Solving Optimal Power Flow on a Data-Budget: Feature Selection on Smart Meter Data

Vassilis Kekatos, *Senior Member, IEEE*, Ridley Annin, *Student Member, IEEE*,
Manish K. Singh, *Senior Member, IEEE*, and Junjie Qin, *Member, IEEE*

Abstract—How much data is needed to optimally schedule distributed energy resources (DERs)? Does the distribution system operator (DSO) have to precisely know load demands and solar injections at each bus of the feeder to solve an optimal power flow (OPF)? This work exploits redundancies in OPF's structure and data to avoid communicating such data deluge, and explores the trade-off between data compression and grid's performance. We propose an OPF data distillation framework involving two steps. The DSO first collects OPF data from only a subset of nodes. The DSO subsequently reconstructs the complete OPF data from the partial ones, and feeds them into the OPF solver. Selecting and reconstructing OPF data may be performed to either maximize the fidelity of reconstructed OPF data, or maximize the fidelity of OPF solutions corresponding to reconstructed data. Under the first objective, OPF data distillation is posed as a sparsity-regularized convex problem. Under the second objective, it is posed as a sparsity-regularized bilevel program. Both problems are solved using accelerated proximal gradient (PGD) algorithms. Numerical tests corroborate that the bilevel formulation enhances fidelity and feasibility of reconstructed OPF solutions, and that OPF solutions can be approximated reasonably well even from partial data.

Index Terms—Linearized distribution flow, optimal meter placement, smart meter data compression, group lasso.

I. Introduction

With the rampant integration of DERs, distribution system operators (DSOs) need to dispatch resources optimally at increasingly finer spatiotemporal scales. Solving the OPF requires collecting a large amount of data in near real-time. Currently, smart meters report packetized load demand data hourly or daily. Nonetheless, to control DERs effectively, the DSO may have to communicate with all customers and collect their active and reactive power demands every few minutes. This technical specification may be hard to meet due to communication, data privacy, and cyber-attack considerations. This work leverages redundancies in OPF features and the structure of the OPF to approximately solve the OPF using only a carefully selected subset of load demands.

In transmission systems, the operator estimates voltages from grid measurements, infers loads by plugging voltages into the power flow equations, and feeds the inferred load values into real-time market operations. This workflow may not be replicated in real-time distribution systems operations due to a lack of observability at the grid edge. Under this setting, can DSOs control DERs based on partial grid data? The OPF can be viewed as a parametric optimization problem: Given load demands at all buses captured by vector θ, the OPF returns the optimal DER dispatch denoted by vector $\mathbf{x}(\theta)$. A key challenge in real-time DSO operations is that θ may be uncertain or incomplete. The OPF mapping $\theta \rightarrow \mathbf{x}$ has been extensively studied in the literature; see [1], [2] and references

therein. Uncertainty quantification studies how a probabilistic characterization postulated on θ propagates through the OPF [3]. Sensitivity analysis computes the Jacobian matrix of $\mathbf{x}$ with respect to θ; see [4, Ch. 6]. Stochastic or robust programming aims at securing system's performance when θ is not precisely known [5]. Reference [6] studies the value of data in grid operations and how obfuscation to preserve privacy can affect OPF solutions. Schemes for securing customer's private load demand information are devised in [7].

The focus here is on compressing θ to approximately solve the OPF. Kron/Ward reduction techniques from the 1950's and more recent ones aim at compressing the OPF problem itself via a reduced network model [8], [9], [10]. However, mapping results back to the actual system is non-trivial. Reference [11] applies principal component analysis (PCA) to a dataset of OPF solutions $\mathbf{x}$ so that a learning-to-optimize deep neural network (DNN) is trained to learn only a limited subspace of $\mathbf{x}$. Instead of $\mathbf{x}$, we aim to compress θ. References [12] and [13] suggest a joint learning and optimization approach, according to which OPF data are unknown and can only be predicted from primitive features ϕ, such as weather forecasts, historical trends, or partial OPF data. Rather than first inferring θ from ϕ and then mapping θ to $\mathbf{x}$ via the OPF or an ML model, [12] and [13] advocate learning the mapping from ϕ to $\mathbf{x}$. In our setting, the DSO has no access to ϕ and needs to decide which OPF data should be sampled in real time. Keeping the OPF module in the workflow ensures that data distillation does not affect the OPF solver so that the same OPF solver is reused regardless of whether the DSO collects more or less data.

The task of selecting relevant OPF data is reminiscent of *sparse learning* or *feature selection* in machine learning [14], wherein given *feature vector* θ, one tries to infer a target variable $\mathbf{x}(\theta)$ using only a subset of features. Although feature selection and reconstruction have been extensively studied in the ML context, we revisit them with a fresh look under the OPF lens. Clearly, selecting OPF data is closely related to the optimal placement of meters [15], [16], or actuators [17]. Rather than improving the grid's observability or controllability, we select data to support OPF operations.

The technical contributions of this work are on three fronts:
- *i)* We formulate the task of OPF data distillation to select the most influential data to optimally schedule DERs on a data budget (Section II);
- *ii)* If the goal is to improve the fidelity of OPF data, we adopt a greedy algorithm based on the discrete empirical interpolation method (DEIM) and a proximal gradient algorithm for solving a group-lasso-type convex optimization problem (Section III); and

iii) If the goal is to improve the fidelity of OPF solutions, we put forth a bilevel optimization program and develop a proximal gradient algorithm for solving it (Section IV).

The developed methods are evaluated on a single-phase simplification of the IEEE 37-bus benchmark feeder and the IEEE 123-bus multiphase feeder. We solved prototypical examples of the OPF that control reactive power injections from DERs to minimize losses while maintaining bus voltages within allowable limits. Grid loading conditions were simulated using a combination of synthetic and real-world load, solar, and electric vehicle data. The tests corroborate that fidelity improves when selecting more OPF features, OPF-aware data distillation yields better results in terms of approximating optimal DER schedules and complying with constraints, and OPF solutions can be approximated reasonably well given only a relatively small subset of OPF data. Conclusions and exciting open research directions are presented in Section VI. Partial preliminary results have been reported in the conference precursor of this work [18].

II. PROBLEM STATEMENT

A distribution system operator (DSO) wants to decide the optimal setpoints for reactive power injections by inverter-interfaced DERs. To this end, the operator routinely collects data across a feeder, which are subsequently used as parameters in an OPF. Such data may be of sizable volume as they include active injections by DERs as well as active and reactive demands from inflexible loads. Such data may be aggregated at the level of primary network buses. Under limited communication and/or sensing capabilities, the operator may not be able to possess all such data promptly to solve the OPF in real-time. In other words, if the OPF has to be solved every few minutes, the operator may be unable to communicate and read out data from every single bus of the primary network. A more realistic setup is that the operator can afford to install sensors only at K out of the N buses of a feeder. Hence, in real-time, the operator would communicate only with these K sensors and rely solely on their data feeds to solve the OPF. Under this setup, two questions arise: *i) How to sample data optimally?* and *ii) How to solve the OPF using partial data?*

To address these questions, we deal with a prototypical OPF version. Consider the task of optimally deciding the reactive power setpoints by DERs. Given re/active loads and active solar generation on each bus, the DSO intends to decide optimally on DER reactive power setpoints. Such setpoints could be found by solving an OPF that minimizes ohmic power losses on distribution lines while ensuring bus voltages lie within allowable limits. The OPF considers a linearized grid model of a single-phase radial feeder. Although the technical exposition considers this stylized version of the OPF, our proposed methodology applies readily to other settings; Section V includes numerical tests on a multiphase feeder.

Consider a single-phase feeder with $N+1$ buses, indexed by $n = 0, \ldots, N$. The substation bus is indexed by $n = 0$. Let v_n be the voltage magnitude, and $p_n + jq_n$ the complex power injection at bus n. Collect all but the substation injections and voltages in the N-length vectors $(\mathbf{p}, \mathbf{q}, \mathbf{v})$. Vectors $(\mathbf{p}, \mathbf{q})$

can be decomposed into inverter injections $(\mathbf{p}^g, \mathbf{q}^g)$ and load withdrawals $(\mathbf{p}^\ell, \mathbf{q}^\ell)$ as

$$\mathbf{p} = \mathbf{p}^g - \mathbf{p}^\ell \tag{1a}$$
$$\mathbf{q} = \mathbf{q}^g - \mathbf{q}^\ell. \tag{1b}$$

Without loss of generality, each bus is assumed to host at most one (aggregate) load and at most one DER.

We adopt an approximate grid model rather than the exact AC power flow equations. Modeling inaccuracies can be justified due to the *planning* rather than operational nature of our OPF data distillation task: The goal at this point is *not* to schedule DERs, but to find which data are most valuable for scheduling DERs. The approximate grid model is obtained upon linearizing the power flow equations at the flat voltage profile of unit magnitudes and zero angles [19]. Then, nodal voltage magnitudes depend linearly on power injections as

$$\mathbf{v}(\mathbf{p}, \mathbf{q}) \simeq \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q} + v_0 \mathbf{1} \tag{2}$$

where matrices $(\mathbf{R}, \mathbf{X})$ depend on the feeder topology and line impedances; v_0 is the substation voltage; and $\mathbf{1}_N$ is the vector of all-one entries.

To formulate the OPF, we would also need to express ohmic losses on lines as functions of $(\mathbf{p}, \mathbf{q})$. If we use the flat voltage profile again as the linearization point and pursue a first-order Taylor's series expansion, ohmic losses take the constant value of zero. If we use a second-order Taylor's series expansion, ohmic losses on all lines can be expressed as a convex quadratic function of power injections as [19]

$$L(\mathbf{p}, \mathbf{q}) \simeq 2\mathbf{p}^\top \mathbf{R}\mathbf{p} + 2\mathbf{q}^\top \mathbf{R}\mathbf{q}. \tag{3}$$

Based on (2) and (3), the optimal DER setpoints can be found as the solution to a convex quadratic program as

$$\min_{\mathbf{q}^g} \quad \mathbf{q}^\top \mathbf{R}\mathbf{q} \tag{4a}$$
$$\text{subject to (s.to)} \quad -\bar{\mathbf{v}} \leq \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q} \leq \bar{\mathbf{v}} \tag{4b}$$
$$-\bar{\mathbf{q}}^g \leq \mathbf{q}^g \leq \bar{\mathbf{q}}^g \quad \text{and} \quad (1) \tag{4c}$$

where $\bar{\mathbf{v}} = 0.03\mathbf{1}$. Problem (4) minimizes losses with respect to DER reactive setpoints $\mathbf{q}^g$. Constraint (4b) maintains voltage deviations within the allowable range of $\pm 3\%$ per unit (pu). To derive (4b), it has been tacitly assumed that the substation voltage is $v_0 = 1$ pu, so that $\mathbf{v} - v_0 \mathbf{1} \simeq \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q}$ from (2). Alternatively, the substation voltage v_0 could be included as a decision variable, in which case, constraint (4b) should be replaced by $0.97 \cdot \mathbf{1} \leq \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q} + v_0 \mathbf{1} \leq 1.03 \cdot \mathbf{1}$. Moreover, voltage regulators (either remotely controllable or operating autonomously driven by local settings) can also be included in the OPF as in [20]. We skip both extensions to keep the presentation succinct. Constraint (4c) limits inverter setpoints within reactive power ratings collected in vector $\bar{\mathbf{q}}_g$, assumed known to the DSO and fixed.

Under extreme loading conditions, limited VAR capacity, and/or imprecise OPF data $(\mathbf{p}^g, \mathbf{p}^\ell, \mathbf{q}^\ell)$, problem (4) may become infeasible. In other words, it may be impossible for DERs to maintain voltages within the specified limits. To deal with this possibility and comply with industry practice, voltage constraints can be converted to *soft constraints*: A

slack variable $s \geq 0$ is added on voltage constraints and is penalized in the objective as

$$\min_{\mathbf{q}^g, s} \quad \mathbf{q}^\top \mathbf{R} \mathbf{q} + \nu s^2 + \rho s \tag{5a}$$

$$\text{s.to} \quad -s\mathbf{1} - \bar{\mathbf{v}} \leq \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q} \leq \bar{\mathbf{v}} + s\mathbf{1} \tag{5b}$$

$$-\bar{\mathbf{q}}^g \leq \mathbf{q}^g \leq \bar{\mathbf{q}}^g. \tag{5c}$$

If the positive constants (ν, ρ) are selected sufficiently large, problem (5) exhibits the following neat property [19]. For feasible instances of (4), problem (5) yields optimal $s^* = 0$ and the $\mathbf{q}$-minimizers of the two problems coincide. On the other hand, for infeasible instances of (4), problem (5) returns $s^* > 0$ and its $\mathbf{q}$-minimizer satisfies voltage limits stretched out by s^*. In this way, problem (5) deals with feasible and infeasible instances of the OPF systematically. Reference [19] provides analytical bounds and practical heuristics on how to select ρ and ν appropriately.

Substituting (1) into (5) yields an optimization problem over $\mathbf{q}^g$ and s. For simplicity, the minimizer of (5) will be henceforth denoted by $\mathbf{x} = [\mathbf{q}^g; \ s]$ using a MATLAB-like notation. The OPF depends on $(\mathbf{R}, \mathbf{X})$ and *OPF data vector*

$$\boldsymbol{\theta} = \left[\mathbf{p}^g - \mathbf{p}^\ell; \ \ \mathbf{q}^\ell \right]. \tag{6}$$

The DSO would like to solve (5) for different values of $\boldsymbol{\theta}$ and/or $(\mathbf{R}, \mathbf{X})$. To keep the exposition uncluttered, we assume the feeder topology to remain fixed, and vary only the grid loading conditions $\boldsymbol{\theta}$. Either way, it is reasonable to assume $(\mathbf{R}, \mathbf{X})$ to be known to the DSO, whereas $\boldsymbol{\theta}$ is the communication-intensive component of (5). Even though $\boldsymbol{\theta}$ in (6) has been defined to be of dimension $2N$, some of the buses may be zero-injection buses, or their entries may be already known to the operator through other means. We will henceforth denote the actual length of $\boldsymbol{\theta}$ by P, where $P \leq 2N$. We slightly abuse notation and index the entries of $\boldsymbol{\theta}$ by subscript p as θ_p. We often denote the OPF minimizer as $\mathbf{x}(\boldsymbol{\theta})$ to emphasize its dependence on $\boldsymbol{\theta}$. To put it differently, the OPF is a *parametric optimization problem* and can be interpreted as a mapping from $\boldsymbol{\theta}$ to $\mathbf{x}$; see Fig. 1(a).

To solve (5), the DSO needs to know $\boldsymbol{\theta}$ in nearly real-time. This requirement may be impractical due to communication or privacy concerns. As a remedy, the operator may install sensors reading injections on a subset K out of P data sources. If a bus is instrumented with a sensor, it is reasonable to assume the operator has real-time access to the corresponding entries of $(\mathbf{p}_g - \mathbf{p}_\ell)$ and $\mathbf{q}_\ell$. The pertinent questions are:

q1) *How to select K out of P OPF data sources?*
q2) *Once K data sources have been selected, how to reconstruct a P-dimensional vector to feed into problem (5)?*

This task of *OPF data distillation* is visualized in Fig. 1(b). The operator selects K out of P entries of $\boldsymbol{\theta}$ to create vector $\tilde{\boldsymbol{\theta}}$. Given the selected OPF data $\tilde{\boldsymbol{\theta}}$, the DSO reconstructs a data vector $\hat{\boldsymbol{\theta}}$ to feed into the OPF. The reconstructed vector $\hat{\boldsymbol{\theta}}$ should be of the original dimension P. This ensures backward compatibility with the OPF solver and keeps the solver independent of the data distillation scheme.

An OPF data distillation scheme involves *data selection* and *data reconstruction*. The selection step can be expressed as $\tilde{\boldsymbol{\theta}} = \mathbf{S}^\top \boldsymbol{\theta}$, where $\mathbf{S}$ is a selection matrix with a single one-entry per column and zeros elsewhere. Vector $\tilde{\boldsymbol{\theta}}$ is a subvector of $\boldsymbol{\theta}$. If we limit ourselves to linear schemes, the reconstruction step can be expressed with the help of a $P \times K$ matrix $\mathbf{C}$ as $\hat{\boldsymbol{\theta}} = \mathbf{C}\tilde{\boldsymbol{\theta}}$. Combining the two steps into $\mathbf{W} = \mathbf{C}\mathbf{S}^\top$ yields:

$$\hat{\boldsymbol{\theta}} = \mathbf{C}\tilde{\boldsymbol{\theta}} = \mathbf{C}\mathbf{S}^\top \boldsymbol{\theta} = \mathbf{W}\boldsymbol{\theta}. \tag{7}$$

OPF data distillation amounts to finding $(\mathbf{C}, \mathbf{S})$. These two matrices can be designed offline using a dataset of T grid loading scenarios of OPF data, arranged as columns of the $P \times T$ matrix

$$\boldsymbol{\Theta} = [\boldsymbol{\theta}_1 \ \boldsymbol{\theta}_2 \ \cdots \ \boldsymbol{\theta}_T]. \tag{8}$$

Once designed, matrices $(\mathbf{C}, \mathbf{S})$ can be held constant over a longer operational period depending on the application or until the statistics of OPF data change from the statistics of $\boldsymbol{\Theta}$.

The training OPF data have been centered and scaled so each row of $\boldsymbol{\Theta}$ is zero-mean and unit-variance. Such preprocessing is necessary to render all data sources tentatively equally important. Otherwise, an almost time-invariant load demand of 1 pu may seem more important than a highly time-variable load demand of 0.5 pu. All subsequent approaches operate on the preprocessed normalized OPF data $\boldsymbol{\Theta}$. Once normalized OPF data have been distilled, the data fed into the OPF are returned to their original (non-normalized) form.

Matrices $(\mathbf{C}, \mathbf{S})$ can be selected to optimize a meaningful metric. One idea is to seek $(\mathbf{C}, \mathbf{S})$ so $\hat{\boldsymbol{\theta}}$ is a good approximation of $\boldsymbol{\theta}$. Such distillation schemes focus on the fidelity of OPF data and will be henceforth termed *Type-1 schemes*. Taking an end-to-end vantage point, a second idea seeks $(\mathbf{C}, \mathbf{S})$ so that $\mathbf{x}(\hat{\boldsymbol{\theta}})$ is a good approximant of $\mathbf{x}(\boldsymbol{\theta})$. Such schemes focus on the effect of distilled data on OPF decisions and will be termed *Type-2 schemes*. Type-1 schemes are developed in Section III and Type-2 schemes in Section IV.

III. TYPE-1 OPF DATA DISTILLATION

This section presents Type-1 schemes so that distilled data $\hat{\boldsymbol{\theta}}$ approximates well original data $\boldsymbol{\theta}$ in the training dataset.

A. OPF Data Compression using PCA

Given dataset $\boldsymbol{\Theta}$, the first attempt could be to apply a dimensionality-reduction technique, such as principal component analysis (PCA). This seems intuitive as some entries of $\boldsymbol{\theta}_t$ may exhibit linear dependencies, e.g., due to solar irradiance. To this end, we perform an eigenvalue decomposition (EVD) on the *data covariance matrix*

$$\mathbf{C}_\theta = \frac{1}{T}\boldsymbol{\Theta}\boldsymbol{\Theta}^\top = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top \tag{9}$$

and collect the eigenvectors related to the largest K eigenvalues as columns of matrix $\mathbf{U}_K$. A 'compressed' version of $\boldsymbol{\theta}_t$ can be obtained as $\tilde{\boldsymbol{\theta}} = \mathbf{U}_K^\top \boldsymbol{\theta}_t$, and a reconstructed one as

$$\hat{\boldsymbol{\theta}}_t = \mathbf{U}_K \tilde{\boldsymbol{\theta}}_t = \mathbf{U}_K \mathbf{U}_K^\top \boldsymbol{\theta}_t = \mathbf{W}_{\text{PCA}} \boldsymbol{\theta}_t \tag{10}$$

where $\mathbf{W}_{\text{PCA}} = \mathbf{U}_K \mathbf{U}_K^\top$ is a projection matrix and is known to be the minimizer of

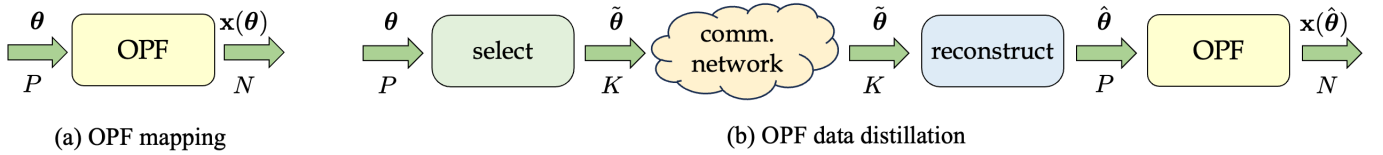$$\min_{\mathbf{W}} \{ f_1(\mathbf{W}) : \text{rank}(\mathbf{W}) = K \} \tag{11}$$

Fig. 1. *(a) OPF mapping:* The OPF as a parametric optimization problem. The OPF maps input data $\boldsymbol{\theta} \in \mathbb{R}^P$ to minimizers $\mathbf{x} \in \mathbb{R}^N$. *(b) OPF data distillation:* The operator first selects K out of P data to form $\tilde{\boldsymbol{\theta}} \in \mathbb{R}^K$. The entries of $\tilde{\boldsymbol{\theta}}$ are communicated from each selected bus to the operator over a communication network. The operator uses $\tilde{\boldsymbol{\theta}}$ to reconstruct an OPF parameter vector $\hat{\boldsymbol{\theta}}$ of length P, equal to the original parameter vector $\boldsymbol{\theta}$. The operator feeds $\hat{\boldsymbol{\theta}}$ into the OPF solver to find the DER dispatch $\mathbf{x}(\hat{\boldsymbol{\theta}})$. The OPF solver remains the same. How to optimally design the *selection function* (green) and the *reconstruction function* (blue)? These two functions are designed once and offline, and then used in real time. Should they be designed to minimize $\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|$ or $\|\mathbf{x}(\boldsymbol{\theta}) - \mathbf{x}(\hat{\boldsymbol{\theta}})\|$? See Sections III and IV, respectively.

where $f_1(\mathbf{W})$ is the data fitting cost over the training dataset

$$f_1(\mathbf{W}) = \frac{1}{2T} \sum_{t=1}^{T} \|\boldsymbol{\theta}_t - \mathbf{W}\boldsymbol{\theta}_t\|^2 = \frac{1}{2T}\|\boldsymbol{\Theta} - \mathbf{W}\boldsymbol{\Theta}\|_F^2. \quad (12)$$

Although $\mathbf{W}_{\text{PCA}}\boldsymbol{\Theta}$ is the best rank-K approximation of $\boldsymbol{\Theta}$ in the Frobenius norm, matrix $\mathbf{U}_K^\top$ cannot be used for data selection because it is dense in general. Then, computing $\tilde{\boldsymbol{\theta}}_t = \mathbf{U}_K^\top \boldsymbol{\theta}_t$ would require knowing the entire $\boldsymbol{\theta}_t$. To put it differently, matrix $\mathbf{W}_{\text{PCA}}$ does not comply with the format $\mathbf{CS}^\top$ of (7). However, the reconstruction error $\mathbf{W}_{\text{PCA}}$ attains

$$f_1(\mathbf{W}_{\text{PCA}}) = \frac{1}{2T}\|\boldsymbol{\Theta} - \mathbf{W}_{\text{PCA}}\boldsymbol{\Theta}\|_F^2 \quad (13)$$

serves as a benchmark for Type-1 data distillation schemes as PCA minimizes $f_1(\mathbf{W})$ over all rank-K matrices.

B. OPF Data Selection using DEIM

To achieve data selection, we propose using the *discrete empirical interpolation method* (DEIM) [21]. By design, DEIM complies with the format of (7). Given a selection matrix $\mathbf{S}$, DEIM sets matrix $\mathbf{C}$ in (7) as $\mathbf{C} = \mathbf{U}_K(\mathbf{S}^\top \mathbf{U}_K)^{-1}$, where $\mathbf{U}_K$ carries again the top-K eigenvectors of $\mathbf{C}_\theta$ from (10) and assuming $\mathbf{S}^\top \mathbf{U}_K$ is invertible. DEIM reconstructs data as

$$\hat{\boldsymbol{\theta}}_t = \mathbf{U}_K(\mathbf{S}^\top \mathbf{U}_K)^{-1}\tilde{\boldsymbol{\theta}}_t = \mathbf{W}_{\text{DEIM}}\boldsymbol{\theta}_t$$

where $\mathbf{W}_{\text{DEIM}} = \mathbf{U}_K(\mathbf{S}^\top \mathbf{U}_K)^{-1}\mathbf{S}^\top$ is a projection matrix as $\mathbf{W}_{\text{DEIM}}^2 = \mathbf{W}_{\text{DEIM}}$. Although both $\mathbf{W}_{\text{PCA}}$ and $\mathbf{W}_{\text{DEIM}}$ are rank-K projection matrices, the latter comes with the neat feature that $\hat{\boldsymbol{\theta}}_t$ is computed using *only* K entries of $\boldsymbol{\theta}_t$. Moreover, for every vector in the dataset, it holds that $\mathbf{S}^\top\hat{\boldsymbol{\theta}}_t = \mathbf{S}^\top\boldsymbol{\theta}_t$. In other words, the reconstructed $\hat{\boldsymbol{\theta}}_t$ perfectly matches *(interpolates)* $\boldsymbol{\theta}_t$ at the K selected entries. On the other hand, matrix $\mathbf{W}_{\text{DEIM}}$ is clearly suboptimal in terms of solving (11). Therefore, its data fitting cost is lower bounded as $f_1(\mathbf{W}_{\text{DEIM}}) \geq f_1(\mathbf{W}_{\text{PCA}})$. Interestingly enough, DEIM's data fitting cost is also upper bounded as [21]

$$f_1(\mathbf{W}_{\text{PCA}}) \leq f_1(\mathbf{W}_{\text{DEIM}}) \leq \|(\mathbf{S}^\top \mathbf{U}_K)^{-1}\|^2 \cdot f_1(\mathbf{W}_{\text{PCA}})$$

where $\|\cdot\|$ is the spectral matrix norm. This bound holds for all selection matrices $\mathbf{S}$ with invertible $\mathbf{S}^\top \mathbf{U}_K$. Since $\mathbf{U}_K$ is found via EVD, matrix $\mathbf{S}$ can be selected to minimize $\|(\mathbf{S}^\top \mathbf{U}_K)^{-1}\|$ and bring $f_1(\mathbf{W}_{\text{DEIM}})$ closer to $f_1(\mathbf{W}_{\text{PCA}})$.

Nonetheless, minimizing $\|(\mathbf{S}^\top \mathbf{U}_K)^{-1}\|$ over $\mathbf{S}$ is nontrivial. As a practical solution, DEIM has a greedy algorithm for selecting $\mathbf{S}$ to minimize approximately $\|(\mathbf{S}^\top \mathbf{U}_K)^{-1}\|$;

see [21] for details. Bounds on the suboptimality of DEIM's greedy algorithm have been derived in [21]. Empirical tests on different application domains demonstrate that DEIM performs significantly better than the analytically derived bounds. Heed that DEIM's greedy algorithm first requires computing $\mathbf{U}_K$ by applying an EVD on $\mathbf{C}_\theta$. Critically, the computational complexity of the greedy algorithm does not exceed the complexity of EVD. Therefore, DEIM has the same complexity order as computing the leading K EVD components.

C. OPF Data Selection using Group Lasso

As an alternative to DEIM, we next formulate OPF data distillation as a type of a *group lasso (GL) problem* [14]. The reason for introducing yet another method is to bring more flexibility to Type-1 and Type-2 designs, as will be clear soon. Our GL-based, Type-1 data distillation entails solving the convex program:

$$\mathbf{W}_{\text{GL}} = \arg\min_{\mathbf{W}} f_1(\mathbf{W}) + \lambda_1 g(\mathbf{W}) \quad (14)$$

where $g(\mathbf{W}) = \sum_{p=1}^{P} \|\mathbf{w}_p\|$, vector $\mathbf{w}_p$ is the p-th column of $\mathbf{W}$, and $\lambda_1 > 0$ is a tunable parameter.

The cost in (14) consists of two terms. The first term is the data fitting cost of PCA. The second term $g(\mathbf{W})$ is a regularization term that promotes column-sparse solutions [14]. In other words, the minimizer $\mathbf{W}_{\text{GL}}$ will have many zero columns with increasing λ_1. Expanding (7), we get that the reconstructed vector is

$$\hat{\boldsymbol{\theta}} = \sum_{p=1}^{P} \theta_p \mathbf{w}_p.$$

If column $\mathbf{w}_p = \mathbf{0}$, OPF feature θ_p does not contribute to $\hat{\boldsymbol{\theta}}$.

GL offers more flexibility over DEIM. According to (6), active injection $p_n = p_n^g - p_n^\ell$ at bus n corresponds to the n-th entry of $\boldsymbol{\theta}$, while its reactive injection $q_n = -q_n^\ell$ corresponds to the $(n+N)$-th entry of $\boldsymbol{\theta}$. Designing $\mathbf{W}$ according to (14) may give us $\mathbf{w}_n \neq \mathbf{0}$ and $\mathbf{w}_{n+N} = \mathbf{0}$. The same could happen with DEIM. Such design means the operator collects p_n, but not q_n. This may be impractical. If the operator is to install a sensor and communicate with bus n, they might as well either select both (p_n, q_n) or none. This specification can be satisfied by GL (yet not DEIM) by simply placing columns $\mathbf{w}_n$ and $\mathbf{w}_{n+N}$ into the same group and replacing norms

$$\|\mathbf{w}_n\| + \|\mathbf{w}_{n+N}\| \quad \longrightarrow \quad \left\| \begin{bmatrix} \mathbf{w}_n \\ \mathbf{w}_{n+N} \end{bmatrix} \right\|$$

Algorithm 1 APG for Type-1 OPF Data Distillation

Input: OPF input data $\{\boldsymbol{\theta}_t\}_{t=1}^T$ stored in matrix $\boldsymbol{\Theta}$.
Output: OPF data distillation matrix $\mathbf{W}$.
 1: Initialize $\mathbf{W}^1 = \mathbf{W}^0$ from standard normal distribution.
 2: Initialize $\alpha_1 = 1$ and $\alpha_0 = 0$; set μ and λ_1.
 3: **for** $i = 1, 2, \ldots,$ **do**
 4:     Compute $\mathbf{Y}^i$ from (15a)–(15b) via (16).
 5:     Compute the groups of $\mathbf{W}^{i+1}$ from (15c) via (17).
 6:     Compute α_{i+1} from (15d).
 7: **end for**

---

in $g(\mathbf{W})$. GL offers additional flexibility as it can be modified to yield Type-2 designs, as discussed later in Section IV.

Although (14) can be solved using interior point-based solvers as a second-order cone program (SOCP), such solvers do not scale favorably with the problem size. To improve computational complexity, we devise a first-order algorithm called *accelerated proximal gradient* (APG) to solve (14). Even though an APG algorithm has been developed for the general GL problem before [22], it is adopted to (14) and reviewed here to ease the exposition of Type-2 schemes.

APG is an extension of accelerated gradient descent when the cost of an optimization problem consists of a differentiable and a non-differentiable term [23]. For our task in (14), the differentiable term is $f_1(\mathbf{W})$, and the non-differentiable term is $g(\mathbf{W})$. Let APG iterations be indexed by i and $\mathbf{W}^i$ denote the estimate of minimizer $\mathbf{W}$ at iteration i. Each iteration involves four steps [23]:

$$\bar{\mathbf{W}}^i = \mathbf{W}^i + \frac{\alpha_{i-1} - 1}{\alpha_i}\left(\mathbf{W}^i - \mathbf{W}^{i-1}\right) \qquad (15a)$$

$$\mathbf{Y}^i = \bar{\mathbf{W}}^i - \mu_i \nabla_{\mathbf{W}} f_1(\bar{\mathbf{W}}^i) \qquad (15b)$$

$$\mathbf{W}^{i+1} = \arg\min_{\mathbf{W}} \ \lambda_1 g(\mathbf{W}) + \frac{1}{2\mu_i}\|\mathbf{W} - \mathbf{Y}^i\|_F^2 \qquad (15c)$$

$$\alpha_{i+1} = \frac{1 + \sqrt{4\alpha_i^2 + 1}}{2} \qquad (15d)$$

Step (15b) coincides with a gradient descent step if we were to minimize $f_1(\mathbf{W})$ alone. It uses $\mu_i > 0$ as a step size. However, the gradient descent step is not performed at $\mathbf{W}^i$ but $\bar{\mathbf{W}}^i$, computed as a linear combination of $\mathbf{W}^i$ and $\mathbf{W}^{i-1}$ in step (15a). The linear combination coefficients change across iterations according to (15d). After some mundane algebra, the gradient needed in (15b) can be found to be

$$\nabla_{\mathbf{W}} f_1(\mathbf{W}) = (\mathbf{W} - \mathbf{I})\mathbf{C}_\theta \qquad (16)$$

where $\mathbf{C}_\theta$ is the OPF data covariance defined in (9).

The output $\mathbf{Y}^i$ of the gradient descent step is then applied to the *proximal operator* on $\mathbf{Y}^i$ to get $\mathbf{W}^{i+1}$. The proximal operator is defined as the solution to the optimization in (15c). Problem (15c) can be solved separately for each $\mathbf{w}_p$. In fact, each block $\mathbf{w}_p$ can be updated in closed form as [22]:

$$\mathbf{w}_p^{i+1} = \text{prox}\left(\mathbf{y}_p^i; \lambda_1 \mu_i\right). \qquad (17)$$

Operator prox takes a vector $\mathbf{x}$ and a scalar β, and returns:

$$\text{prox}(\mathbf{x}; \beta) = \begin{cases} \left(1 - \frac{\beta}{\|\mathbf{x}\|}\right)\mathbf{x} & , \ \|\mathbf{x}\| \geq \beta \\ \mathbf{0} & , \ \text{otherwise.} \end{cases} \qquad (18)$$

From (17)–(18), if $\|\mathbf{y}_p^i\| \geq \lambda_1 \mu_i$, vector $\mathbf{w}_p^{i+1}$ is set to a scaled-down version of $\mathbf{y}_p^i$; otherwise, it is set to zero. This hints at why $g(\mathbf{W})$ effects group sparsity on $\mathbf{W}_{\text{GL}}$. Step (15b) applies regardless of whether the entries of $\mathbf{W}$ are grouped in columns or other groups as long as groups do not share common entries. The APG algorithm is tabulated as Algorithm 1. If $\nabla_{\mathbf{W}} f_1$ is L-smooth, the APG iterations with constant step size $\mu_i = \mu \leq 1/L$ converge to the minimum cost at a rate of $\mathcal{O}(1/i^2)$; see [23].

Some comments on λ_1 are due. For $\lambda_1 = 0$, the GL solution $\mathbf{W}_{\text{GL}}$ coincides with $\mathbf{W}_{\text{PCA}}$, which is dense in general. As λ_1 increases, more columns of $\mathbf{W}_{\text{GL}}$ become zero monotonically. Lemma 1 provides an upper bound $\bar{\lambda}_1$ on λ_1 beyond which the minimizer of (14) becomes zero; see the appendix for a proof. By solving (14) for varying $\lambda_1 \in [0, \bar{\lambda}_1]$, we can make $\mathbf{W}_{\text{GL}}$ have exactly K non-zero columns using bisection.

Lemma 1. *Let $\bar{\lambda}_1 = \max_p \|\mathbf{c}_p\|$, where $\mathbf{c}_p$ is the p-th column of covariance matrix $\mathbf{C}_\theta$. The zero matrix is a minimizer of* (14) *if and only if $\lambda_1 \geq \bar{\lambda}_1$.*

The GL approach is known to successfully identify the sparsity support (non-zero columns) of $\mathbf{W}$. It is nonetheless also known that the GL solution is a biased estimate of the column-sparse $\mathbf{W}$ yielding the minimum fitting cost $f_1(\mathbf{W})$ alone. This is well understood for the general GL setting; our numerical tests indicate that the claim carries over to the OPF distillation task. As a remedy, we can employ a *two-stage GL approach*: First, use GL only to select K OPF features (columns of $\mathbf{W}$). Based on the sparsity pattern of $\mathbf{W}_{\text{GL}}$, we determine the selection matrix $\mathbf{S}$ in (7) to the value of $\mathbf{S}_{\text{GL}}$. Second, design the reconstruction matrix $\mathbf{C}$ via a standard least-squares fit as:

$$\begin{aligned} \mathbf{C}_{\text{GL2}} &= \arg\min_{\mathbf{C}} \|\boldsymbol{\Theta} - \mathbf{C}\mathbf{S}_{\text{GL}}^\top \boldsymbol{\Theta}\|_F^2 \\ &= \boldsymbol{\Theta}\boldsymbol{\Theta}^\top \mathbf{S}_{\text{GL}}(\mathbf{S}_{\text{GL}}^\top \boldsymbol{\Theta}\boldsymbol{\Theta}^\top \mathbf{S}_{\text{GL}})^{-1} \\ &= \mathbf{C}_\theta \mathbf{S}_{\text{GL}}(\mathbf{S}_{\text{GL}}^\top \mathbf{C}_\theta \mathbf{S}_{\text{GL}})^{-1}. \end{aligned}$$

The two-stage GL (GL2) returns the distillation matrix:

$$\mathbf{W}_{\text{GL2}} = \mathbf{C}_{\text{GL2}}\mathbf{S}_{\text{GL}}^\top = \mathbf{C}_\theta \mathbf{S}_{\text{GL}}(\mathbf{S}_{\text{GL}}^\top \mathbf{C}_\theta \mathbf{S}_{\text{GL}})^{-1}\mathbf{S}_{\text{GL}}^\top.$$

Type-1 methods are agnostic to the fact that distilled OPF data will eventually be used to produce OPF solutions and that we are primarily interested in OPF solutions rather than OPF data per se. To this end, we next explore Type-2 approaches.

IV. TYPE-2 OPF DATA DISTILLATION

The ultimate goal of replacing $\boldsymbol{\theta}$ by $\hat{\boldsymbol{\theta}} = \mathbf{W}\boldsymbol{\theta}$ is *not* to approximate $\boldsymbol{\theta}$ closely. The goal is to design $\hat{\boldsymbol{\theta}}$ so that when fed into the OPF, it yields an OPF minimizer close to the original minimizer. To develop Type-2 designs, suppose the operator has access to a dataset $\mathcal{D} = \{(\boldsymbol{\theta}_t, \mathbf{x}_t)\}_{t=1}^T$ of T representative OPF scenarios paired with their OPF minimizers.

We use the shorthand notation $\mathbf{x}_t = \mathbf{x}(\boldsymbol{\theta}_t)$ to denote the OPF minimizer of (5) given $\boldsymbol{\theta}_t$. We use a similar notation for the OPF minimizer given the reconstructed data as

$$\hat{\mathbf{x}}_t = \mathbf{x}(\hat{\boldsymbol{\theta}}_t) = \mathbf{x}(\mathbf{W}\boldsymbol{\theta}_t) \quad \text{for } t = 1, \ldots, T. \qquad (19)$$

We seek a column-sparse $\mathbf{W}$ so that $\hat{\mathbf{x}}_t$ is close to $\mathbf{x}_t$ for all t. Paralleling (12), we define the *OPF fitting error*:

$$f_2(\mathbf{W}) = \frac{1}{2T} \sum_{t=1}^{T} \|\mathbf{x}_t - \hat{\mathbf{x}}_t(\mathbf{W})\|^2. \tag{20}$$

Minimizing $f_2(\mathbf{W})$ over a column-sparse $\mathbf{W}$ is not amenable to a DEIM-type approach, but to a GL-type approach. We propose a Type-2 GL approach that we term *bilevel group lasso* (BGL) that finds $\mathbf{W}$ as the minimizer of:

$$\mathbf{W}_{\text{BGL}} = \arg\min_{\mathbf{W}} F_2(\mathbf{W}) = f_2(\mathbf{W}) + \lambda_2 g(\mathbf{W}) \tag{21a}$$

$$\text{s.to } \hat{\mathbf{x}}_t \text{ minimizes (5) for } \hat{\boldsymbol{\theta}}_t = \mathbf{W}\boldsymbol{\theta}_t \ \forall t. \tag{21b}$$

Like GL, the minimizer $\mathbf{W}_{\text{BGL}}$ becomes more column-sparse for larger values of $\lambda_2 > 0$. The fitting term $f_2(\mathbf{W})$ aims at bringing $\hat{\mathbf{x}}_t$ close to $\mathbf{x}_t$. Recall that $\mathbf{x}_t$'s have been already computed and given as part of dataset $\mathcal{D}$. On the other hand, each $\hat{\mathbf{x}}_t$ is the minimizer of the OPF fed by $\hat{\boldsymbol{\theta}}_t = \mathbf{W}\boldsymbol{\theta}_t$, and so $\hat{\mathbf{x}}_t$ depends on $\mathbf{W}$. This is a *bilevel optimization program*. The total cost $F_2(\mathbf{W})$ of the outer optimization over $\mathbf{W}$ is determined by the minimizers of T inner optimization problems over $\mathbf{x}_t$'s. Each inner problem depends parametrically on $\mathbf{W}$.

Problem (21) is non-convex. Bilevel programs are often handled by replacing every inner problem with its optimality conditions [24]. Unfortunately, such an approach would lead to a large-scale mixed-integer nonlinear program (MINLP) that scales unfavorably with N and T. Motivated by advancements in meta-learning, we solve (21) using an APG algorithm. Unfortunately, when it comes to non-convex problems, the standard APG algorithm is not guaranteed to converge to a *critical point*, that is a point where the cost subdifferential contains zero (zero-gradient point if the cost is differentiable). To resolve this issue, we use a modification of APG proposed in [23, Alg. 2]. Although the modified APG algorithm is slightly more complex, it is guaranteed to converge to a critical point even for non-convex problems. The method is tabulated as Algorithm 2. Its differences to Alg. 1 are discussed next.

Compared to Algorithm 1, Algorithm 2 first updates $\bar{\mathbf{W}}^i$ using three rather than two points as shown in Step 5. It then performs a gradient descent step based on $\nabla_{\mathbf{W}} f_2$ evaluated at $\bar{\mathbf{W}}^i$. The gradient $\nabla_{\mathbf{W}} f_2$ can be computed as explained next and established in the appendix.

Lemma 2. *Let $\nabla_{\hat{\boldsymbol{\theta}}_t} \hat{\mathbf{x}}_t$ be the Jacobian of the OPF minimizer $\hat{\mathbf{x}}_t$ given reconstructed OPF data $\hat{\boldsymbol{\theta}}_t = \mathbf{W}\boldsymbol{\theta}_t$ for all t, then*

$$\nabla_{\mathbf{W}} f_2(\mathbf{W}) = \frac{1}{T} \sum_{t=1}^{T} (\nabla_{\hat{\boldsymbol{\theta}}_t} \hat{\mathbf{x}}_t)^\top (\hat{\mathbf{x}}_t - \mathbf{x}_t) \boldsymbol{\theta}_t^\top.$$

To take the gradient descent step, Algorithm 2 needs to solve T OPFs using the reconstructed data $\bar{\boldsymbol{\theta}}_t^i = \bar{\mathbf{W}}^i \boldsymbol{\theta}_t$ computed from Step 6. Along with the minimizer, for each OPF, Algorithm 2 must compute the Jacobian matrix of the OPF minimizer with respect to the given OPF data. For Step 6, the minimizer is $\bar{\mathbf{x}}_t = \mathbf{x}(\bar{\boldsymbol{\theta}}_t)$ and the OPF data is $\bar{\boldsymbol{\theta}}_t$. Matrix $\nabla_{\bar{\boldsymbol{\theta}}_t} \bar{\mathbf{x}}_t$ is an $N \times P$ matrix carrying the sensitivities (partial derivatives) of $\bar{\mathbf{x}}_t$ with respect to $\bar{\boldsymbol{\theta}}_t$. Given the optimal

---

Algorithm 2 APG for Type-2 OPF Data Distillation

Input: Dataset of OPF input/minimizers $\mathcal{D} = \{(\boldsymbol{\theta}_t, \mathbf{x}_t)\}_{t=1}^{T}$.
Output: OPF data distillation matrix $\mathbf{W}$.
1: Initialize $\mathbf{Z}^1 = \mathbf{W}^1 = \mathbf{W}^0$ and evaluate $c_1 = F_2(\mathbf{W}^1)$.
2: Initialize $\alpha_1 = 1$, $\alpha_0 = 0$, $\eta \in [0, 1)$, $\delta > 0$, and $q_1 = 1$.
3: Select step sizes μ and $\bar{\mu}$, and parameter λ_2.
4: **for** $i = 1, 2, \ldots,$ **do**
5:     Update $\bar{\mathbf{W}}^i = \frac{\alpha_i - 1}{\alpha_i} \mathbf{W}^i + \frac{\alpha_{i-1}}{\alpha_i} \mathbf{Z}^i - \frac{\alpha_{i-1}-1}{\alpha_i} \mathbf{W}^{i-1}$.
6:     Run the OPF for reconstructed data $\bar{\boldsymbol{\Theta}}^i = \bar{\mathbf{W}}^i \boldsymbol{\Theta}$.
7:     Update $\bar{\mathbf{Y}}^i$ as $\bar{\mathbf{Y}}^i = \bar{\mathbf{W}}^i - \bar{\mu}_i \nabla_{\bar{\mathbf{W}}} f_2(\bar{\mathbf{W}}^i)$.
8:     Update blocks of $\mathbf{Z}^{i+1}$ as $\mathbf{z}_p^{i+1} = \text{prox}\left(\bar{\mathbf{y}}_p^i; \lambda_2 \bar{\mu}_i\right)$.
9:     **if** $F_2(\mathbf{Z}^{i+1}) \leq c_i - \delta \|\mathbf{Z}^{i+1} - \bar{\mathbf{W}}^i\|_F^2$, **then**
10:        $\mathbf{W}^{i+1} = \mathbf{Z}^{i+1}$.
11:     **else**
12:        Run the OPF for reconstructed data $\hat{\boldsymbol{\Theta}}^i = \mathbf{W}^i \boldsymbol{\Theta}$.
13:        Update $\mathbf{Y}^i$ as $\mathbf{Y}^i = \mathbf{W}^i - \mu_i \nabla_{\mathbf{W}} f_2(\mathbf{W}^i)$.
14:        Update blocks of $\bar{\mathbf{Z}}^{i+1}$ as $\bar{\mathbf{z}}_p^{i+1} = \text{prox}\left(\mathbf{y}_p^i; \lambda_2 \mu_i\right)$.
15:        $\mathbf{W}^{i+1} = \begin{cases} \mathbf{Z}^{i+1} & , \text{if } F_2(\mathbf{Z}^{i+1}) \leq F_2(\bar{\mathbf{Z}}^{i+1}) \\ \bar{\mathbf{Z}}^{i+1} & , \text{otherwise.} \end{cases}$
16:     **end if**
17:     Update $\alpha_{i+1} = \left(1 + \sqrt{4\alpha_i^2 + 1}\right)/2$.
18:     Update $q_{i+1} = \eta q_i + 1$.
19:     Update $c_{i+1} = \frac{\eta q_i c_i + F_2(\mathbf{W}^{i+1})}{q_{i+1}}$.
20: **end for**

---

primal/dual solutions of the OPF, matrix $\nabla_{\bar{\boldsymbol{\theta}}_t} \bar{\mathbf{x}}_t$ can be readily computed by solving of system of linear equations [25], [1].

Subsequently, Algorithm 2 applies the proximal operator on $\bar{\mathbf{Y}}^i$ to compute $\mathbf{Z}^{i+1}$ in Step 8. Algorithm 1 would complete the iteration here and return $\mathbf{Z}^{i+1}$ as $\mathbf{W}^{i+1}$. Algorithm 2 does so only if $\mathbf{Z}^{i+1}$ attains a sufficient decrease in total cost as detailed in Step 9. Note that evaluating $F_2(\mathbf{Z}^{i+1})$ entails solving another batch of T OPFs given data $\mathbf{Z}^{i+1}\boldsymbol{\theta}_t$ for all t. If $\mathbf{Z}^{i+1}$ does not achieve sufficient decrease in F_2, the algorithm tests if a proximal gradient step on the original $\mathbf{W}^i$ would offer a lower total cost than $\mathbf{Z}^{i+1}$. To this end, the algorithm reconstructs data using $\mathbf{W}^i$, takes a gradient descent step on $\mathbf{W}^i$ to find $\mathbf{Y}^i$, and applies the proximal operator on $\mathbf{Y}^i$ to find $\bar{\mathbf{Z}}^{i+1}$ in Steps 12–14. Step 13 does not require solving another batch of T OPFs because the OPFs corresponding to $\mathbf{W}^i$ have already been solved in the previous APG iteration, when the current $\mathbf{W}^i$ was then termed $\mathbf{W}^{i+1}$. On the other hand, to evaluate $F_2(\bar{\mathbf{Z}}^{i+1})$ in Step 15, the algorithm does have to solve another batch of T OPFs corresponding to data $\bar{\mathbf{Z}}^{i+1}\boldsymbol{\theta}_t$ for all t. Sequence α_i is updated in Step 17 exactly as in Algorithm 1. The sequence c_i needed in the sufficient decrease condition of Step 9 is updated in Steps 18–19.

Algorithm 2 solves three batches of T OPFs, one for each one of Steps 7, 9, and 15. Interestingly, if the OPF is a linear/quadratic program as in (4), techniques from multiparametric programming (MPP) can speed up the tasks of finding the minimizers and their sensitivities for a batch of T OPF problem instances. For example, in [19], we were able to speed up the running times of OPF batches by an order of magnitude. Details on sensitivity analysis of the OPF and MPP are omitted as they are quite technical and fall beyond this work's scope.

The next lemma (shown in the appendix) provides a necessary condition on λ_2 for a critical point of BGL to be zero.

Lemma 3. *Let $\bar{\lambda}_2 = \max_p \|\mathbf{k}_p\|$, where $\mathbf{k}_p$ is the p-th column of matrix $\mathbf{K} = \frac{1}{T}(\nabla_{\boldsymbol{\theta}}\mathbf{x}(\mathbf{0}))^{\top}\sum_{t=1}^{T}(\mathbf{x}(\mathbf{0}) - \mathbf{x}_t)\boldsymbol{\theta}_t^{\top}$ and $\mathbf{x}(\mathbf{0})$ is the solution of (5) for $\boldsymbol{\theta} = \mathbf{0}$. The zero matrix is a critical point of (21) if and only if $\lambda_2 \geq \bar{\lambda}_2$.*

Similar to GL, we can also pursue a two-stage BGL approach. First, use BGL to select K OPF features and determine a selection matrix $\mathbf{S}_{\text{BGL}}$ from (7) based on the sparsity pattern of $\mathbf{W}_{\text{BGL}}$. Second, design the reconstruction matrix $\mathbf{C}$ as:

$$\mathbf{C}_{\text{BGL2}} = \arg\min_{\mathbf{C}} \ \frac{1}{2T}\sum_{t=1}^{T}\|\mathbf{x}_t - \hat{\mathbf{x}}_t(\mathbf{C}\mathbf{S}_{\text{BGL}}^{\top})\|^2.$$

The two-stage BGL (BGL2) returns the distillation matrix:

$$\mathbf{W}_{\text{BGL2}} = \mathbf{C}_{\text{BGL2}}\mathbf{S}_{\text{BGL}}^{\top}.$$

V. NUMERICAL TESTS

The proposed OPF data distillation approaches were numerically evaluated using a single-phase rendition of the IEEE 37-bus benchmark feeder and the original, multiphase version of the IEEE 123-bus feeder. Optimal DER reactive power injections $\mathbf{x}_t$ and $\hat{\mathbf{x}}_t$ were obtained by solving (5) using YALMIP and Sedumi in MATLAB. For the multiphase grid, the objective cost in (5) was modified to $\|q\|^2$. All tests were run on a MacBook laptop computer with an M3 Pro chip and 36 GB of RAM. Type-1 methods took less than a few minutes to run. Type-2 methods took from seconds to a few minutes to run for larger K and a few hours for smaller K.

A. Single-phase 37-bus system

Data generation. Active power demand and solar generation data were collected from the Pecan Street dataset [26]. We collected minute-based kW load and solar generation data from 25 households between 8 AM and 5 PM during the entire year of 2019. We also collected synthetically generated electric vehicle (EV) profiles over a year from [27]. Among the 37 feeder buses, only 25 correspond to non-zero injections and were simulated to host loads. These 25 buses correspond to medium-voltage nodes, serving multiple residential customers each. To simulate aggregate load profiles, for each one of the medium-voltage nodes, we randomly sampled 5 out of the 25 Pecan Street households and added their profiles. The obtained load profiles were scaled to match twice the kW benchmark load. Because the original dataset does not include reactive power demands, lagging power factors were simulated by randomly drawing them uniformly within $[0.85, 1.00]$ across time and nodes. We simulated 10 PVs installed on buses $\{2, 4, 7, 9, 11, 14, 17, 20, 22, 25\}$. Solar profiles from 5 randomly selected Pecan Street households were aggregated to generate the solar profile for each of the 10 medium-voltage nodes. Solar profiles were scaled to match twice the peak kW benchmark load. We randomly sampled 25 EV profiles from [27] and assigned each to the aggregate load profiles.

Fidelity in OPF data under single-phase system. The first test compares the developed methods in terms of fidelity
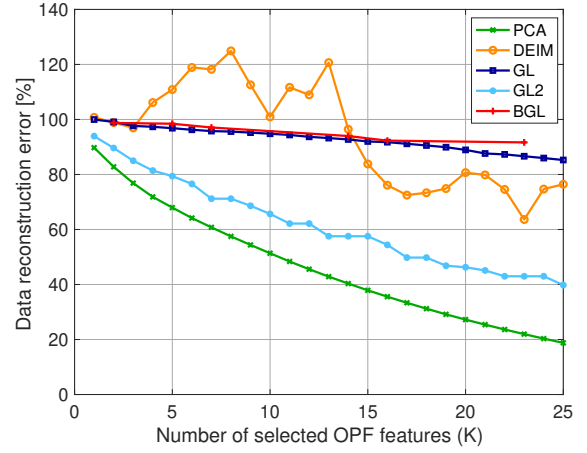

Fig. 2. Normalized data reconstruction error $f_1(\mathbf{W})/\|\boldsymbol{\Theta}\|_F^2 = \|\boldsymbol{\Theta} - \hat{\boldsymbol{\Theta}}\|_F^2/\|\boldsymbol{\Theta}\|_F^2$ obtained by OPF data distillation methods for increasing numbers of OPF features for the single-phase 37-bus system.
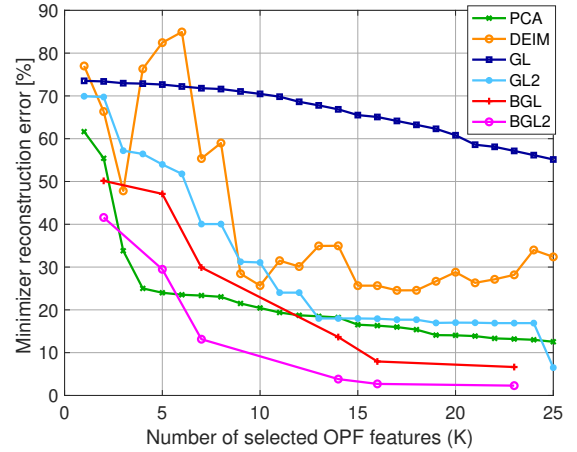

Fig. 3. Normalized minimizer error $f_2(\mathbf{W})/\|\mathbf{X}\|_F^2 = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2/\|\mathbf{X}\|_F^2$ obtained by OPF data distillation methods for increasing numbers of OPF features for the single-phase 37-bus system.

in reconstructing OPF data. The test involved four Type-1 approaches (PCA, DEIM, GL, and GL2) and one Type-2 approach (BGL). We used the normalized squared error $\frac{f_1(\mathbf{W})}{\|\boldsymbol{\Theta}\|_F^2}$ as a data fidelity metric. This metric was evaluated for an increasing number K of selected OPF features. Data reconstruction errors are expected to decrease for increasing K. Parameter K is entered explicitly for PCA and DEIM. For GL-type approaches, we can obtain K non-zero groups in $\mathbf{W}$ by varying parameters λ_1 and λ_2. Figure 2 illustrates the data reconstruction error for varying K. PCA serves only as a lower bound as it does not offer data selection. Although DEIM does select data, its performance is unpredictable and exhibits no clear trend. The errors attained by plain GL decrease only marginally as K increases. On the contrary, the GL2 reduces errors more dramatically and follows the trend set by PCA while featuring data selection. BGL behaves similarly to GL and bears no meaningful improvement. Overall, the GL2 approach is the best method to approximate $\boldsymbol{\Theta}$ when considering sensor selection.

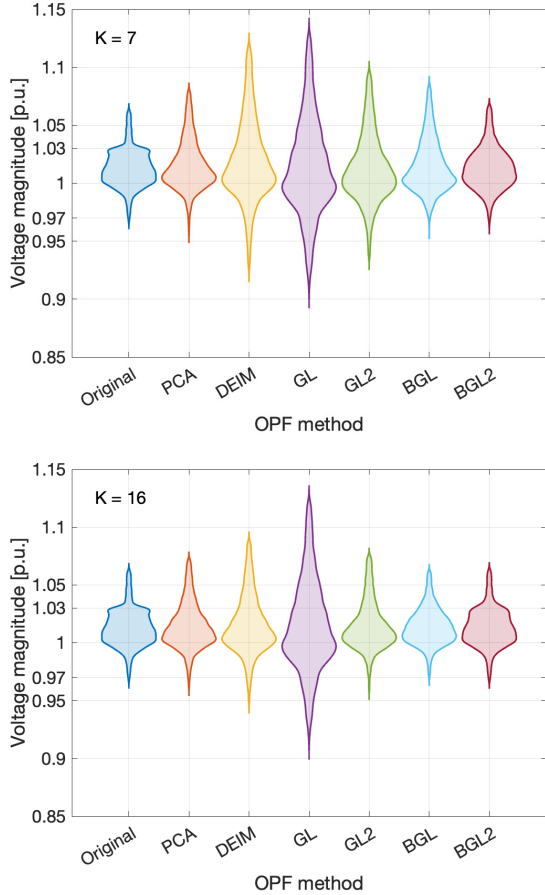*Fidelity in OPF solutions.* The second test compares four

Fig. 4. Violin plots of voltage magnitudes for the single-phase network experienced under all OPF methods for $K = 7$ (top panel) and $K = 16$ (bottom panel) OPF features. Violin plots are computed across all buses and time instances. The violin plots obtained using both BGL and two-stage BGL for $K = 16$ are almost identical to that of the original OPF method.

Type-1 approaches to two Type-2 approaches (BGL and BGL2) in terms of fidelity in finding OPF minimizers $\mathbf{X}$. Figure 3 illustrates the normalized squared error $\frac{f_2(\mathbf{W})}{\|\mathbf{X}\|_F^2}$ for increasing K. PCA attains the smallest error across all Type-1 designs except for a few cases where GL2 attains smaller errors. DEIM does not show a clear trend and GL shows slight improvement for increasing K. Although the errors attained by GL2, BGL and BGL2 decrease when K increases, BGL2 achieves smaller errors. The tests corroborate that BGL2 is the preferred method for approximating OPF solutions $\mathbf{X}$.

Feasibility of OPF solutions. In addition to minimizing $f_2(\mathbf{W})$, it is critical that $\hat{\mathbf{x}}_t = \mathbf{x}(\hat{\boldsymbol{\theta}}_t)$ does not violate voltage constraints when applied to the grid under actual loading $\boldsymbol{\theta}_t$. We computed the nodal voltages obtained by all OPF data distillation methods according to (2). Regardless of how DER setpoints were obtained, their voltage effect on the grid was computed using the actual loading conditions. Figure 4 depicts the violin plots of all bus voltages across all instances for each method and two values of $K = 7$ and $K = 16$. Note that even the OPF fed with the original data experiences voltages outside the $\pm 3\%$ range because voltage constraints are treated as soft. The tests show that voltage magnitudes are more concentrated
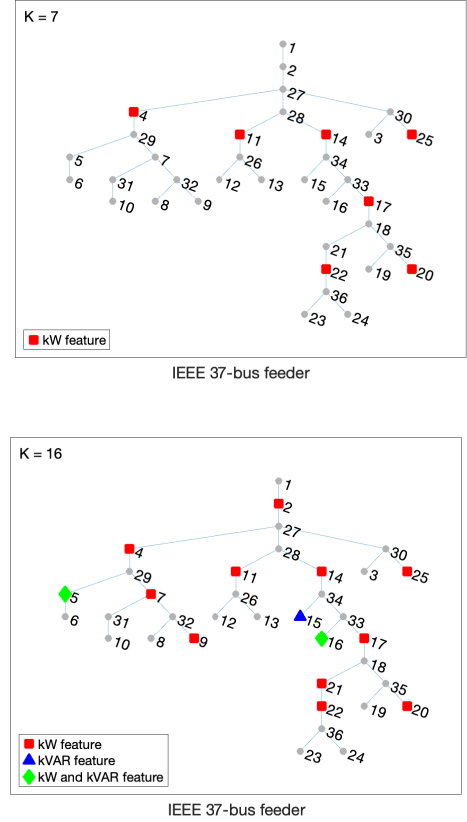

Fig. 5. OPF features selected by BGL for $K = 7$ and $K = 16$.

under both BGL and BGL2 for both values of K. Interestingly, the voltage distributions attained by BGL and BGL2 using only $K = 16$ OPF features are almost identical to those achieved by the original OPF fed by all $P = 50$ OPF features.

Selected OPF features. We next observed which features were selected for different values of K. Figure 5 shows that when $K = 7$, only active power demands at seven different nodes were selected. However, when $K = 16$, both active and reactive power demands were selected for some nodes, and for other nodes either active or reactive demands were selected.

B. Multiphase 123-bus system

Data generation. We also ran tests on the IEEE 123-bus multiphase distribution feeder [28]. Out of its 262 nodes (combinations of phases and buses excluding the substation), only 95 host non-zero injections. We obtained active power demand and solar generation data at one-minute intervals from the Smart* project [29]. The dataset contained active loads from 444 homes and solar generation from 45 PVs between 5:00 AM and 9:30 PM. We aggregated load demands from multiple homes to simulate 95 active load profiles for the 95 non-zero injection nodes of the tested feeder. Load profiles were scaled to match the kW benchmark load. Reactive load profiles were simulated by drawing power factors uniformly at random within $[0.85, 0.99]$ across time and nodes. We also added 95 EV profiles by randomly sampling from [27]. We simulated 10 PVs by aggregating the solar generation from the

TABLE I
FIDELITY IN OPF SOLUTION FOR THE IEEE 123-BUS SYSTEM

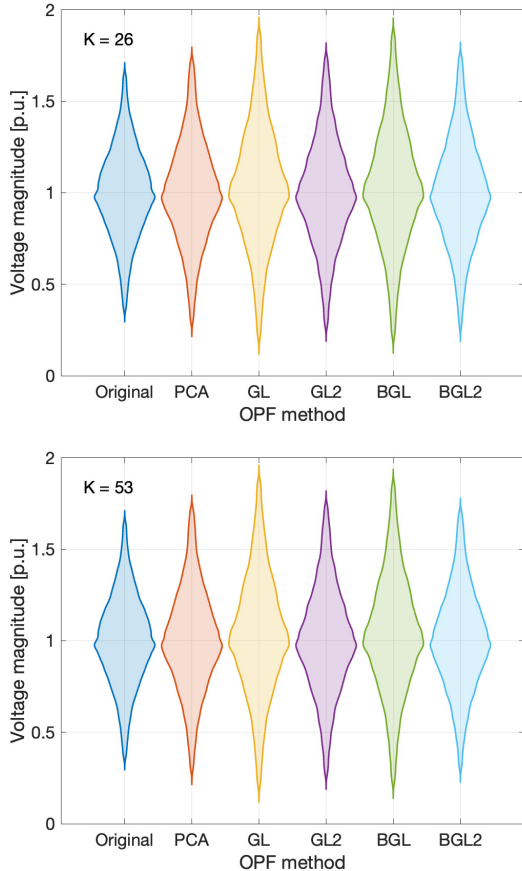| # selected features K	Normalized error in OPF solutions [%]				
	PCA	**GL**	**GL2**	**BGL**	**BGL2**
3	63.62	64.77	65.23	64.72	54.94
26	52.18	64.73	58.10	63.99	31.27
32	46.96	64.72	51.30	63.98	27.59
53	35.03	64.64	42.69	62.80	24.67

Fig. 6. Violin plots of voltages for the multiphase system for $K = 26$ and 53. BGL2 is almost identical to the original method for $K = 53$.

45 PVs from [29] and installed them on nodes 2a, 10b, 34a, 46c, 51c, 65a, 76b, 93a, 103b, and 118c.

Fidelity. The first test for the multiphase network compares GL/GL2 and BGL/BGL2 in terms of reconstructing the OPF minimizers $\mathbf{X}$. Table I shows the normalized squared errors for 4 different values of K. Similar to the single-phase network, PCA attains the smallest errors among all Type-1 designs in the multiphase network. Again, BGL2 is the best method for approximating OPF solutions. The reported errors verify the importance of the two-stage approach.

Feasibility. We simulated the approximate OPF solutions along with the true loading conditions to compute nodal voltages per (2). Figure 6 shows the violin plots across all nodes and scenarios for all methods. The tests show that the voltage magnitudes are more concentrated for BGL2 and that by using only $K = 53$ OPF features, we can achieve voltage distributions almost identical to those achieved by the original OPF fed by all $P = 190$ OPF features.

VI. CONCLUSIONS

We have proposed an OPF data distillation framework to improve the fidelity of either OPF features or OPF minimizers. Type-2 OPF data distillation gives rise to a non-convex program, solved by a proximal gradient algorithm. To alleviate the computational burden of finding the minimizers and their sensitivities for a batch of OPF instances, we have leveraged results from multiparametric programming. Extensive numerical tests using synthetic data on a single-phase and real-world data on a multiphase feeder demonstrate the advantages of Type-2 over Type-1 methods in approximating the minimizer and satisfying network constraints at the expense of increased complexity. Numerical tests have shown that optimal DER schedules can be approximated reasonably well upon collecting only $K = 16$ rather than $P = 50$ OPF features for the IEEE 37-bus single-phase feeder. For the multiphase system, DER schedules can be approximated within 25% error with $K = 53$ out of $P = 190$ features while nearly matching the voltage distribution of the exact OPF solutions. The proposed framework sets a solid foundation for exciting research directions: *i)* Extending data distillation to AC-OPF formulations or other OPF variants; *ii* Pursuing stochastic variants of APGD that process one or a few OPF instances per iteration to reduce the computational cost; *iii)* Leveraging OPF data distillation tools towards optimal meter placement to support OPF operations; *iv)* Upgrading the linear to a nonlinear reconstruction module (deep neural network); *v)* Pursuing unsupervised counterparts of OPF data distillation to waive the need of generating a labeled OPF dataset; and *vi)* Identifying the most influential OPF features so their communication to the DSO is protected against cyber-attacks.

APPENDIX

Proof of Lemma 1: The subdifferential of $g(\mathbf{W})$ with respect to the p-th column of $\mathbf{W}$ is

$$\partial_{\mathbf{w}_p} g(\mathbf{W}) = \begin{cases} \frac{\mathbf{w}_p}{\|\mathbf{w}_p\|} & , \text{ if } \mathbf{w}_p \neq \mathbf{0} \\ \{\mathbf{g}_p \in \mathbb{R}^P : \|\mathbf{g}_p\| \leq 1\} & , \text{ if } \mathbf{w}_p = \mathbf{0} \end{cases} \quad (22)$$

for all p. Function $g(\mathbf{W})$ is differentiable with respect to $\mathbf{w}_p$ for all $\mathbf{w}_p \neq \mathbf{0}$. Its subdifferential at $\mathbf{w}_p = \mathbf{0}$ can be defined as the set of all vectors having ℓ_2-norm not larger than one.

Optimality conditions for subdifferentiable functions predicate that a matrix $\mathbf{W}_{\text{GL}}$ is a minimizer of (14) if and only if the zero matrix belongs to the subdifferential of $f_1 + \lambda_1 g$ evaluated at $\mathbf{W}_{\text{GL}}$. Therefore, the minimizer of (14) satisfies:

$$\mathbf{0} \in \nabla_{\mathbf{W}} f_1(\mathbf{W}_{\text{GL}}) + \lambda_1 \partial_{\mathbf{W}} g(\mathbf{W}_{\text{GL}}).$$

Substituting $\nabla_{\mathbf{W}} f_1(\mathbf{W})$ from (16) above yields

$$\mathbf{0} \in (\mathbf{W}_{\text{GL}} - \mathbf{I})\mathbf{C}_\theta + \lambda_1 \partial_{\mathbf{W}} g(\mathbf{W}_{\text{GL}}). \quad (23)$$

If $\mathbf{W}_{\text{GL}} = \mathbf{0}$, the optimality condition of (23) provides:

$$\lambda_1 \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 & \cdots & \mathbf{g}_P \end{bmatrix} = \mathbf{C}_\theta \quad \text{or} \quad \lambda_1 \mathbf{g}_p = \mathbf{c}_p \; \forall p$$

where $\mathbf{g}_p$ is defined by the lower branch of (22) as it evaluates $\partial_{\mathbf{w}_p} g$ at $\mathbf{W}_{\text{GL}} = \mathbf{0}$. Since $\|\mathbf{g}_p\| \leq 1$ for all p by the definition of the subgradient at zero, it follows that $\|\mathbf{c}_p\| = \lambda_1 \|\mathbf{g}_p\| \leq \lambda_1$

for all p. This establishes that $\lambda_1 \geq \bar{\lambda}_1$. Conversely, if $\lambda_1 \geq \bar{\lambda}_1$, matrix $\mathbf{W}_{\text{GL}} = \mathbf{0}$ can be shown to satisfy (23) by selecting $\mathbf{g}_p = \frac{1}{\lambda_1}\mathbf{c}_p$ so that $\|\mathbf{g}_p\|_2 \leq 1$ for all p. ∎

Proof of Lemma 2: We first compute the gradient of f_2 defined in (20) with respect to vector $\mathbf{w} = \text{vec}(\mathbf{W})$. This is a P^2-long vector wherein all columns of $\mathbf{W}$ are stacked vertically. We will use the property that for any three matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ of conformable dimensions, it holds that:

$$\text{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\,\text{vec}(\mathbf{B}) \tag{24}$$

where $\otimes$ is the Kronecker product.

The gradient of the t-th summand of f_2 is

$$\nabla_{\mathbf{w}}\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2 = 2(\nabla_{\mathbf{w}}\hat{\mathbf{x}}_t)^\top(\hat{\mathbf{x}}_t - \mathbf{x}_t).$$

The Jacobian $\nabla_{\mathbf{w}}\hat{\mathbf{x}}_t$ can be found using the chain rule as

$$\nabla_{\mathbf{w}}\hat{\mathbf{x}}_t = \nabla_{\hat{\boldsymbol{\theta}}_t}\hat{\mathbf{x}}_t \cdot \nabla_{\mathbf{w}}\hat{\boldsymbol{\theta}}_t.$$

Thanks to (24), we can express data $\hat{\boldsymbol{\theta}}_t = \mathbf{W}\boldsymbol{\theta}_t$ as

$$\hat{\boldsymbol{\theta}}_t = \text{vec}(\hat{\boldsymbol{\theta}}_t) = \text{vec}(\mathbf{W}\boldsymbol{\theta}_t) = (\boldsymbol{\theta}_t^\top \otimes \mathbf{I}_P)\mathbf{w}$$

from which it follows that $\nabla_{\mathbf{w}}\hat{\boldsymbol{\theta}}_t = \nabla_{\mathbf{w}}\hat{\boldsymbol{\theta}}_t = \boldsymbol{\theta}_t^\top \otimes \mathbf{I}_P$. Substituting the latter into the gradient of $\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2$ yields:

$$\nabla_{\mathbf{w}}\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2 = 2(\boldsymbol{\theta}_t \otimes \mathbf{I}_P)(\nabla_{\hat{\boldsymbol{\theta}}_t}\hat{\mathbf{x}}_t)^\top(\hat{\mathbf{x}}_t - \mathbf{x}_t).$$

Therefore, the gradient of $\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2$ with respect to $\mathbf{W}$ is

$$\nabla_{\mathbf{W}}\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2 = 2(\nabla_{\hat{\boldsymbol{\theta}}_t}\hat{\mathbf{x}}_t)^\top(\hat{\mathbf{x}}_t - \mathbf{x}_t)\boldsymbol{\theta}_t^\top.$$

This can be verified using (24) for $\mathbf{A} = 2(\nabla_{\hat{\boldsymbol{\theta}}_t}\hat{\mathbf{x}}_t)^\top(\hat{\mathbf{x}}_t - \mathbf{x}_t)$, $\mathbf{B} = 1$, and $\mathbf{C} = \boldsymbol{\theta}^\top$, which shows that

$$\text{vec}(\nabla_{\mathbf{W}}\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2) = \nabla_{\mathbf{w}}\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2.$$

Summing up the gradients of all $\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2$ across t and dividing by $2T$ completes the proof of this lemma. ∎

Proof of Lemma 3: If $\mathbf{W}_{\text{BGL}}$ is a critical point, then

$$\mathbf{0} \in \nabla_{\mathbf{W}}f_2(\mathbf{0}_{P\times P}) + \lambda_2\partial g(\mathbf{0}_{P\times P}). \tag{25}$$

The gradient of f_2 can be found from Lemma 2. For $\mathbf{W}_{\text{BGL}} = \mathbf{0}$, it holds $\hat{\boldsymbol{\theta}}_t = \mathbf{W}_{\text{BGL}}\boldsymbol{\theta}_t = \mathbf{0}$ for all t. We therefore have that $\hat{\mathbf{x}}_t = \mathbf{x}(\mathbf{0})$ and $\nabla_{\hat{\boldsymbol{\theta}}_t}\hat{\mathbf{x}}_t = \nabla_{\boldsymbol{\theta}}\mathbf{x}(\mathbf{0})$ for all t. The rest of the proof follows the proof of Lemma 1 and is omitted. ∎

REFERENCES

[1] M. K. Singh, V. Kekatos, and G. B. Giannakis, "Learning to solve the AC-OPF using sensitivity-informed deep neural networks," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 2833–2846, Jul. 2022.

[2] M. Jalali, M. K. Singh, V. Kekatos, G. B. Giannakis, and C. C. Liu, "Fast inverter control by learning the OPF mapping using sensitivity-informed Gaussian processes," *IEEE Trans. Smart Grid*, vol. 14, no. 3, pp. 2432–2445, May 2022.

[3] T. Mühlpfordt, V. Hagenmeyer, and T. Faulwasser, "Uncertainty quantification for optimal power flow problems," *Proceedings in Applied Mathematics and Mechanics*, vol. 19, no. 1, 2019.

[4] A. J. Conejo, E. Castillo, R. Minguez, and R. Garcia-Bertrand, *Decomposition techniques in mathematical programming*. Springer, 2006.

[5] L. Roald, D. Pozo, A. Papavasiliou, D. Molzahn, J. Kazempour, and A. Conejo, "Power systems optimization under uncertainty: A review of methods and applications," *Electric Power Systems Research*, vol. 214, 2023.

[6] R. Mieth, J. M. Morales, and H. V. Poor, "Data valuation from data-driven optimization," *IEEE Trans. Control of Network Systems*, pp. 1–12, 2024.

[7] V. Dvorkin, F. Fioretto, P. Van Hentenryck, P. Pinson, and J. Kazempour, "Differentially private optimal power flow for distribution grids," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2186–2196, May 2021.

[8] S. Chevalier and M. R. Almassalkhi, "Towards optimal Kron-based reduction of networks (Opti-KRON) for the electric power grid," in *Proc. IEEE Conf. on Decision and Control*, 2022, pp. 5713–5718.

[9] I. P. Nikolakakos, H. H. Zeineldin, M. S. El-Moursi, and J. L. Kirtley, "Reduced-order model for inter-inverter oscillations in islanded droop-controlled microgrids," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4953–4963, 2018.

[10] W. Jang, S. Mohapatra, T. J. Overbye, and H. Zhu, "Line limit preserving power system equivalent," in *2013 IEEE Power and Energy Conference at Illinois (PECI)*, 2013, pp. 206–212.

[11] S. Park, W. Chen, T. W. Mak, and P. Van Hentenryck, "Compact optimization learning for AC optimal power flow," *IEEE Trans. Power Syst.*, vol. 39, no. 2, pp. 4350–4359, Mar. 2024.

[12] S. Gupta, V. Kekatos, and M. Jin, "Deep learning for reactive power control of smart inverters under communication constraints," in *Proc. IEEE Intl. Conf. on Smart Grid Commun.*, Tempe, AZ, Nov. 2020, pp. 1–6.

[13] J. Kotary, V. Vito, J. Cristopher, P. Van Hentenryck, and F. Fioretto, "Learning joint models of prediction and optimization," 09 2024.

[14] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2007.

[15] S. Bhela, V. Kekatos, and S. Veeramachaneni, "Power distribution system observability with smart meter data," in *Proc. IEEE Global Conf. on Signal and Inf. Process.*, Montreal, Canada, Nov. 2017.

[16] P. Buason, S. Misra, and D. K. Molzahn, "A data-driven sensor placement approach for detecting voltage violations in distribution systems," 2023.

[17] F. Dorfler, M. R. Jovanovic, M. Chertkov, and F. Bullo, "Sparsity-promoting optimal wide-area control of power networks," *IEEE Trans. Power Syst.*, vol. 29, no. 5, pp. 2281–2291, Sep. 2014.

[18] V. Kekatos, R. Annin, M. K. Singh, and J. Qin, "Scheduling distributed energy resources under limited observability of distribution grids," in *Bulk Power System Dynamics and Control Symposium (IREP)*, Sorrento, Italy, Jun. 2025, (submited December 2024). [Online]. Available: https://engineering.purdue.edu/~kekatos/papers/IREP2025.pdf

[19] S. Taheri, M. Jalali, V. Kekatos, and L. Tong, "Fast probabilistic hosting capacity analysis for active distribution systems," *IEEE Trans. Smart Grid*, no. 3, pp. 2000–2012, May 2021.

[20] M. K. Singh, S. Taheri, V. Kekatos, K. P. Schneider, and C.-C. Liu, "Joint grid topology reconfiguration and design of Watt-VAR curves for DERs," in *Proc. IEEE Power & Energy Society General Meeting*, Denver, CO, Jul. 2022.

[21] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, "An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations," *Comptes Rendus Mathematique*, vol. 339, no. 9, pp. 667–672, 2004, publisher: Elsevier.

[22] S. Mosci, L. Rosasco, M. Santoro, A. Verri, and S. Villa, "Solving structured sparsity regularization with proximal methods," in *Machine Learning and Knowledge Discovery in Databases*. Berlin, Heidelberg: Springer, 2010, pp. 418–433.

[23] H. Li and Z. Lin, "Accelerated proximal gradient methods for nonconvex programming," in *Intl. Conf. on Neural Information Processing Systems*, vol. 1. Montreal, Canada: MIT Press, 2015, pp. 379—-387.

[24] S. Dempe and S. Franke, "Solution of bilevel optimization problems using the KKT approach," *Optimization*, vol. 68, no. 8, pp. 1471–1489, 2019.

[25] M. K. Singh, S. Gupta, V. Kekatos, G. Cavraro, and A. Bernstein, "Learning to optimize power distribution grids using sensitivity-informed deep neural networks," in *Proc. IEEE Intl. Conf. on Smart Grid Commun.*, Tempe, AZ, Nov. 2020, pp. 1–6.

[26] (2018) Pecan Street Inc. Dataport. [Online]. Available: https://dataport.cloud/

[27] M. Muratori, "Impact of uncoordinated plug-in electric vehicle charging on residential power demand - supplementary data," Golden, CO, 2017.

[28] M. Bazrafshan and N. Gatsis, "Comprehensive modeling of three-phase distribution systems via the bus admittance matrix," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 2015–2029, 2018.

[29] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, "Smart*: An open data set and tools for enabling research in sustainable homes," in *Proc. Workshop on Data Mining Applications in Sustainability*, Beijing, China, Aug. 2012.