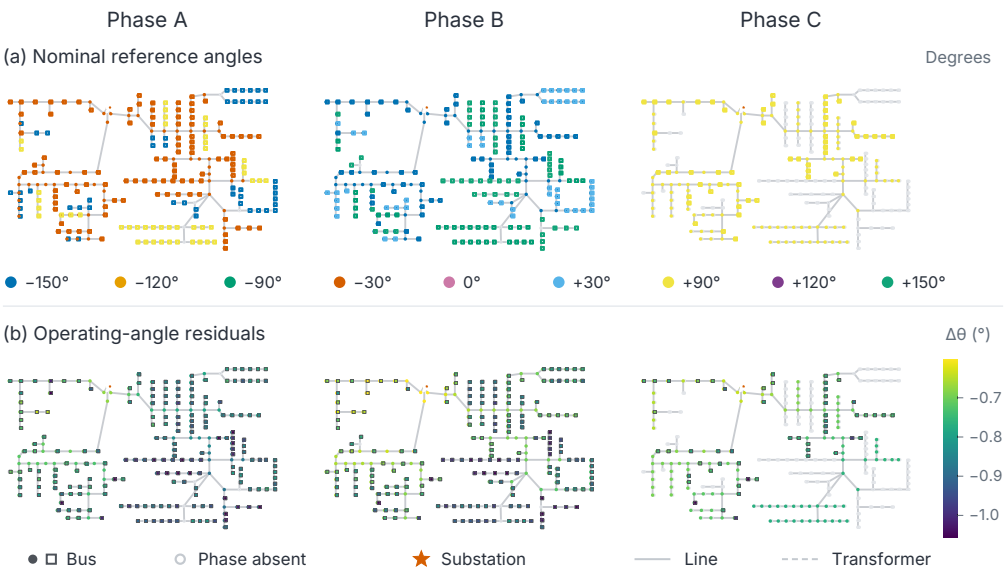

Figure 5: Nominal angle parameterization on an example network. Columns show phases A, B, and C. (a) Nominal references θ^{nom} encode source and transformer phase conventions. (b) Residuals $\Delta\theta = \mathrm{wrap}(\theta - \theta^{\mathrm{nom}})$ capture operating angle deviations. Nominal values use categorical colors, and residuals share a continuous scale. Light gray markers indicate absent phases, and solid and dashed edges denote lines and transformers, respectively. Angles are shown in degrees.

For each branch e, resistance and reactance are converted to per unit values using a fixed 1 MVA system base and the local voltage base. Transformer impedances use the primary side base. Line impedances are obtained from scalar sequence style summaries of their phase impedance matrices. The Dijkstra edge weight is

$$w_e = \sqrt{(r_e^{\mathrm{pu}})^2 + (x_e^{\mathrm{pu}})^2}. \tag{37}$$

For bus i, the selected substation rooted path is

$$\mathcal{P}_i = \arg\min_{\mathcal{P}: s \rightsquigarrow i} \sum_{e \in \mathcal{P}} w_e. \tag{38}$$

Along this path, we record the hop count ℓ_i, accumulated resistance R_i, accumulated reactance X_i, transformer count n_i^{tr}, cumulative phase shift ψ_i, and number of nonzero shift transformer crossings n_i^{shift}. Transformer shifts respect traversal direction. A transformer contributes a phase shift only when its resolved phase mappings agree on a single value. Otherwise its scalar contribution is zero.

The nine-dimensional descriptor is

$$\begin{aligned}\mathbf{a}_i^{\mathrm{pos}} = \big[&\ell_i/D_{\max},\ \log(1+\ell_i),\ \log(1+R_i/R_0),\ \log(1+X_i/X_0),\\ &\log(1+n_i^{\mathrm{tr}}),\ \log(V_i^{\mathrm{base}}/V_s^{\mathrm{base}}),\ \sin\psi_i,\ \cos\psi_i,\ \log(1+n_i^{\mathrm{shift}})\big]^\top.\end{aligned} \tag{39}$$

We use corpus calibrated scales $R_0 = 0.66$, $X_0 = 0.54$, and $D_{\max} = 57$. Unreachable buses receive zero valued raw path statistics.

A shared MLP and LayerNorm map the descriptor to

$$\mathbf{p}_i = \mathrm{LN}\left(\mathrm{MLP}_{\mathrm{PE}}(\mathbf{a}_i^{\mathrm{pos}})\right) \in \mathbb{R}^{16}. \tag{40}$$

This embedding is concatenated with each bus's static attributes and masked measurement inputs before the bus encoder:

$$\mathbf{h}_{i,t}^{(0)} = \mathrm{Enc}_{\mathrm{bus}}\left([\mathbf{s}_i \parallel \widetilde{\mathbf{x}}_{i,t} \parallel \mathbf{p}_i]\right). \tag{41}$$

The same $\mathbf{p}_i$ is used at every timestep.

Table 9: Mycelium configuration. Parameter count includes the task heads.

Component	Setting
Transformer layers	8
Hidden dimension	128
Attention heads	4
Node type FFN hidden width	256
Temporal window	24 hourly steps
Positional encoding	$9 \rightarrow 16$ dimensions
Normalization	Pre-LN and final LN
Dropout	0.1
Total parameters	Approximately 12.2M

C.6 Heterogeneous Graph Transformer

C.6.1 Residual Updates

The attention output is followed by a residual update and a type specific feedforward sublayer:

$$\mathbf{u}_{i,t}^{(\ell)} = \mathbf{h}_{i,t}^{(\ell)} + \text{Dropout}\left(\mathbf{m}_{i,t}^{(\ell)}\right), \tag{42}$$

$$\mathbf{h}_{i,t}^{(\ell+1)} = \mathbf{u}_{i,t}^{(\ell)} + \text{Dropout}\left[\text{FFN}_{\tau(i)}^{(\ell)}\left(\text{LN}_{\tau(i)}^{(\ell,\text{ffn})}(\mathbf{u}_{i,t}^{(\ell)})\right)\right]. \tag{43}$$

Each FFN has the form

$$\text{FFN}_{\tau}^{(\ell)}(\mathbf{z}) = \mathbf{W}_{2,\tau}^{(\ell)}\,\text{Dropout}\left(\text{GELU}\left(\mathbf{W}_{1,\tau}^{(\ell)}\mathbf{z} + \mathbf{b}_{1,\tau}^{(\ell)}\right)\right) + \mathbf{b}_{2,\tau}^{(\ell)}, \tag{44}$$

with intermediate width $2d$. The attention operator has no additional internal root projection in its output, leaving a single residual pathway around each sublayer. After the final layer, a type specific LayerNorm produces the representations supplied to the task readouts.

C.6.2 Preserving the Temporal Dimension

Type specific input encoders produce

$$\mathbf{H}_{\tau}^{(0)} \in \mathbb{R}^{|\mathcal{V}_\tau| \times T \times d}.$$

Static attributes are broadcast across the window, while measurements and observability masks retain their timestep indices. Every spatial layer preserves this shape:

$$\mathbf{H}_{:,t}^{(\ell+1)} = F^{(\ell)}\left(\mathbf{H}_{:,t}^{(\ell)}, \mathbf{E}_{:,t}, \mathcal{G}\right), \qquad t = 1, \ldots, T. \tag{45}$$

The same $F^{(\ell)}$ is applied at every timestep, with no cross time attention or temporal pooling inside the spatial stack. We implement these operations by batching the snapshots as disjoint graphs and restoring the explicit time axis afterward.

Consequently, the backbone outputs a sequence of spatially informed representations for every node. Temporal modeling and aggregation are performed by the task specific readouts, allowing tasks to use different summaries of the same observation window.

Let $\mathbf{h}_{i,t} \in \mathbb{R}^d$ denote a node's final backbone representation and $\mathbf{e}_{ij,t} \in \mathbb{R}^d$ an encoded branch, with $d = 128$. Each task has independently parameterized temporal readouts.

C.6.3 Context Construction

For branch $e = (i, j)$, the readout input combines both endpoints and the branch embedding:

$$\mathbf{x}_{e,t} = [\mathbf{h}_{i,t} \| \mathbf{h}_{j,t} \| \mathbf{e}_{ij,t}]. \tag{46}$$

The network representation combines encoded feeder head measurements $\mathbf{u}_{\mathcal{G},t}$, mean pooled bus embeddings, and the substation embedding:

$$\mathbf{g}_{\mathcal{G},t} = f_{\mathcal{G}}\left([\mathbf{u}_{\mathcal{G},t} \parallel \text{mean}_{i \in \mathcal{V}_{\text{bus}}}\,\mathbf{h}_{i,t} \parallel \mathbf{h}_{s,t}]\right). \tag{47}$$

For consumer c attached to bus $b(c)$, we use

$$\mathbf{x}_{c,t} = [\mathbf{h}_{c,t} \| \mathbf{h}_{b(c),t} \| \mathbf{g}_{\mathcal{G},t}]. \tag{48}$$

C.7 Task Readout

C.7.1 State Estimation

The state estimation readout applies four residual blocks of causal dilated temporal convolutions, each with kernel size three and dilation $\delta_b \in \{1, 2, 4, 8\}$:

$$\mathbf{z}_{i,t}^{(0)} = \mathbf{h}_{i,t}, \tag{49}$$

$$\mathbf{z}_{i,t}^{(b+1)} = \mathrm{LN}_b\left(\mathbf{z}_{i,t}^{(b)} + \mathrm{Dropout}\left(\mathrm{SiLU}\left(\mathbf{b}_b + \sum_{q=0}^{2} \mathbf{W}_{b,q}\mathbf{z}_{i,t-q\delta_b}^{(b)}\right)\right)\right). \tag{50}$$

Left zero padding preserves sequence length and ensures that the readout accesses only current and preceding backbone representations. This establishes causality of the temporal refiner. End-to-end causality also depends on preprocessing and input encoding.

A shared linear map produces three magnitude and three angle residual outputs per bus and timestep:

$$[\mathbf{a}_{i,t} \| \mathbf{b}_{i,t}] = \mathbf{W}_{\mathrm{SE}}\mathbf{z}_{i,t}^{(4)} + \mathbf{b}_{\mathrm{SE}}, \qquad \mathbf{a}_{i,t}, \mathbf{b}_{i,t} \in \mathbb{R}^3. \tag{51}$$

The outputs are converted to per unit voltage magnitudes and deviations from the nominal angles defined in Appendix C.4:

$$\widehat{|V|}_{i,p,t}^{\mathrm{pu}} = 1 + \frac{a_{i,p,t}}{100}, \qquad \widehat{\Delta\theta}_{i,p,t} = \pi \tanh(b_{i,p,t}). \tag{52}$$

We recover absolute angles by adding the nominal reference,

$$\widehat{\theta}_{i,p,t} = \mathrm{wrap}\left(\theta_{i,p}^{\mathrm{nom}} + \widehat{\Delta\theta}_{i,p,t}\right). \tag{53}$$

Observed angle inputs and angle targets are likewise expressed as wrapped deviations from $\theta_{i,p}^{\mathrm{nom}}$.

For a qualifying center-tapped secondary, the angle of one leg is derived from the other:

$$\widehat{\theta}_{i,\mathrm{derived},t} = \mathrm{wrap}\left(\widehat{\theta}_{i,\mathrm{reference},t} + \pi\right). \tag{54}$$

Here, "reference" denotes the other physical leg. Validity masks restrict supervision to existing (bus, phase) targets with a defined nominal angle reference.

C.7.2 Phase Attribution

Let $m_{c,t}$ indicate availability of the consumer's phase attribution voltage observation. Temporal attention uses this indicator as an additional scoring feature:

$$\beta_{c,t}^{\mathrm{PA}} = \mathrm{softmax}_t\left(a_{\mathrm{PA}}([\mathbf{x}_{c,t} \| m_{c,t}])\right). \tag{55}$$

The readout pools projected values and predicts a physical phase:

$$\mathbf{z}_c^{\mathrm{PA}} = \sum_{t=1}^{T} \beta_{c,t}^{\mathrm{PA}} f_{\mathrm{PA}}(\mathbf{x}_{c,t}), \tag{56}$$

$$\mathbf{p}_c^{\mathrm{PA}} = \mathrm{softmax}\left(g_{\mathrm{PA}}(\mathbf{z}_c^{\mathrm{PA}})\right). \tag{57}$$

The output classes are A, B, C. All timesteps participate in the softmax, including those without a local voltage observation, because spatial message passing can supply information from other entities.

C.7.3 Switch State Prediction

For each line, $m_{e,t}$ indicates whether any phase current measurement is available. A separate attention scorer produces

$$\beta_{e,t}^{\mathrm{SW}} = \mathrm{softmax}_t\left(a_{\mathrm{SW}}([\mathbf{x}_{e,t} \| m_{e,t}])\right). \tag{58}$$

We fuse three complementary summaries:

$$\mathbf{z}_e^{\mathrm{SW}} = f_{\mathrm{SW}}\left(\left[\sum_{t=1}^{T} \beta_{e,t}^{\mathrm{SW}}\mathbf{x}_{e,t} \parallel \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{e,t} \parallel \max_t \mathbf{x}_{e,t}\right]\right). \tag{59}$$

The maximum is elementwise, and all three summaries use the complete sequence. The open state probability is

$$p_e^{\text{open}} = \sigma\left(g_{\text{SW}}(\mathbf{z}_e^{\text{SW}})\right). \tag{60}$$

One label is predicted for the window, with supervision restricted to switch bearing lines.

C.7.4 Fault Diagnosis

Fault windows place the event at timestep T. For an input sequence $\mathbf{x}_{1:T}$, the readout constructs an attended history:

$$\gamma_t = \text{softmax}_{t<T}\left(a_{\text{hist}}(\mathbf{x}_t + \mathbf{p}_t)\right), \tag{61}$$

$$\mathbf{r} = \sum_{t=1}^{T-1} \gamma_t \mathbf{x}_t, \tag{62}$$

where $\mathbf{p}_t$ is a learned history position embedding. It then compares the endpoint with this history:

$$\mathcal{R}(\mathbf{x}_{1:T}) = f_{\text{event}}\left([\mathbf{x}_T \| \mathbf{r} \| (\mathbf{x}_T - \mathbf{r}) \| |\mathbf{x}_T - \mathbf{r}|]\right). \tag{63}$$

The candidate set is $\mathcal{C}(\mathcal{G}) = \mathcal{V}_{\text{bus}} \cup \mathcal{E}_{\text{line}} \cup \mathcal{E}_{\text{transformer}}$. Bus candidates use their node sequences directly. Line and transformer contexts share an affine projection into the same d-dimensional space:

$$\mathbf{q}_{c,t} = \begin{cases} \mathbf{h}_{c,t}, & c \in \mathcal{V}_{\text{bus}}, \\ \mathbf{W}_{\text{edge}}\mathbf{x}_{c,t} + \mathbf{b}_{\text{edge}}, & c \in \mathcal{E}_{\text{line}} \cup \mathcal{E}_{\text{transformer}}. \end{cases} \tag{64}$$

A shared candidate readout and a separately parameterized network readout produce

$$\mathbf{z}_c^{\text{F}} = \mathcal{R}_{\text{cand}}(\mathbf{q}_{c,1:T}), \qquad \mathbf{z}_{\mathcal{G}}^{\text{F}} = \mathcal{R}_{\text{graph}}(\mathbf{g}_{\mathcal{G},1:T}^{\text{F}}). \tag{65}$$

The fault specific network sequence is constructed with whole window grid summaries, including tap statistics, supplied only to its final timestep.

All candidate types use the same location scorer:

$$s_c = g_{\text{loc}}([\mathbf{z}_c^{\text{F}} \| \mathbf{z}_{\mathcal{G}}^{\text{F}}]), \qquad \widehat{c} = \arg\max_{c \in \mathcal{C}(\mathcal{G})} s_c. \tag{66}$$

The winning candidate determines both equipment type and location. Because the scorer is shared, its output set follows the current network rather than a fixed vocabulary of training locations. Localization supervision applies to applicable faulted samples.

Fault classification combines the network readout with elementwise maximum evidence across all candidate types:

$$\mathbf{p}^{\text{F}} = \text{softmax}\left(g_{\text{cls}}\left(\left[\mathbf{z}_{\mathcal{G}}^{\text{F}} \parallel \max_{c \in \mathcal{C}(\mathcal{G})} \mathbf{z}_c^{\text{F}}\right]\right)\right). \tag{67}$$

The six classes are normal operation, line-to-ground (LG), line-to-line (LL), double-line-to-ground (LLG), three-phase (LLL), and three-phase-to-ground (LLLG) faults. Fault detection follows from the distinction between the normal class and the five fault classes. The network representation already incorporates mean pooled bus context, while the candidate maximum retains localized evidence.

The phase, switch, fault class, and location heads each use a 128-dimensional hidden layer with SiLU activation and dropout 0.1, followed by an output linear layer. The state decoder is a single linear layer following the causal temporal refiner.

C.8 Multi-task Objective and Supervision

The joint objective uses unit coefficients for the four task losses:

$$\mathcal{L} = \mathcal{L}_{\text{SE}} + \mathcal{L}_{\text{PA}} + \mathcal{L}_{\text{SW}} + \mathcal{L}_{\text{F}}. \tag{68}$$

Here, SE, PA, SW, and F denote the phase, switch, and fault losses used in the main text. Unit coefficients apply after each task's internal scaling and reduction. They do not imply equal loss magnitudes or gradient contributions.

Reduction over valid targets For an entity level task q, let $M^q_{g,u}$ indicate whether target u in network window g is supervised, and let $\mathcal{B}_q$ contain windows with at least one supervised target. We use

$$\mathrm{Avg}_q(\ell) = \frac{1}{|\mathcal{B}_q|} \sum_{g \in \mathcal{B}_q} \frac{\sum_u M^q_{g,u} \ell_{g,u}}{\sum_u M^q_{g,u}}. \tag{69}$$

Thus, each applicable network window contributes its mean valid target loss, preventing larger networks from receiving greater weight solely because they contain more supervised entities. Windows without valid targets are omitted from the corresponding average.

State estimation Over valid (bus, timestep, phase) targets, we minimize

$$\mathcal{L}_{\mathrm{SE}} = \frac{1}{2} \mathrm{Avg}_{\mathrm{SE}} \left[\rho\Big(\lambda_V(|\widehat{V}|^{\mathrm{pu}} - |V|^{\mathrm{pu}})\Big) + \rho\Big(\lambda_\theta \,\mathrm{wrap}(\widehat{\theta} - \theta)\Big) \right]. \tag{70}$$

Here ρ is Smooth-L_1 with unit transition threshold. Angles are in radians, and scaling precedes application of ρ. The constants λ_V and λ_θ control the magnitude and angle residual scales, respectively.

Phase attribution and switch state Phase attribution uses three-class cross entropy on structurally eligible consumers with at least one observed phase voltage reading in the window:

$$\mathcal{L}_{\mathrm{PA}} = \mathrm{Avg}_{\mathrm{PA}}[-\log p_c(y_c)]. \tag{71}$$

switch state prediction uses unweighted binary cross entropy on switch bearing lines, with $y_e = 1$ for OPEN:

$$\mathcal{L}_{\mathrm{SW}} = \mathrm{Avg}_{\mathrm{SW}}[-y_e \log p_e - (1 - y_e) \log(1 - p_e)]. \tag{72}$$

Fault diagnosis The classification head predicts normal operation or one of five fault types. With inverse frequency weights $w_k \propto f_k^{-1}$, where f_k is the configured training corpus frequency, its loss is

$$\mathcal{L}_{\mathrm{cls}} = -\frac{\sum_{g \in \mathcal{B}} w_{y_g} \log p_g(y_g)}{\sum_{g \in \mathcal{B}} w_{y_g}}. \tag{73}$$

For faulted windows with valid location targets $\mathcal{B}_{\mathrm{F}}$, localization applies softmax cross entropy over the bus, line, and transformer candidates $\mathcal{C}(g)$:

$$\mathcal{L}_{\mathrm{loc}} = -\frac{1}{|\mathcal{B}_{\mathrm{F}}|} \sum_{g \in \mathcal{B}_{\mathrm{F}}} \log \frac{\exp(s_{g,c^\star_g})}{\sum_{c \in \mathcal{C}(g)} \exp(s_{g,c})}. \tag{74}$$

We combine the two components when the batch contains a valid fault location target:

$$\mathcal{L}_{\mathrm{F}} = \begin{cases} \frac{1}{2}(\mathcal{L}_{\mathrm{cls}} + \mathcal{L}_{\mathrm{loc}}), & |\mathcal{B}_{\mathrm{F}}| > 0, \\ \mathcal{L}_{\mathrm{cls}}, & \text{otherwise.} \end{cases} \tag{75}$$

D SENSOR POLICIES

Every electrical case is rendered under 7 named sensor policies. A policy fixes (i) *which* positions carry sensors (structural placement, drawn once per network per policy and persisted), and (ii) the noise and missingness applied to the resulting readings at load time (redrawn from persisted per sample seeds, so every sample is reproducible).

Coverage levels. Coverage fractions are calibrated to utility practice: The six degraded policies apply three fixed tiers under two placement strategies, while `clean` is the fully observed, noiseless reference.

Tier	Bus voltage	DT meter (joint $V+I$)	Branch flow	Customer P/Q	of which Q
sparse	1%	60%	0.5%	10%	20%
medium	5%	90%	2%	30%	20%
dense	15%	100%	5%	70%	20%
`clean`	100%	100%	100%	100%	100%

Table 10: Structural coverage per tier. A "DT meter" is one physical device: its terminal bus voltage and transformer edge current channels are always granted together, never sampled independently. Customer coverage is a binary AMI/unobserved split. Among AMI customers, reactive power is reported by the stated fraction. The substation reference voltage and feeder head P/Q are observed under every policy.

Placement algorithms. The `noisy_missing_{sparse,medium,dense}` policies place sensors *uniformly at random* within each eligible pool (plain buses, DT terminal buses, branches, customers) at the tier's target count. The `observability_ami_{sparse,medium,dense}` policies use the same target counts but place voltage and branch sensors by greedy farthest point (k-center) selection on the network's hop distance graph: sensors are added iteratively at the bus currently at maximum hop distance from its nearest existing sensor, minimizing the worst-case distance to the nearest sensor rather than allowing random clustering. The DT pool is filled first and the plain bus pass is seeded from it. Customer AMI assignment is clustered by serving transformer: entire DT service areas receive AMI together, modeling area wide utility rollout programs rather than an independent coin flip per meter.

Noise and missingness. For all policies except `clean`, each observed reading receives multiplicative Gaussian noise $x \mapsto x\,(1+\varepsilon)$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, with σ per channel: voltage magnitude 0.2%, voltage angle 0.02° (additive), P/Q injections 2%, currents 1%. Regulator tap readback is exact. Each reading is independently dropped with per channel probability: voltage 1%, injections 5%, currents 5%. `clean` sets all noise and missingness to zero.

E Further Experiment Results

E.1 State Estimation

$$\arg\min_{\mathbf{x}_t} \sum_{m \in \mathcal{O}t} \frac{[z_{m,t} - h_m(\mathbf{x}_t; \mathcal{G}_t)]^2}{\sigma_m^2}, \tag{76}$$

where $\mathcal{O}_t$ is the set of available measurements, h_m is the measurement function, and σ_m is the corresponding noise scale. We evaluate voltage magnitudes and angles on the same valid (bus, phase, timestep) entries used for the learned estimators.

WLS receives the exact per phase network model, operational switch states, timestep specific tap positions, measurement mapping, and simulator noise scales. It also uses known zero injection constraints and a substation angle reference. It is therefore an oracle-assisted reference, rather than a baseline operating under Mycelium's information constraints. Table 11 reports results and per-case runtimes across the five frozen benchmark networks. WLS performs well in several industrial and rural settings, whereas Mycelium obtains lower errors in many of the Iowa network cases. WLS does not converge on the urban network. Its reported urban predictions are the solver initialization and should not be interpreted as converged WLS estimates.

Privileged Information Required by WLS. Our WLS baseline receives simulator derived information beyond the measurements available to Mycelium:

1. **Exact network model:** per phase connectivity, impedances, transformers, regulators, shunts, and low voltage secondaries. In a 240-bus diagnostic, an erroneous regulator shunt term increased voltage error from approximately 11 to 105V.
2. **True operating topology:** the actual switch states used to construct the admittance matrix.

Table 11: state estimation errors and per-case runtimes for WLS and Mycelium across benchmark networks and sensor policies. Each result is voltage magnitude MAE in volts (per unit) / voltage angle MAE in degrees. † marks WLS divergence, and ‡ marks urban cases in which WLS did not converge and returned its initialization.

Network	Policy	WLS		Ours	
		Result	Time	Result	Time
240_bus	nm_dense	34.0 (.0091) / .63	7s	**10.3 (.0022) / .67**	.4s
	nm_medium	**11.4 (.0029) / .23**	7s	13.8 (.0026) / .81	.4s
	nm_sparse	18.8 (.0049) / **.35**	6s	**13.0 (.0028)** / .80	.4s
	ami_dense	21.6 (.0057) / **.40**	7s	**12.4 (.0024)** / .76	.4s
	ami_medium	34.0 (.0091) / .63	8s	**9.9 (.0022) / .65**	.4s
	ami_sparse	78.3 (.0149) / **.62**	5s	**10.4 (.0023)** / .66	.4s
296_bus	nm_dense	66.1 (.0183) / 1.23	10s	**11.2 (.0020) / .62**	.4s
	nm_medium	58.7 (.0164) / 1.10	9s	**10.1 (.0022) / .58**	.4s
	nm_sparse	36.1 (.157†) / 4.08†	13s	**11.0 (.0024) / .67**	.4s
	ami_dense	65.7 (.0180) / 1.23	9s	**10.5 (.0020) / .63**	.4s
	ami_medium	66.1 (.0183) / 1.23	8s	**11.0 (.0021) / .62**	.4s
	ami_sparse	68.4 (.0188) / 1.27	7s	**10.4 (.0020) / .65**	.4s
industrial	nm_dense	**19.6** (.0091) / **.81**	2.2m	20.7 (**.0056**) / 1.09	1.0s
	nm_medium	**19.8 (.0085) / .84**	1.9m	38.6 (.0082) / 1.15	1.0s
	nm_sparse	**12.0 (.0044) / .81**	9.7m	53.7 (.0129) / 1.39	1.0s
	ami_dense	**5.8 (.0024) / .18**	5.7m	41.5 (.0077) / 1.27	1.0s
	ami_medium	**7.9 (.0041) / .28**	3.8m	49.0 (.0091) / 1.28	1.0s
	ami_sparse	66.5 (.0114) / 2.77	10.1m	**57.6 (.0109) / 1.32**	1.0s
rural	nm_dense	**3.82 (.0049) / .59**	4.0h	4.01 (.0056) / 1.26	4.5s
	nm_medium	**3.72 (.0045) / .54**	2.3h	4.09 (.0060) / 1.37	4.5s
	nm_sparse	**3.12 (.0019) / .20**	1.4h	4.37 (.0072) / 1.43	4.5s
	ami_dense	3.82 (.0049) / **.59**	4.0h	**1.29 (.0044)** / 1.26	4.5s
	ami_medium	3.82 (.0049) / **.60**	2.3h	**1.43 (.0049)** / 1.36	4.5s
	ami_sparse	3.82 (.0049) / **.59**	1.3h	**1.90** (.0070) / 1.45	4.5s
urban	nm_dense‡	556.4 (.0530) / 4.67	6.2h	**471.6 (.0453) / 1.97**	18.9s
	nm_medium‡	556.1 (.0517) / 4.59	4.2h	**543.1 (.0504) / 1.45**	18.9s
	nm_sparse‡	556.4 (.0530) / 4.67	2.9h	557.6 (.0551) / **1.50**	18.9s
	ami_dense‡	556.1 (.0517) / 4.59	6.1h	**72.4 (.0115) / 4.53**	18.9s
	ami_medium‡	556.4 (.0530) / 4.67	4.0h	**68.8 (.0169)** / 4.71	18.9s
	ami_sparse‡	556.1 (.0517) / 4.59	3.1h	**451.1 (.0472) / 2.16**	18.9s

3. **True tap positions:** transformer and regulator taps at every timestep, including those used for initialization.
4. **Correct mappings and zero injections:** bus and phase assignments for measurements, plus known zero injection buses imposed as hard Hachtel constraints.
5. **True noise scales:** weights derived from the measurement noise distributions used by the simulator.
6. **Substation angle reference:** an available source angle at every timestep, including under sparse sensor policies.

These assumptions make WLS an oracle-assisted benchmark: its results reflect state estimation with exact topology, device settings, phase mapping, and noise statistics. Mycelium receives less operating information and must infer some of these quantities from sparse measurements. Performance comparisons should therefore be interpreted in light of these different inputs.

Table 12 compares Mycelium and CANOS under each sensor policy. Mycelium has lower per-unit magnitude error under five of the seven policies and lower angle error under six, while CANOS has lower magnitude error in volts under every policy. Because networks have different voltage bases,

Table 12: Per policy state estimation on the benchmark (all five networks, with fault, 24 timesteps).

	CANOS				Ours			
Policy	MAE (V)	MAE (p.u.)	MAE (°)	MAPE	MAE (V)	MAE (p.u.)	MAE (°)	MAPE
clean	**1.98**	**.0017**	**.63**	**0.35%**	18.71	.0079	1.77	5.43%
nm_dense	**6.21**	**.0100**	3.83	**4.97%**	116.36	.0145	**1.37**	5.31%
nm_medium	**10.24**	.0205	4.15	6.28%	135.07	**.0166**	**1.32**	**5.38%**
nm_sparse	**12.96**	.0276	3.66	7.31%	140.04	**.0186**	**1.36**	**5.58%**
ami_dense	**8.74**	.0113	3.53	5.73%	22.08	**.0063**	**1.99**	**5.25%**
ami_medium	**12.17**	.0230	3.50	6.85%	23.48	**.0080**	**2.09**	**5.40%**
ami_sparse	**14.21**	.0288	3.47	7.65%	113.26	**.0167**	**1.56**	**5.63%**
Aggregate	**9.50**	.0175	3.25	5.59%	81.28	**.0127**	**1.64**	**5.42%**

the per-unit and volt metrics can rank models differently. A breakdown by voltage level would be needed to identify the source of this reversal.

E.2 Phase Attribution

We also compare Mycelium with a correlation-based phase matcher. For each eligible customer, the matcher compares its 24-step meter profile with the profiles for the three phases at its serving transformer and assigns the phase with the highest correlation. We evaluate variants using transformer active power (P_{xfmr}) and active and reactive power jointly ($P{+}Q_{\mathrm{xfmr}}$). All methods are scored on the same eligible customer predictions.

Table 13 shows that the 20,000-bus Mycelium model exceeds both reported power profile variants on the test split and frozen benchmark. The 10,000-bus model closely matches the P_{xfmr} variant numerically (0.5713 versus 0.5741 on test and 0.4001 versus 0.3999 on the benchmark), but trails the $P{+}Q_{\mathrm{xfmr}}$ variant on the benchmark (0.4001 versus 0.4200). The two matching variants also reverse rank across splits, showing that their relative performance depends on the evaluation distribution and choice of signals.

Table 13: Pooled phase attribution accuracy. Populations are matched to the models': $n{=}1{,}315{,}577$ on the benchmark, $n{=}2{,}862$ on the test split.

Method	**Benchmark**	**Test**
correlation-based (P_{xfmr})	0.3999	0.5741
correlation-based ($P{+}Q_{\mathrm{xfmr}}$)	0.4200	0.3637
Mycelium 10k	0.4001	0.5713
Mycelium 20k	**0.4387**	**0.6150**

No single number characterizes the baseline. The two variants swap rank between splits – $P{+}Q$ is stronger on the benchmark, P on the test split. We report both rather than select one, since choosing a single variant would move the comparison by more than 0.2 in either direction.

E.3 Fault Localization

We compare Mycelium with STGATv2 on the frozen benchmark, reporting results by sensor policy and network. Localization is evaluated on faulted samples over each network's full candidate set of buses, lines, and transformers. Loc. denotes exact localization accuracy, Equip. denotes equipment type accuracy, and Hop@k measures localization within k hops. Class reports fault class accuracy.

Table 14 shows that Mycelium achieves higher exact localization accuracy under six sensor policies and ties STGATv2 under dense AMI sensing. It also improves equipment type accuracy and all hop-based metrics under every policy. However, performance remains sensitive to observation quality: Mycelium's exact localization accuracy falls from 0.500 under clean measurements to 0.102 under sparse, noisy, and missing measurements.

Table 14: Fault diagnosis by sensor policy on the frozen benchmark, with 108 faulted samples per policy for localization. Det. F1 is the fault versus normal detection F1 obtained by collapsing the fault class matrix. Class_5 is the 5-way type accuracy among genuinely faulted samples. Higher is better for all metrics. Best comparable values within each policy are bold, including ties.

Policy	Model	Det. F1	class_5	Loc.	Hop@1	Hop@2	Hop@3
clean	STGATv2	-	-	0.213	0.250	0.269	0.306
	Mycelium	1.000	0.704	**0.500**	**0.537**	**0.583**	**0.583**
nm_dense	STGATv2	-	-	0.130	0.213	0.222	0.306
	Mycelium	1.000	0.704	**0.185**	**0.315**	**0.370**	**0.435**
nm_medium	STGATv2	-	-	0.130	0.231	0.241	0.296
	Mycelium	1.000	0.769	**0.185**	**0.324**	**0.361**	**0.417**
nm_sparse	STGATv2	-	-	0.065	0.157	0.157	0.222
	Mycelium	1.000	0.769	**0.102**	**0.259**	**0.296**	**0.361**
ami_dense	STGATv2	-	-	**0.111**	0.194	0.194	0.259
	Mycelium	1.000	0.722	**0.111**	**0.213**	**0.259**	**0.324**
ami_medium	STGATv2	-	-	0.102	0.176	0.185	0.241
	Mycelium	1.000	0.732	**0.148**	**0.315**	**0.343**	**0.407**
ami_sparse	STGATv2	-	-	0.074	0.139	0.185	0.250
	Mycelium	1.000	0.648	**0.139**	**0.259**	**0.278**	**0.352**

Table 15: Fault diagnosis by network on the frozen benchmark, pooled across sensor policies. n counts faulted samples used for localization. Det. F1 is the fault versus normal detection F1 obtained by collapsing the fault class matrix. Class_5 is the 5-way type accuracy among genuinely faulted samples. Higher is better for all metrics. Best comparable values within each network are bold.

Network	Model	n	Det. F1	class_5	Loc.	Hop@1	Hop@2	Hop@3
240_bus	STGATv2	189	-	-	0.159	0.159	0.159	0.159
	Mycelium	189	1.000	0.667	**0.312**	**0.455**	**0.481**	**0.545**
296_bus	STGATv2	189	-	-	0.005	0.090	0.138	0.201
	Mycelium	189	1.000	0.788	**0.116**	**0.270**	**0.307**	**0.307**
industrial	STGATv2	126	-	-	0.127	0.127	0.127	0.254
	Mycelium	126	1.000	0.754	**0.214**	**0.238**	**0.310**	**0.381**
rural	STGATv2	189	-	-	0.148	**0.339**	**0.344**	0.439
	Mycelium	189	1.000	0.783	**0.185**	0.317	0.333	**0.444**
urban	STGATv2	63	-	-	**0.222**	**0.317**	**0.317**	**0.317**
	Mycelium	63	1.000	0.429	0.079	0.206	0.286	0.286

The network-level breakdown in Table 15 reveals substantial variation across networks. Mycelium improves exact localization on both Iowa networks and the industrial and rural networks. STGATv2 performs better on the urban network, with exact accuracy of 0.222 versus 0.079, and retains a small advantage in rural Hop@1 and Hop@2. Thus, the aggregate gains do not imply uniformly better localization across networks.

E.4 Per-network Benchmark Results

Table 16 reveals variation obscured by pooled metrics. The urban network has the largest per-unit and absolute voltage errors, while the industrial network has a MAPE of 60.92%, compared with less than 1.5% on every other benchmark network. This raises network macro MAPE to 12.86%, versus 5.43% pooled. Exact fault localization accuracy ranges from 0.0794 on urban to 0.3122 on 240_bus. Phase attribution is evaluated only on rural and urban, where accuracy is 0.4898 and 0.3782, respectively.

The industrial MAPE is highly sensitive to near-zero targets. Although buses without a valid voltage base are excluded, some de-energized buses retain a positive base and enter the metric with true magnitudes as low as 1.6×10^{-4} p.u. Dividing absolute errors by these targets can produce extreme percentage errors. The industrial network's per-unit MAE is 0.0097. Its median and 90th percentile absolute percentage errors are 0.39% and 0.86%, and its MAPE after excluding the worst decile is 0.56%. We therefore emphasize per-unit MAE for voltage magnitude and interpret MAPE cautiously.

Table 16: Per-network performance of Mycelium 10k on the frozen benchmark. Network macro averages weight networks equally within each metric's evaluation population. Pooled metrics aggregate predictions across networks. Sample counts n refer to evaluation windows. Phase attribution has eligible targets only on the rural and urban networks. Superscripts k indicate the number of networks included when fewer than five contribute. All results use the same checkpoint and evaluation run.

		State estimation				Phase	Switch			Fault					
Split	Configuration	p.u. ↓	v_{mag} (V) ↓	v_{ang} (deg) ↓	MAPE (%) ↓	Acc. ↑	Prec. ↑	Rec. ↑	F1 ↑	Det. F1 ↑	class_5 ↑	Loc. ↑	Hop@1 ↑	Hop@2 ↑	Hop@3 ↑
Benchmark	240_bus (n=259)	0.0039	18.26	0.870	0.63	–	–	–	–	1.000	0.6667	0.3122	0.4550	0.4815	0.5450
	296_bus (n=259)	0.0034	18.32	0.742	0.76	–	–	–	–	1.000	0.7884	0.1164	0.2698	0.3069	0.3069
	industrial (n=196)	0.0097	40.66	1.213	60.92	–	0.9974	0.5955	0.7458	1.000	0.7540	0.2143	0.2381	0.3095	0.3810
	rural (n=259)	0.0059	2.70	1.286	0.59	0.4898	0.9644	0.9564	0.9604	1.000	0.7831	0.1852	0.3175	0.3333	0.4444
	urban (n=70)	0.0342	318.20	2.971	1.40	0.3782	0.9777	0.9714	0.9746	1.000	0.4286	0.0794	0.2063	0.2857	0.2857
	Network macro	0.0114	79.63	1.416	12.86	$0.4340^{k=2}$	$0.9798^{k=3}$	$0.8411^{k=3}$	$0.8936^{k=3}$	1.000	0.6841	0.1815	0.2974	0.3434	0.3926
	Pooled (n=1,043)	0.0127	81.28	1.636	5.43	0.4001	0.9553	0.7198	0.8210	1.000	0.7209	0.1958	0.3175	0.3558	0.4114

F Training and Memory Optimization

Hardware and precision. We train on eight NVIDIA A100 GPUs, each with 80 GB of memory, using FP32 throughout, without mixed precision.

Activation checkpointing. The backbone preserves separate hidden states for all $T = 24$ timesteps across eight message passing layers, making message passing the dominant source of activation memory. We checkpoint each layer using `torch.utils.checkpoint.checkpoint` with `use_reentrant=False` and `preserve_rng_state=True`. This reduces memory usage by recomputing intermediate activations as needed during backpropagation, allowing the configured per GPU batch to fit within 80 GB.

Preserving the random number generator state reproduces dropout masks during recomputation. Checkpointing adds no parameters and does not change the training objective or state dictionary structure. It trades additional computation during backpropagation for reduced activation storage and is disabled during inference.