# PHYSICS-AWARE DISTILLATION OF A GRID FOUNDATION MODEL FOR REAL-TIME EDGE INFERENCE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Grid foundation models (GridFMs) are graph transformers pretrained on a masked power-flow objective, and once trained they evaluate power flow orders of magnitude faster than an iterative solver. Their size, however, targets data-centre inference, whereas low-inertia operation — grids with little rotating reserve, where disturbances evolve within milliseconds — increasingly asks for inference at the edge, next to the measurement, at batch size one and within a single synchrophasor (PMU) reporting frame. We distil a 20.1M-parameter, 12-layer GridFM teacher into a 0.62M-parameter, 6-layer student, a $32.5\times$ parameter reduction at half the message-passing depth, using an objective that combines the masked power-flow loss with output, physics-residual, latent-state and attention transfer. The physics-residual term, which asks the student to reproduce the teacher's active- and reactive-power residual profile rather than only to minimise its own, has not previously been applied to typed power-grid graphs. We evaluate on a renewable variant of the IEEE 118-bus benchmark grid in which 26 of the conventional (synchronous) generators are replaced by inverter-based renewables with battery flexibility and standardised reactive-power limits (IEEE 1547), giving $97\,529$ physically valid (AC-OPF-feasible) operating scenarios, and we ablate all eight combinations of the transfer terms, reporting accuracy against ground truth and fidelity to the teacher separately. Distillation lowers student error on all four predicted quantities, by up to $49\%$ on active-power dispatch, and one of the eight distilled configurations is more accurate than the teacher on reactive-power dispatch. The gain is largely carried by the output matching term. The physics-residual term, in the layer-aggregated form used here, reduces mean accuracy on all four quantities, which we attribute to the spatial averaging it applies to a bus-level residual field. We study the effect of the parameter reduction on latency on the target device to validate the effectiveness of the reduced layer count.

## 1 INTRODUCTION

Power-system operation is moving to the grid edge. Converter-interfaced renewables, batteries and flexible loads have displaced the synchronous machines that once provided inertia, and the dynamics of the resulting low-inertia systems unfold on timescales that leave little room for a round-trip to a control centre (Milano et al., 2018; Dörfler & Groß, 2023). Visions of autonomous energy grids therefore place monitoring and control decisions locally, next to the distributed resources they act on (Kroposki et al., 2020), and the measurement infrastructure already runs at that pace: synchrophasors report at 10–50 frames per second, i.e. every 20–100 ms (IEEE, 2011). Edge intelligence, running learned models on embedded hardware at the point of measurement, is the natural enabling paradigm (Zhou et al., 2019; Gooi et al., 2023), but it requires models that fit the device and answer within the frame.

Graph neural networks (GNNs) are the natural learned representation of a grid, with buses, generators and branches as typed nodes and relations (Liao et al., 2022). Recent work has scaled this idea into grid foundation models (GridFMs): graph transformers pretrained on a masked power-flow (PF) objective, regularised by a Kirchhoff physics loss, across many topologies and operating points (Hamann et al., 2024; Puech et al., 2025). Once trained, such a model acts as a PF surrogate



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.

that is three to four orders of magnitude faster than iterative numerical solution (Puech et al., 2024), which makes it attractive precisely for the high-rate, local inference that edge operation demands.

The obstacle is size. The GridFM teacher used in this work has 12 layers, 8 attention heads and 20 million parameters, and was designed for cloud or HPC inference. An edge device must instead execute it at batch size one, within a bounded latency and memory budget. For GNNs the parameter count is a poor proxy for that cost: irregular gather/scatter and the number of message-passing layers dominate the runtime (Barillaro et al., 2025), so depth, not width, is what has to shrink. The standard compression toolbox, quantization (Jacob et al., 2018; Nagel et al., 2021; Tailor et al., 2021), hardware-aware pruning (Shen et al., 2022) and once-for-all supernets (Cai et al., 2020), reduces arithmetic and storage but is agnostic to physics: nothing in these procedures guarantees that the compressed model still respects nodal power balance.

Knowledge distillation (KD) addresses the functional side of the problem by training a small student to reproduce a large teacher through its outputs, its intermediate representations or its attention maps (Bucila et al., 2006; Ba & Caruana, 2014; Hinton et al., 2015; Romero et al., 2015; Zagoruyko & Komodakis, 2017; Gou et al., 2021), and has been extended to graphs (Tian et al., 2025). For a physical system, however, a student should inherit more than the teacher's predictions: it should inherit the teacher's physical consistency, that is, how its active- and reactive-power residuals are distributed over the network. Classification-style KD has no term for that, and the physics-aware distillation that has recently appeared for PDE solvers and PDE foundation models (Bounja et al., 2025; Musekamp et al., 2026) has not been applied to typed power-grid graphs with bus-type-dependent masking. This paper closes that gap by distilling a pretrained GridFM into a $32.5\times$ smaller, half-depth student with a physics-aware objective, and by measuring the result on edge hardware. Our contributions are:

- **A physics-aware distillation objective** for grid foundation models that combines output, physics-residual, latent and attention terms with a frozen GridFM teacher (Section 3, Eq. 1). The physics-residual term, which matches the teacher's active/reactive residual profile rather than only penalising the student's own residual, is new.
- **A renewable-grid benchmark** derived from IEEE 118, in which 26 synchronous units are replaced by converter-interfaced generators with battery flexibility and IEEE 1547 reactive bands, and $100\,000$ AC-OPF-feasible scenarios are generated (Section 3.2).
- **A systematic ablation** over all combinations of the four losses (Tables 2 and 3) that reports accuracy against ground truth and fidelity to the teacher separately, since the two are known to diverge (Stanton et al., 2021).
- **Edge deployment** of the $0.6$M-parameter student on an NVIDIA Jetson Orin, with measured latency and memory (Section 4.2).

## 2 State of the Art

### 2.1 Related Work

**Learning-based power flow and grid foundation models.** The first GNN solver for AC power flow was trained by minimising the Kirchhoff violation directly, without labelled solutions (Donon et al., 2020); physics-guided and physics-informed variants followed for power flow (Hu et al., 2021), state estimation (Ngo et al., 2024; Cao et al., 2024) and optimal power flow (Nellikkath & Chatzivasileiadis, 2022; Varbella et al., 2024). Message-passing surrogates such as PowerFlowNet report 4–48$\times$ speed-ups over Newton–Raphson (Lin et al., 2024), iterative self-ensembling schemes improve physical consistency (Zhao et al., 2025), and large GNN OPF solvers scale to thousands of buses and $N-1$ contingencies (Piloto et al., 2024), though generalisation across topologies remains the open problem (Arowolo & Cremer, 2025). Foundation-model pretraining is the most recent answer to it (Hamann et al., 2024; Puech et al., 2025); the closest competitor to our student is MxGPS, a 1.6M-parameter multiplex graph transformer that is small *by design* and trained jointly on state estimation and PF (Papaioannou et al., 2026). We take the complementary route: we start from an existing pretrained GridFM, shrink it by distillation, and measure the result on hardware.

**Knowledge distillation for graphs, transformers and physics.** Distillation descends from model compression (Bucila et al., 2006) and the observation that shallow networks can mimic deep ones (Ba & Caruana, 2014), and was popularised through soft targets (Hinton et al., 2015), intermediate hints (Romero et al., 2015) and attention transfer (Zagoruyko & Komodakis, 2017); Gou et al. (2021) survey the field. On graphs, KD has been used to transfer local structure between GNNs (Yang et al., 2020), to train shallow students from deep teachers (Yan et al., 2020), to compare representation-level objectives (Joshi et al., 2024) and, at the extreme, to replace the GNN by an MLP for inference speed (Zhang et al., 2022; Tian et al., 2025). For transformers, TinyBERT established the multi-loss template of matching embeddings, hidden states, attention matrices and predictions (Jiao et al., 2020), MiniLM showed that distilling only the last-layer self-attention avoids layer-mapping heuristics (Wang et al., 2020), and attention distillation has been adapted to graph transformers (Huang et al., 2024). Closest to our physics term, KD-PINN distils a large physics-informed solver into a millisecond-latency student using output and PDE-residual losses (Bounja et al., 2025), and PDE foundation models have been distilled into surrogates orders of magnitude smaller (Musekamp et al., 2026). None of these targets power-grid models, and none matches the *distribution* of physical residuals between teacher and student.

**Efficient inference at the grid edge.** Edge intelligence for power systems is an active field (Gooi et al., 2023; Zhou et al., 2019), and the general compression toolbox is mature: integer quantization (Jacob et al., 2018; Nagel et al., 2021), its GNN-specific form that controls aggregation variance (Tailor et al., 2021), latency-budgeted structural pruning (Shen et al., 2022), once-for-all supernets (Cai et al., 2020) and node-adaptive propagation for GNN inference (Gao et al., 2022). These techniques are orthogonal to distillation and can be stacked on top of it. Measured GNN inference on Jetson-class devices has so far been reported only for generic benchmark graphs (Barillaro et al., 2025), not for grid models, and without physics metrics. Our deployment uses the standard PyTorch Geometric stack (Fey & Lenssen, 2019).

### 2.2 Research Gap

Three ingredients exist separately but have not been combined. Grid foundation models exist, but their reported speed-ups are obtained on data-centre GPUs and no edge deployment has been shown. Distillation for GNNs and transformers is mature, but it ignores physical consistency, and the physics-aware distillation that has appeared targets PDE solvers rather than typed grid graphs with bus-type-dependent masking. Edge benchmarks of GNNs exist, but not for grid models. This paper sits at the intersection: it distils a pretrained GridFM into a half-depth student with an objective that transfers outputs, latent states, attention and the teacher's physics-residual profile, ablates every combination of these terms on a renewable version of IEEE 118, and reports latency and memory on an NVIDIA Jetson Orin.

## 3 Methodology

**Setup.** Let $f_T$ denote a frozen, pretrained GridFM teacher and $f_S$ a reduced-parameter student with the same input interface. Both receive a typed grid graph in which the unknown quantities at each bus are masked according to bus type (Section 3.2), and both are trained on the same masked power-flow objective $\mathcal{L}_{\mathrm{GraphKit}}$. The student is additionally trained to reproduce the teacher's outputs, latent states, attention maps and physics-residual profile through the objective in Eq. 1, and is evaluated on accuracy against ground truth, fidelity to the teacher, physics residuals, and measured latency and memory on the target edge device. Notation is summarised in Appendix A.

### 3.1 Distillation

A reduced-parameter Graph Network-based Simulator (GNS) model is created by reducing the number of GNN layers, attention heads and hidden size, with exact hyperparameters reported in Table 1.

While reducing the number of parameters generally reduces the computational power required to run a model, for a GNS a reduction in the number of GNN layers has the highest impact on reducing the latency from data ingestion to finished inference. Thus, a student is trained using the same stochastic masked-learning paradigm as the teacher GridFM model Hamann et al. (2024). By using

| | Teacher | Student |
|---|---|---|
| Hidden | 48 | 24 |
| Heads | 8 | 4 |
| Layers | 12 | 6 |
| Params | 20,069,187 | 616,707 |
| Reduction | 1.0x | 32.5x |

Table 1: Comparison between full model and reduced parameter student.

distillation, we are able to leverage the learned internal representations of the teacher model to improve the quality of the student model Hinton et al. (2015); Gou et al. (2021).

To train the reduced-parameter model, we employ a distillation-based learning framework that incorporates multiple complementary distillation objectives. Specifically, we introduce: (i) an output distillation loss that enforces agreement between the student model's predictions and those of the teacher model Hinton et al. (2015); (ii) a physics-informed distillation loss that aligns the student's residuals with those of the teacher, distinct from the Kirchhoff's Current Law physics-informed neural network loss already included in the GraphKit objective Hamann et al. (2024); Ngo et al. (2024); (iii) a latent distillation loss that constrains sparse intermediate representations of the student to match corresponding internal states of the teacher Romero et al. (2015); and (iv) an attention distillation loss that transfers information encoded in the teacher's transformer and graph attention maps to the student network Zagoruyko & Komodakis (2017); Vaswani et al. (2017); Velickovic et al. (2018). It should be noted that the teacher model's weights are frozen during training, such that backpropagation only occurs within the student model.

The training loss used during distillation is composed of the GraphKit objective, including supervised training loss and the Kirchhoff's Current Law physics-informed neural network term, together with the weighted distillation terms described above. This total objective is given in Equation 1.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{GraphKit}} + \lambda_{\text{out}}\mathcal{L}_{\text{out}} + \lambda_{\text{phys}}\mathcal{L}_{\text{physKD}} + \lambda_{\text{lat}}\mathcal{L}_{\text{lat}} + \lambda_{\text{att}}\mathcal{L}_{\text{att}}. \tag{1}$$

The output term in Equation 2 resembles the most basic form of distillation Hinton et al. (2015). Specifically, by training on the outputs of the teacher rather than merely on the objective, a student model integrates information about the accuracy and structure of teacher predictions without overconstraining the training target.

$$\mathcal{L}_{\text{out}} = \frac{1}{|\mathcal{V}_b|}\sum_{i\in\mathcal{V}_b}\left\|\hat{\mathbf{y}}^{\,b}_{S,i} - \hat{\mathbf{y}}^{\,b}_{T,i}\right\|_2^2 + \frac{1}{|\mathcal{V}_g|}\sum_{j\in\mathcal{V}_g}\left\|\hat{\mathbf{y}}^{\,g}_{S,j} - \hat{\mathbf{y}}^{\,g}_{T,j}\right\|_2^2. \tag{2}$$

Similarly, physics residual distillation allows the student model to better integrate information about which physical residuals can be cleanly reduced by imitating the teacher's physical residual profile. In this sense, the teacher provides a smoother target for the reduced-capacity student than the physics residual alone. A direct physics residual penalty may force the student to spend excessive capacity on poorly constrained values, such as end-node voltages with limited information, while residual distillation can instead guide the student toward the residual patterns that remain meaningful under the teacher's learned physical representation. The resulting loss is shown in Equation 3.

$$\mathcal{L}_{\text{physKD}} = \frac{1}{|\mathcal{M}|}\sum_{(\ell_S,\ell_T)\in\mathcal{M}}\left\|\rho_S^{(\ell_S)} - \rho_T^{(\ell_T)}\right\|_2^2. \tag{3}$$

Physics residuals in GridFM are calculated for active and reactive power. In Equation 4, these residuals are summarized as a mean bus-level residual norm for each matched model layer.

$$\rho_M^{(\ell)} = \frac{1}{|\mathcal{V}_b|}\sum_{i\in\mathcal{V}_b}\left\|\begin{bmatrix} r_{M,i}^{P,(\ell)} \\ r_{M,i}^{Q,(\ell)} \end{bmatrix}\right\|_2, \qquad M \in \{S,T\}. \tag{4}$$

Latent distillation loss is introduced to allow the student to imitate the intermediate representations of the teacher model, following the general idea of using intermediate hints to train thinner networks Romero et al. (2015). Since the student model has fewer layers, an experiment-specific assignment of student layers to teacher layers must be defined. The latent-state matching objective is given in Equation 5.

$$\mathcal{L}_{\text{lat}} = \frac{1}{|\mathcal{M}|\,|\mathcal{T}|} \sum_{(\ell_S,\ell_T)\in\mathcal{M}} \sum_{\tau\in\mathcal{T}} \frac{1}{|\mathcal{V}_\tau|} \sum_{i\in\mathcal{V}_\tau} \left\| P_\tau^{(\ell_S,\ell_T)} \mathbf{h}_{S,i}^{\tau,(\ell_S)} - \mathbf{h}_{T,i}^{\tau,(\ell_T)} \right\|_2^2 . \tag{5}$$

The final type of distillation loss implemented, shown in Equation 6, is the attention loss, following the general principle of attention transfer Zagoruyko & Komodakis (2017). This term encourages the attention maps of the student model to match the attention maps of the teacher model. In this way, the student learns from the teacher what data it must attend to. Since the student model typically has fewer heads, only averaged attention maps are compared, such that attention may be redistributed among heads by the student.

$$\mathcal{L}_{\text{att}} = \frac{1}{|\mathcal{M}|\,|\mathcal{R}|} \sum_{(\ell_S,\ell_T)\in\mathcal{M}} \sum_{r\in\mathcal{R}} \frac{1}{|\mathcal{E}_r|} \sum_{e\in\mathcal{E}_r} \left\| \bar{\alpha}_{S,e}^{r,(\ell_S)} - \bar{\alpha}_{T,e}^{r,(\ell_T)} \right\|_2^2 . \tag{6}$$

### 3.2 DOMAIN

Distillation is applied here to a power system domain foundation model, as described in Section 3. The monitoring task it performs is load flow, or power flow (PF), the most basic study of a power grid network: given how generation and load are balanced across the network, PF determines the voltage magnitude and the phase angle at every node, and the flow on a branch follows from those two quantities at its two end nodes together with the branch admittance. Each node carries four quantities, the active power $P_g$ and the reactive power $Q_g$ injected by generation there, and the voltage magnitude $V_m$ and phase angle $V_a$, with the demand $(P_d, Q_d)$ specified as an input. What is unknown at a node, and therefore what has to be predicted, depends on its bus type, and this classical convention is exactly what makes PF a masked prediction problem: PQ buses predict the voltage $(V_m, V_a)$, PV buses predict $(V_a, Q_g)$, and the slack bus predicts $(P_g, Q_g, V_m)$.

Power flow is traditionally solved by numerical methods, most commonly Newton-Raphson (NR), and we use an open-source Julia tool to generate the data and labels for both the foundation model and the distilled model. The experiments run on the IEEE 118-node benchmark grid as provided in the PGLib library IEEE PES Task Force on Benchmarks for Validation of Emerging Power System Algorithms (2017), whose nominal load is 4242 MW / 1438 Mvar. That benchmark is a conventional synchronous system, where synchronous generators manage active and reactive power inherently and, as long as the load stays within the total generation capacity, balance it with few violations. It also leaves the balancing to a single reference node, the slack bus, which takes up whatever the fixed generation fails to cover, and that is where the resulting stress becomes visible.

Grid operation is bounded by the standard steady-state limits below, and one breach anywhere is enough for a scenario to count as stressed:

$$0.94 \text{ p.u.} \;\le V_{m,i} \;\le\; 1.06 \text{ p.u.}, \qquad i \in \mathcal{B}, \tag{7}$$

$$|S_\ell| \;\le\; S_\ell^{\max}, \qquad \ell \in \mathcal{L}, \tag{8}$$

where $V_{m,i}$ is the voltage magnitude at bus $i$, given per unit so that 1.0 p.u. is the nominal value, $|S_\ell|$ is the apparent-power flow on branch $\ell$, and $S_\ell^{\max}$ is that branch's MVA (thermal) rating, ranging over $[72,\ 7218]$ MVA across the 186 branches.

This case study moves the benchmark towards a renewable grid: the flexibility to meet the additional load demand comes from distributed generation rather than from the slack bus, and with it the active and reactive power and hence the voltage are regularised on the renewable side. The slack bus is retained as the angle reference, but carries only about 7% of the scenario-to-scenario variation in active dispatch against roughly three quarters at the flexible units. Each renewable generator carries a boolean flag, set to 1 when it participates in the flexible dispatch, and that flag releases two resources: a battery for active power, and converter control for reactive power. The flag is

redrawn independently for every scenario: a count $k \sim \mathcal{U}\{5, \ldots, 15\}$ is sampled and a random $k$-subset of the 26 renewable buses is activated, so that both how much flexibility is available and where it sits vary across scenarios, while the remaining buses are deactivated ($P_g^{\max} = Q_g^{\max} = Q_g^{\min} = 0$) and reverted to PQ so their solved voltage cannot enter the model as a known input. The battery enters the merit order, the order in which resources are dispatched cheapest first, at roughly 18 EUR/MWh. The PGLib coefficients put the cheapest conventional unit at 13 EUR/MWh and the next at 22 EUR/MWh, with the rest of the fleet spread over 24–35 EUR/MWh and the reference unit at 26 EUR/MWh, so the battery is dispatched second, after the cheapest unit and ahead of the whole synchronous fleet. Each of the 26 converted renewable buses is now a dispatchable generator, sized from the synchronous unit that once stood there: its active capacity matches that unit's rating, and its reactive band is the wider of the unit's native range and $\pm 0.44\, P_g^{\max}$, the inverter reactive requirement of IEEE Std 1547-2018 IEEE (2018).

Each load scenario is solved as a single-period AC OPF with PowerModels.jl and Ipopt, minimising generation cost subject to the power-flow equations, the voltage band equation 7, the branch thermal limits equation 8 and every unit's active and reactive bounds. The dispatch is therefore re-optimised for every scenario rather than frozen at a base case, so any converged solution satisfies the constraints to solver tolerance and is a feasible operating point; non-converged scenarios are discarded. The learning target is thus a mapping from load to a feasible state.

**Dataset Generation** Each of the 100 000 load scenarios pairs one global scaling factor with localised per-bus noise Puech et al. (2025):

$$X_{d,k}^{(i)} = s^{(i)}\, \varepsilon_{X,k}^{(i)}\, X_{d,k}^{\text{base}}, \qquad \varepsilon_{X,k}^{(i)} \sim \mathcal{U}(1-\sigma,\, 1+\sigma), \qquad X \in \{P, Q\}, \tag{9}$$

over the load buses $k$. The scalar $s^{(i)}$ comes from a normalised aggregate load curve and is shared by every bus, preserving the system-wide co-movement of demand, while $\varepsilon_{X,k}^{(i)}$ is drawn independently across buses and separately for $P$ and $Q$ ($\sigma = 0.18$, seed 42); total demand spans 1855–4051 MW (mean 2605 MW). Of the 100 000 attempts, 97 529 converge; feasibility falls with $k$, from 99.7% at $k = 15$ to 93.2% at $k = 5$, so the retained set is mildly skewed towards higher flexibility. The scenarios are split $80/10/10$ by load scenario, giving $78\,025/9\,752/9\,752$ and preventing a load profile from appearing in more than one set.

## 4 Experiments

Every model in this section shares the student architecture of Table 1 and the dataset of Section 3.2, and differs only in which distillation terms are active. Two reference points bound the comparison: the frozen teacher, and a *baseline* student trained on $\mathcal{L}_{\text{GraphKit}}$ alone, that is, the same reduced architecture with no transfer from the teacher. A prior hyperparameter search fixed the weights at $\lambda_{\text{out}} = 0.75$, $\lambda_{\text{physKD}} = 0.05$, $\lambda_{\text{lat}} = 0.005$ and $\lambda_{\text{att}} = 0.005$; these values are held constant throughout, so that the ablation varies the presence of each term and not its weight. Because output distillation anchors the other three terms to a common reference, all eight combinations that include $\mathcal{L}_{\text{out}}$ were trained, and the results are reported in Table 2. Accuracy is reported as RMSE against the AC-OPF ground truth, and fidelity as RMSE against the teacher's own prediction, because the two are known to diverge (Stanton et al., 2021). All values are scaled by $10^3$. Reported means and sample standard deviations are computed across three seeds.

**Accuracy against ground truth.** Every distilled configuration improves on the baseline student for every predicted quantity, so the transfer is never harmful in aggregate. The size of the improvement, however, differs sharply by quantity. Taking the best configuration for each, distillation removes 48.7% of the baseline error on active-power dispatch $P_g$, 29.1% on the voltage angle $V_a$, 34.4% on the voltage magnitude $V_m$ and 39.5% on reactive-power dispatch $Q_g$. A more informative measure is how much of the gap between the baseline student and the teacher is recovered: 65.7% for $V_m$, 57.4% for $P_g$ and 35.6% for $V_a$. In absolute terms the best student remains 1.38× the teacher error on $V_m$ but 3.88× on $V_a$ and 3.38× $P_g$, so halving the message-passing depth costs most in the angle and dispatch channels, which are the ones that depend on propagating information across the network rather than on local nodal quantities.

| Ablations | | | | Quantities ($\times 10^3$) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | RMSE vs. ground truth | | | | RMSE vs. teacher | | |
| $\mathcal{L}_{\text{out}}$ | $\mathcal{L}_{\text{physKD}}$ | $\mathcal{L}_{\text{lat}}$ | $\mathcal{L}_{\text{att}}$ | $V_m$ | $V_a$ | $P_g$ | $Q_g$ | $V_m$ | $P_g$ | $Q_g$ |
| **Teacher** | | | | **0.770** ±**0.175** | **5.581** ±**1.481** | **6.705** ±**1.615** | 28.301 ±6.598 | – | – | – |
| **Baseline** | | | | 1.618 ±0.086 | 30.581 ±6.808 | 44.102 ±5.120 | 46.269 ±4.570 | 1.547 ±0.127 | 40.876 ±6.109 | 44.850 ±6.302 |
| • | | | | 1.197 ±0.151 | **21.675** ±**3.644** | 23.055 ±2.812 | 31.142 ±3.511 | 0.862 ±0.264 | 19.053 ±4.189 | 13.437 ±5.196 |
| • | • | | | 1.092 ±0.122 | 21.956 ±4.174 | 22.868 ±2.394 | 28.342 ±6.042 | 0.781 ±0.047 | 18.769 ±3.055 | 10.694 ±1.372 |
| • | | • | | 1.170 ±0.112 | 22.596 ±4.294 | 23.008 ±2.013 | 29.485 ±3.874 | 0.848 ±0.230 | 18.987 ±3.070 | 12.084 ±2.298 |
| • | | | • | **1.061** ±**0.083** | 21.971 ±5.365 | **22.642** ±**3.217** | 28.314 ±5.663 | **0.726** ±**0.119** | **18.738** ±**4.034** | **9.743** ±**2.082** |
| • | • | • | | 1.211 ±0.118 | 22.466 ±2.616 | 24.210 ±0.647 | 29.699 ±4.312 | 0.915 ±0.143 | 20.415 ±1.305 | 11.673 ±2.964 |
| • | • | | • | 1.164 ±0.060 | 23.928 ±4.992 | 24.596 ±3.258 | 30.111 ±4.451 | 0.833 ±0.152 | 20.529 ±4.676 | 11.675 ±3.328 |
| • | | • | • | 1.103 ±0.085 | 23.012 ±6.516 | 24.070 ±3.604 | **27.980** ±**5.243** | 0.818 ±0.163 | 20.085 ±4.438 | 12.597 ±2.824 |
| • | • | • | • | 1.192 ±0.109 | 26.360 ±4.727 | 26.294 ±2.457 | 28.896 ±5.078 | 0.924 ±0.193 | 22.211 ±3.587 | 13.606 ±4.058 |

Table 2: Distillation ablation. Columns left of the rule mark active terms; $\mathcal{L}_{\text{GraphKit}}$ is present in every row. **Teacher** is the frozen 20.1M-parameter GridFM and **Baseline** the 0.62M-parameter student without transfer. Values are mean RMSE $\pm$ sample standard deviation across three seeds, scaled by $10^3$. Lower is better, bold is the best mean.

Output distillation combined with attention distillation is the best configuration for $V_m$ and $P_g$, while output distillation alone is best for $V_a$. For $Q_g$, the best configuration additionally includes $\mathcal{L}_{\text{lat}}$ and $\mathcal{L}_{\text{att}}$. The $Q_g$ column spans 27.980 to 31.142 across the eight distilled configurations; with three seeds, the ordering within this relatively narrow range should still be interpreted cautiously.

**Reactive power can fall below the teacher.** One of the eight distilled configurations achieves a lower mean $Q_g$ error than the teacher (28.301), while all eight improve strongly over the undistilled baseline (46.269); the effect is therefore attributable to distillation rather than to the reduced architecture. A student surpassing its teacher on a subset of outputs is a documented outcome of distillation (Stanton et al., 2021; Beyer et al., 2022): the teacher's predictions are a smoother regression target than the AC-OPF labels, and a reduced-capacity model fitted to that target can generalise better on the noisiest channel. $Q_g$ is the plausible candidate, being the least constrained quantity in the case study, set by converter control within the wide IEEE 1547 band rather than pinned by the power-flow equations. Isolating the mechanism would require varying student capacity, which we have not done.

**Fidelity to the teacher selects different objectives.** The configuration that best matches ground truth is not always the one that best imitates the teacher. Output plus attention distillation minimises teacher-referenced error for all three reported quantities and also minimises ground-truth error for $V_m$ and $P_g$; $V_a$ is minimised by output distillation alone, and for $Q_g$ the most accurate configuration additionally includes $\mathcal{L}_{\text{lat}}$. Relative to the baseline, the best configurations reduce teacher-referenced error by $53.0\%$ on $V_m$, $54.2\%$ on $P_g$ and $78.3\%$ on $Q_g$, all larger than the corresponding gains against ground truth. The added terms therefore buy fidelity more readily than accuracy, and the two should be reported separately, as Stanton et al. (2021) argue for classification.

**Marginal effect of each term.** Table 3 isolates each term by averaging the change in error over every pair of configurations that differ only in that term. Output distillation dominates: it accounts for $+21.05 \pm 6.76$ on $P_g$ and $+15.13 \pm 4.19$ on $Q_g$. Its entry is estimated from the baseline-to-

$\mathcal{L}_{\text{out}}$ contrast within each run, since no configuration without output distillation was trained; the remaining three terms each average four contrasts within each run.

Against that, the three structural terms are second-order. Physics-residual distillation has a negative mean effect on all four quantities across the three seeds; its $V_a$ and $P_g$ effects are negative in all three seeds, while the $V_m$ and $Q_g$ effects change sign across seeds. Latent distillation improves $Q_g$ ($+0.46 \pm 0.17$) in all three seeds and is negative on the other three. Attention distillation has positive mean effects on $V_m$ and $Q_g$, and negative mean effects on $V_a$ and $P_g$, with the same sign in all three seeds. On this evidence, latent distillation has a small reactive-power benefit, attention distillation has mixed, comparatively small effects, and the physics-residual term is harmful on average.

| **Loss term** | $\Delta V_m$ | $\Delta V_a$ | $\Delta P_g$ | $\Delta Q_g$ |
|---|---|---|---|---|
| Output KD | $\mathbf{+0.4210}$ $\mathbf{\pm 0.0661}$ | $\mathbf{+8.9056}$ $\mathbf{\pm 8.1190}$ | $\mathbf{+21.0467}$ $\mathbf{\pm 6.7596}$ | $\mathbf{+15.1275}$ $\mathbf{\pm 4.1936}$ |
| Physics KD | $-0.0321$ $\pm 0.0689$ | $-1.3640$ $\pm 0.7896$ | $-1.2982$ $\pm 0.7447$ | $-0.0320$ $\pm 0.4751$ |
| Latent | $-0.0405$ $\pm 0.0314$ | $-1.2256$ $\pm 0.6767$ | $-1.1049$ $\pm 0.8391$ | $+0.4621$ $\pm 0.1722$ |
| Attention | $+0.0375$ $\pm 0.0370$ | $-1.6443$ $\pm 1.7849$ | $-1.1152$ $\pm 1.2020$ | $+0.8418$ $\pm 0.9918$ |

Table 3: Marginal impact of each distillation loss on test RMSE. Values are mean $\pm$ sample standard deviation across three seeds, scaled by $10^3$; positive is better. Physics KD, Latent and Attention average all paired configuration contrasts within each seed; Output KD uses the Baseline-to-$\mathcal{L}_{\text{out}}$ contrast.

### 4.1 Discussion

**Output matching carries the transfer.** Matching the teacher's outputs recovers most of what is recoverable, and the three terms that reach into its internal computation add little. This is consistent with systematic comparisons of representation-level objectives on graphs (Joshi et al., 2024): a PF teacher's prediction is already a dense, per-node, physically typed signal, unlike the single soft label of a classification task, so the marginal information in its hidden states is smaller. For this class of surrogate the simplest objective is therefore most of the method, and the cheapest, needing no layer mapping, no projection matrices and no access to teacher internals.

**The physics-residual term is harmful in its aggregated form.** The term we introduced as the novel component of the objective is the one that degrades mean accuracy on all four quantities, and the formulation in Eq. 4 is, we believe, the reason. It reduces the entire bus-level residual field to one scalar per layer, the mean norm over buses, before matching. Two residual patterns with the same mean but different spatial distributions are indistinguishable under this target, so the term carries almost no information about *where* the teacher's physical error sits, which was the property it was meant to transfer. What it does carry is a magnitude, and matching a magnitude is wrong in a specific way: wherever the teacher's residual exceeds the student's, the term pushes the student to become less physically consistent in order to agree, working against the absolute Kirchhoff penalty already inside $\mathcal{L}_{\text{GraphKit}}$. The layer mapping compounds this, since residuals at an intermediate layer of a 12-layer teacher and at the matched layer of a 6-layer student are at different stages of refinement.

The term therefore seems worth reformulating rather than discarding. Matching the residual *vector* at each bus, tail-weighted so that a few weak or heavily loaded buses cannot be averaged away, would preserve the spatial information the present form discards; making the penalty one-sided, active only where the student's residual exceeds the teacher's, would remove the incentive to inherit teacher error. We state these as the natural next experiment rather than as a result.

**What these experiments do not establish.** All configurations are evaluated across three seeds. The resulting standard deviations quantify seed-to-seed variation, but three seeds remain limited for characterising the smaller effects, particularly attention and the ordering within the $Q_g$ column. More importantly, we report no physics-residual metric, so physics-residual distillation may

improve nodal power balance while costing RMSE; mean, p95 and maximum nodal residuals per configuration would resolve this trade-off.

### 4.2 On Edge Device

The $32.5\times$ parameter reduction in Table 1 reduces FP32 weight storage from $76.6$ MiB to $2.35$ MiB, but weight count predicts neither latency nor peak memory. At batch size one, GNN inference is dominated by irregular gather/scatter and message-passing depth rather than weight arithmetic (Barillaro et al., 2025), so the effect of halving depth must be measured on device. Table 4 therefore reports batch-one measurements with the CUDA graph and topology baked, the model hotloaded, fixed power mode, synchronized timing, and warm-up discarded.

| Model | Params | p50 (ms) | p95 (ms) | p99 (ms) | Peak mem. (MiB) |
|---|---|---|---|---|---|
| Teacher | 20.07M | 24.85 | 35.22 | 41.19 | 179.0 |
| Student | 0.62M | 6.08 | 10.36 | 13.13 | 23.7 |
| Speed-up | $32.5\times$ | $4.08\times$ | $3.40\times$ | $3.14\times$ | $7.6\times$ |

Table 4: Inference latency and peak memory on an NVIDIA Jetson Orin Nano 8 GB, 15 W mode, JetPack 5.1.3 (L4T R35.5.0), PyTorch 2.1.0 and CUDA 11.4, at batch size one. Percentiles use 500 synchronized post-warm-up inferences.

The student meets the 20–100 ms synchrophasor frame of Section 1 at every percentile: p99 is 13.13 ms and all 500 inferences complete within 20 ms, whereas no teacher inference does (p50 24.85 ms). The $4.08\times$ p50 and $3.14\times$ p99 speed-ups are about twice the $2\times$ depth reduction, yet far below the $32.5\times$ parameter reduction. Transfer and normalization account for under $1\%$ of teacher and $3\%$ of student time; message passing accounts for $75.7\%$ and $54.5\%$, followed by the per-layer physics layer at $15.3\%$ and $30.4\%$, respectively.

## 5 Conclusions

We asked whether a grid foundation model can be reduced to a size appropriate for edge inference without losing the behaviour that makes it useful, and which parts of the teacher are worth transferring. Distilling a 12-layer, 20.1M-parameter GridFM into a 6-layer, 0.62M-parameter student on a renewable variant of IEEE 118 recovers much of the capacity lost to the reduction: against the same architecture trained without a teacher, error falls by up to $49\%$, and the best configuration closes about two thirds of the student-teacher gap on voltage magnitude.

The decomposition of the objective is the more useful result, and it is largely negative. Output matching accounts for almost all of the transfer; latent-state matching adds a small reactive-power improvement; attention matching has mixed, comparatively small effects; and physics-residual distillation, the term we proposed as the contribution of the objective, degrades mean accuracy on all four quantities. We attribute this to the aggregated form in which we posed it, which collapses a bus-level residual field to one scalar per layer and so transfers a magnitude rather than a spatial pattern, while giving the student an incentive to reproduce the teacher's physical error rather than reduce it. A bus-level, tail-weighted and one-sided reformulation would test the formulation rather than the underlying idea, and is the obvious next step. That accuracy and fidelity can select different configurations is a further reason to report the two separately here rather than treat teacher agreement as a proxy for quality.

A limit of these conclusions is the absence of a physics-residual metric, without which we cannot say whether the physics term trades accuracy for physical consistency, which for grid operation would be a different verdict on it. The $32.5\times$ parameter reduction and the halved depth are properties of the architecture, and we have kept them separate from any latency claim, which for a GNN at batch size one is governed by memory access rather than weight count and must be established on the device (Section 4.2). Natural continuations are the reformulation above, quantization and hardware-aware pruning (Tailor et al., 2021; Shen et al., 2022), adaptive inference depth (Rahmath P et al., 2024), and moving from inference to control, which would require a safety mechanism and closed-loop evaluation.

### AI USE STATEMENT

In this work, we used generative AI tools (Claude, ChatGPT, Codex) to assist with the literature search, with drafting and editing the text of the Abstract, Introduction, State of the Art, the discussion of results and the Conclusions, for formatting tables, for arithmetic checks on values computed by the authors from experimental logs, to ensure consistency of claims with data and for proofreading. We have not used generative AI tools to generate research ideas, data, experiments or results. All AI-suggested references were verified by the authors against the original publications, and all AI-assisted text was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

Olayiwola Arowolo and Jochen L. Cremer. Towards generalization of graph neural networks for AC optimal power flow. *arXiv preprint arXiv:2510.06860*, 2025. URL https://arxiv.org/abs/2510.06860.

Lei Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, volume 27, 2014. URL https://arxiv.org/abs/1312.6184.

Luca Barillaro, Chiara Zucco, Marianna Milano, Giuseppe Agapito, and Mario Cannataro. Evaluating GNN inference on edge computers. In *Companion Proceedings of the 16th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB Companion)*, 2025. doi: 10.1145/3768322.3769021.

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10925–10934, 2022.

Karim Bounja, Lahcen Laayouni, and Abdeljalil Sakat. KD-PINN: Knowledge-distilled PINNs for ultra-low-latency real-time neural PDE solvers. *arXiv preprint arXiv:2512.13336*, 2025. URL https://arxiv.org/abs/2512.13336.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 535–541, 2006. doi: 10.1145/1150402.1150464.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1908.09791.

Di Cao, Junbo Zhao, Weihao Hu, Nanpeng Yu, Jiaxiang Hu, and Zhe Chen. Physics-informed graphical learning and Bayesian averaging for robust distribution state estimation. *IEEE Transactions on Power Systems*, 39(2):2879–2892, 2024. doi: 10.1109/TPWRS.2023.3282413.

Balthazar Donon, Rémy Clément, Benjamin Donnot, Antoine Marot, Isabelle Guyon, and Marc Schoenauer. Neural networks for power flow: Graph neural solver. *Electric Power Systems Research*, 189:106547, 2020. doi: 10.1016/j.epsr.2020.106547.

Florian Dörfler and Dominic Groß. Control of low-inertia power systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 6:415–445, 2023. doi: 10.1146/annurev-control-052622-032657.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. URL https://arxiv.org/abs/1903.02428.

Xinyi Gao, Wentao Zhang, Yingxia Shao, Quoc Viet Hung Nguyen, Bin Cui, and Hongzhi Yin. Efficient graph neural network inference at large scale. *arXiv preprint arXiv:2211.00495*, 2022. URL https://arxiv.org/abs/2211.00495.

Hoay Beng Gooi, Tianyi Wang, and Yi Tang. Edge intelligence for smart grid: A survey on application potentials. *CSEE Journal of Power and Energy Systems*, 9(5):1623–1640, 2023. doi: 10.17775/CSEEJPES.2022.02210.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021. doi: 10.1007/s11263-021-01453-z.

Hendrik F. Hamann, Thomas Brunschwiler, Blazhe Gjorgiev, Leonardo S. A. Martins, Alban Puech, Anna Varbella, Jonas Weiss, Juan Bernabe-Moreno, Alexandre Blondin Massé, Seong Lok Choi, Ian Foster, Bri-Mathias Hodge, Rishabh Jain, Kibaek Kim, Vincent Mai, François Mirallès, Martin De Montigny, Octavio Ramos-Leaños, Hussein Suprême, Le Xie, El-Nasser S. Youssef, Arnaud Zinflou, Alexander J. Belyi, Ricardo J. Bessa, Bishnu Prasad Bhattarai, Johannes Schmude, and Stanislav Sobolevsky. Foundation models for the electric power grid. *Joule*, 8(12):3245–3258, 2024. doi: 10.1016/j.joule.2024.11.002.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Xinyue Hu, Haoji Hu, Saurabh Verma, and Zhi-Li Zhang. Physics-guided deep neural networks for power flow analysis. *IEEE Transactions on Power Systems*, 36(3):2082–2092, 2021. doi: 10.1109/TPWRS.2020.3029557.

Yisong Huang, Jin Li, Xinlong Chen, and Yang-Geng Fu. Training graph transformers via curriculum-enhanced attention distillation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=j4VMrwgn1M.

IEEE. IEEE standard for synchrophasor measurements for power systems. Technical Report IEEE Std C37.118.1-2011, IEEE Standards Association, 2011.

IEEE. IEEE standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces. Technical Report IEEE Std 1547-2018, IEEE Standards Association, April 2018. Revision of IEEE Std 1547-2003.

IEEE PES Task Force on Benchmarks for Validation of Emerging Power System Algorithms. PGLib-OPF: Benchmarks for the optimal power flow problem. https://github.com/power-grid-lib/pglib-opf, 2017. Accessed: July 12, 2026.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2704–2713, 2018. doi: 10.1109/CVPR.2018.00286.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174, 2020. doi: 10.18653/v1/2020.findings-emnlp.372.

Chaitanya K. Joshi, Fayao Liu, Xun Xu, Jie Lin, and Chuan-Sheng Foo. On representation knowledge distillation for graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):4656–4667, 2024. doi: 10.1109/TNNLS.2022.3223018.

Benjamin Kroposki, Andrey Bernstein, Jennifer King, Deepthi Vaidhynathan, Xinyang Zhou, Chin-Yao Chang, and Emiliano Dall'Anese. Autonomous energy grids: Controlling the future grid with large amounts of distributed energy resources. *IEEE Power and Energy Magazine*, 18(6):37–46, 2020. doi: 10.1109/MPE.2020.3014540.

Wenlong Liao, Birgitte Bak-Jensen, Jayakrishnan Radhakrishna Pillai, Yuelong Wang, and Yusen Wang. A review of graph neural networks and their applications in power systems. *Journal of Modern Power Systems and Clean Energy*, 10(2):345–360, 2022. doi: 10.35833/MPCE.2021.000058.

Nan Lin, Stavros Orfanoudakis, Nathan Ordonez Cardenas, Juan S. Giraldo, and Pedro P. Vergara. PowerFlowNet: Power flow approximation using message passing graph neural networks. *International Journal of Electrical Power & Energy Systems*, 160:110112, 2024. doi: 10.1016/j.ijepes.2024.110112.

Federico Milano, Florian Dörfler, Gabriela Hug, David J. Hill, and Gregor Verbič. Foundations and challenges of low-inertia systems. In *Power Systems Computation Conference (PSCC)*, pp. 1–25, 2018. doi: 10.23919/PSCC.2018.8450880.

Daniel Musekamp, Boshra Ariguib, Andrei Manolache, and Mathias Niepert. Distillation of foundation models for time-dependent PDEs. *arXiv preprint arXiv:2608.11937*, 2026. URL https://arxiv.org/abs/2608.11937.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021. URL https://arxiv.org/abs/2106.08295.

Rahul Nellikkath and Spyros Chatzivasileiadis. Physics-informed neural networks for AC optimal power flow. *Electric Power Systems Research*, 212:108412, 2022. doi: 10.1016/j.epsr.2022.108412.

Quang-Ha Ngo, Bang L. H. Nguyen, Tuyen V. Vu, Jianhua Zhang, and Tuan Ngo. Physics-informed graphical neural network for power system state estimation. *Applied Energy*, 358:122602, 2024. doi: 10.1016/j.apenergy.2023.122602.

Charilaos Papaioannou, Ioannis Tsantilas, Dimitris Giannakakos, Vasilis Michalakopoulos, Sotiris Pelekis, Vangelis Marinakis, Arsam Aryandoust, Antonello Monti, Ricardo J. Bessa, Pedro P. Vergara, Jochen Cremer, and Elissaios Sarmas. MxGPS: Multiplex graph transformers for a power grid foundation model. *arXiv preprint arXiv:2607.13763*, 2026. URL https://arxiv.org/abs/2607.13763.

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. URL https://arxiv.org/abs/2403.17660.

A. Puech, M. Mazzonelli, C. Cintas, T. R. Govindasamy, M. Mngomezulu, J. Weiss, M. Baù, A. Varbella, F. Mirallès, K. Kim, L. Xie, H. F. Hamann, E. Vos, and T. Brunschwiler. gridfm-datakit-v1: A Python library for scalable and realistic power flow and optimal power flow data generation. *arXiv preprint arXiv:2512.14658*, 2025. URL https://arxiv.org/abs/2512.14658.

Alban Puech, François Mirallès, Jonas Weiss, Vincent Mai, Alexandre Blondin Massé, Martin de Montigny, Thomas Brunschwiler, and Hendrik F. Hamann. Accelerating quasi-static time series simulations with foundation models. *e & i Elektrotechnik und Informationstechnik*, 141 (7–8):451–457, 2024. doi: 10.1007/s00502-024-01296-9.

Haseena Rahmath P, Vishal Srivastava, Kuldeep Chaurasia, Roberto G. Pacheco, and Rodrigo S. Couto. Early-exit deep neural network – a comprehensive survey. *ACM Computing Surveys*, 57, 2024. doi: 10.1145/3698767.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6550.

Maying Shen, Hongxu Yin, Pavlo Molchanov, Lei Mao, Jianna Liu, and Jose M. Alvarez. Structural pruning via latency-saliency knapsack. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2210.06659.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew Gordon Wilson. Does knowledge distillation really work? In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2106.05945.

Shyam A. Tailor, Javier Fernandez-Marques, and Nicholas D. Lane. Degree-Quant: Quantization-aware training for graph neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=NSBrFgJAHg.

Yijun Tian, Shichao Pei, Xiangliang Zhang, Chuxu Zhang, and Nitesh V. Chawla. Knowledge distillation on graphs: A survey. *ACM Computing Surveys*, 57(8):1–16, 2025. doi: 10.1145/3711121.

Anna Varbella, Damien Briens, Blazhe Gjorgiev, Giuseppe Alessio D'Inverno, and Giovanni Sansavini. Physics-informed GNN for non-linear constrained optimization: PINCO a solver for the AC-optimal power flow. *arXiv preprint arXiv:2410.04818*, 2024. URL https://arxiv.org/abs/2410.04818.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper/7181-attention-is-all-you-need.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2002.10957.

Bencheng Yan, Chaokun Wang, Gaoyang Guo, and Yunkai Lou. TinyGNN: Learning efficient graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1848–1856, 2020. doi: 10.1145/3394486.3403236.

Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Distilling knowledge from graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7074–7083, 2020. doi: 10.1109/CVPR42600.2020.00710.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1612.03928.

Shichang Zhang, Yozen Liu, Yizhou Sun, and Neil Shah. Graph-less neural networks: Teaching old MLPs new tricks via distillation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=4p6_5HBWPCw.

Zhen Zhao, Wenqi Huang, Zicheng Wang, Jiaxuan Hou, Peng Li, and Lei Bai. SenseFlow: A physics-informed and self-ensembling iterative framework for power flow estimation. *arXiv preprint arXiv:2505.12302*, 2025. URL https://arxiv.org/abs/2505.12302.

Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8): 1738–1762, 2019. doi: 10.1109/JPROC.2019.2918951.

# A NOTATION

| | |
|---|---|
| $f_T$ | Frozen teacher model. |
| $f_S$ | Student model trained by distillation. |
| $\mathcal{V}_b$ | Set of bus nodes in the batch graph. |
| $\mathcal{V}_g$ | Set of generator nodes in the batch graph. |
| $\mathcal{V}_\tau$ | Node set for node type $\tau$. |
| $\mathcal{T}$ | Set of matched node types, typically $\{\mathrm{bus}, \mathrm{gen}\}$. |
| $\mathcal{R}$ | Set of graph relation types used for attention matching. |
| $\mathcal{E}_r$ | Edge set for relation type $r$. |
| $\mathcal{M}$ | Set of matched student-teacher layer pairs $(\ell_S, \ell_T)$. |
| $\ell_S$ | Student layer index. |
| $\ell_T$ | Teacher layer index. |
| $\hat{\mathbf{y}}^{b}_{M,i}$ | Bus output predicted by model $M$ for bus node $i$. |
| $\hat{\mathbf{y}}^{g}_{M,j}$ | Generator output predicted by model $M$ for generator node $j$. |
| $\mathbf{h}^{\tau,(\ell)}_{M,i}$ | Hidden state of model $M$ at layer $\ell$ for node $i$ of type $\tau$. |
| $P_\tau^{(\ell_S,\ell_T)}$ | Trainable projection from student latent dimension to teacher latent dimension. |
| $r^{P,(\ell)}_{M,i}$ | Active-power residual for model $M$ at layer $\ell$ and bus $i$. |
| $r^{Q,(\ell)}_{M,i}$ | Reactive-power residual for model $M$ at layer $\ell$ and bus $i$. |
| $\rho^{(\ell)}_M$ | Mean bus-level active/reactive residual norm for model $M$ at layer $\ell$. |
| $\bar{\alpha}^{r,(\ell)}_{M,e}$ | Head-averaged Transformer attention weight for model $M$, relation $r$, edge $e$, layer $\ell$. |
| $\lambda_{\mathrm{out}}$ | Weight for output distillation. |
| $\lambda_{\mathrm{phys}}$ | Weight for physics-residual distillation. |
| $\lambda_{\mathrm{lat}}$ | Weight for latent-state distillation. |
| $\lambda_{\mathrm{att}}$ | Weight for attention distillation. |
| $\mathcal{L}_{\mathrm{GraphKit}}$ | Standard GraphKit training loss, including supervised and physics-informed terms. |