Constrained Diffusion Models for Synthesizing Representative Power Flow Datasets

Milad Hoseinpour, *Student Member, IEEE*, Vladimir Dvorkin, *Member, IEEE*

***Abstract*—High-quality power flow datasets are essential for training machine learning models in power systems. However, security and privacy concerns restrict access to real-world data, making statistically accurate and physically consistent synthetic datasets a viable alternative. We develop a diffusion model for generating synthetic power flow datasets from real-world power grids that both replicate the statistical properties of the real-world data and ensure AC power flow feasibility. To enforce the constraints, we incorporate gradient guidance based on the power flow constraints to steer diffusion sampling toward feasible samples. For computational efficiency, we further leverage insights from the fast decoupled power flow method and propose a variable decoupling strategy for the training and sampling of the diffusion model. These solutions lead to a physics-informed diffusion model, generating power flow datasets that outperform those from the standard diffusion in terms of feasibility and statistical similarity, as shown in experiments across IEEE benchmark systems.**

***Index Terms*—Diffusion model, generative AI in power systems, physics-informed machine learning, power flow, synthetic data.**

I. Introduction

POWER flow datasets [1]–[3] are essential for training and benchmarking machine learning (ML) models for optimal power flow (OPF) [4] and state estimation [5]. However, the real-world power flow datasets are rarely available due to privacy, security, and legal barriers [6]–[10]. Recent advances in generative AI, capable of producing synthetic data with distributions similar to the original data [11]–[20], have partially lifted these barriers, yet statistical consistency alone cannot guarantee adherence to physical grid constraints [21]. Consequently, ML models trained on constraint-agnostic synthetic datasets are likely to perform substantially worse than those trained on original data. This paper introduces a data generation framework to synthesize statistically consistent and physically meaningful power flow datasets. To achieve this, we develop a constrained diffusion model to learn the underlying distribution of power flow data and generate synthetic samples that are both statistically representative and feasible with respect to the AC power flow constraints. This constrained diffusion model can be trained internally by system operators to publicly release high-quality synthetic power flow data to support a wide range of downstream ML applications.

A. Related Work

The literature on generating synthetic datasets for power systems broadly falls into two categories: generic random sampling and historical data-driven approaches.

The former focuses on power flow data generation through iterative uniform sampling of loads followed by solving the OPF problem [22], [23]. In [24], authors use a truncated Gaussian distribution as another variation of sampling, which also accounts for correlations between power injections at different locations. However, the datasets based on generic sampling only represent a small portion of the feasibility region. To solve this, [6] uniformly samples loads from a convex set, containing the feasible region, and iteratively refines this set using infeasibility certificates. In [25], a bilevel optimization is proposed to sample operating conditions close to the boundaries of the feasible region, which is more informative that a random sampling. A basic requirement for ML-based OPF solvers is robustness to grid topology variations, e.g., network topology switching [26]. To meet this requirement, authors in [27] incorporate topological perturbations in addition to load perturbations in their synthetic data generation framework.

Although straightforward, random sampling comes with certain limitations. The resulting datasets do not represent the true underlying distribution of real-world operating conditions. That is, the synthetic data points may fail to capture correlations, patterns, or variability present in historical data. ML-based OPF solvers trained on such data may generalize poorly, leading to inaccurate predictions and erroneous uncertainty quantification [28], [29]. Moreover, the required number of random samples to cover the whole feasible region grows exponentially in the size of the grid [30], [31].

The historical data-driven approaches, instead, learn the underlying data distribution from real operational records. This approach has been enabled by advances in generative models, such as variational autoencoders (VAEs), generative adversarial networks (GANs), and diffusion models. For instance, the VAE from [13] generates synthetic electric vehicles load profiles, and conditional VAE from [12] does the same for snapshots of multi-area electricity demand. Reference [14] presents conditional VAE for synthesizing load profiles of industrial and commercial customers, conditioned on time and typical power exchange with the grid. However, VAEs may struggle with complex and high-dimensional datasets and result in low quality samples [15]. Moreover, there is no principled approach for VAEs to control the generated outputs, making it difficult to enforce domain-specific constraints. GANs have also been used to synthesize load patterns in power systems [16]–[20]. For instance, [19] proposes a GAN model to generate synthetic appliance-level load patterns and usage habits. In [20], authors propose a GAN-based framework for renewable energy scenario generation that effectively captures

The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA. E-mail: {miladh, dvorkin}@umich.edu.

both temporal and spatial patterns across a large number of correlated resources. Nonetheless, GANs also suffer from issues such as training instability, mode collapse, and the lack of principled means for controllability [32].

Addressing the limitations of GANs and VAEs, diffusion models have emerged as the leading choice for generative models [33]. A physics-informed diffusion model is proposed in [11] for generating synthetic net load data, where the solar PV system performance model is embedded into the diffusion model. In [34], authors propose a conditional latent diffusion model for short-term wind power scenario generation, which uses weather conditions as inputs. Authors in [35] developed a framework based on diffusion models to generate electric vehicle charging demand time-series data, which is also capable of capturing temporal correlation between charging stations.

While this line of work advocates for diffusion models to generate power systems data, it primarily focuses on statistical consistency. To the best of our knowledge, no prior work has explored the integration of domain constraints such as the AC power flow constraints directly into the diffusion process.

B. Summary of Contributions

The main contribution of this paper is a generative AI framework that leverages power systems operational data to synthesize credible power flow datasets for ML applications. Specific contributions are summarized as follows:

1) We develop a diffusion model capable of generating high-quality power flow datasets that inherit the statistical properties of the actual power flow records. Unlike random sampling in [6], [22]–[27], the model does not require any distributional assumptions; rather, as a generative model, it learns the distribution of power flow data directly from historical records. Yet, unlike the existing generative models in [11]–[20], it controls the output to ensure compliance of synthetic samples with the grid physics. Our model focuses on synthesizing power injections and voltage variables—the data to support OPF, state estimation, and other applications of ML to power systems optimization and control.
2) We introduce a guidance term within the sampling phase to ensure that the synthesized data points are feasible with respect to the AC power flow constraints. The guidance term corresponds to a single iteration of Riemannian gradient descent on the clean data manifold—the space of all physically valid power flow states. We formally prove that this approach improves physical consistency without pushing samples off the learned distribution. As a result, the generated power flow data points are both feasible and statistically representative.
3) We leverage power systems domain knowledge for implementing the diffusion model. Inspired by the classical fast decoupled power flow method, we decouple the full variable vector and use two smaller denoiser neural networks to improve scalability. We then propose a custom normalization of the AC power flow equations for stabilizing the sampling of power flow variables.

The remainder is organized as follows. The problem statement is presented in Sec. II, followed by Sec. III with preliminaries on diffusion models and power flows. Section IV introduces the proposed manifold-constrained guidance for enforcing the power flow constraints in diffusion sampling. Then, Sec. V provides implementation insights tailored to power systems: variable decoupling and normalization for scale-consistent gradient guidance. Section VI provides numerical results on the standard IEEE test cases. Section VII concludes.

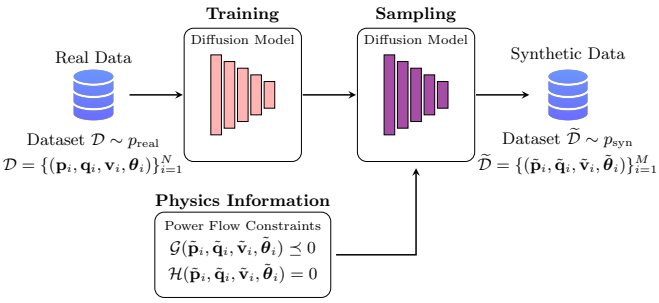


Fig. 1. A high-level view of the diffusion model for synthesizing power flow datasets: The training phase (left) uses the actual power flow data $\mathcal{D}$ to learn the real data distribution p_{real} using a neural network; the sampling phase (right) uses the trained neural network to generate synthetic power flow samples $\widetilde{\mathcal{D}}$; the integration of the power flow constraints (bottom) within the sampling phase ensures that generated samples are physically meaningful.

II. Problem Statement

Consider a power grid characterized by vectors of active power injections $\mathbf{p}$, reactive power injections $\mathbf{q}$, voltage magnitudes $\mathbf{v}$, and phase angles $\boldsymbol{\theta}$. Given a historical dataset $\mathcal{D} = \{(\mathbf{p}_i, \mathbf{q}_i, \mathbf{v}_i, \boldsymbol{\theta}_i)\}_{i=1}^{N}$ with N power flow records, our goal is to generate a synthetic dataset $\widetilde{\mathcal{D}} = \{(\tilde{\mathbf{p}}_i, \tilde{\mathbf{q}}_i, \tilde{\mathbf{v}}_i, \tilde{\boldsymbol{\theta}}_i)\}_{i=1}^{M}$ with M records, which are statistically representative of the given dataset $\mathcal{D}$ and are feasible with respect to the power flow constraints, i.e., data should satisfy the following conditions:

$$\min \textbf{dist}\left(p_{\text{syn}} \,||\, p_{\text{real}}\right), \tag{1a}$$

$$\mathcal{G}(\tilde{\mathbf{p}}_i, \tilde{\mathbf{q}}_i, \tilde{\mathbf{v}}_i, \tilde{\boldsymbol{\theta}}_i) \leq 0, \quad \forall i = 1, \ldots, M, \tag{1b}$$

$$\mathcal{H}(\tilde{\mathbf{p}}_i, \tilde{\mathbf{q}}_i, \tilde{\mathbf{v}}_i, \tilde{\boldsymbol{\theta}}_i) = 0, \quad \forall i = 1, \ldots, M. \tag{1c}$$

where the first condition (1a) requires minimizing the statistical distance between the real p_{real} and synthetic p_{syn} probability distributions, measured by $\textbf{dist}(\cdot||\cdot)$ (e.g., Wasserstein distance). The second condition (1b) requires the synthetic records to satisfy the inequality constraints $\mathcal{G}$, including the injection and voltage limits. The last condition (1c) requires satisfaction of the power flow equality constraints $\mathcal{H}$.

Figure 1 gives a high-level view of the problem setup. We first train a diffusion model based on $\mathcal{D}$ to learn the underlying probability distribution p_{real}. Then, we sample from the learned distribution p_{syn} to build a synthetic dataset $\widetilde{\mathcal{D}}$. To ensure the feasibility of the synthetic samples, we guide the sampling process using the power flow constraints.

III. Preliminaries

This section presents preliminaries on diffusion models and power flow modeling; readers familiar with both topics are invited to proceed to the next section.

A. Diffusion Models

Diffusion models are generative models that synthesize new data through a two-stage process: forward and reverse. Consider $\mathbf{x}_0 = (\mathbf{p}, \mathbf{q}, \mathbf{v}, \boldsymbol{\theta})$ as a real power flow data point from the underlying distribution of the real data $p_{\text{real}} = q_0$. The forward process is a Markov chain that incrementally adds Gaussian noise to a real data point $\mathbf{x}_0 \sim q_0$ and transforms it into pure Gaussian noise through a fixed sequence of steps. For each time step $\{t\}_{t=1}^{T}$, the diffusion transition kernel is

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \mathbf{I}), \tag{2}$$

where β_t is a small positive constant controlling the amount of noise added at each step [36]. From (2), we directly obtain $\mathbf{x}_t$ from $\mathbf{x}_0$ using

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{3}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.

The reverse process aims to recover the underlying data distribution q_0 from the tractable noise distribution q_T. It is modeled as a parameterized Markov chain:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)), \tag{4}$$

with the mean $\boldsymbol{\mu}_\theta(\cdot, t)$ and covariance $\boldsymbol{\Sigma}_\theta(\cdot, t)$ functions learned using neural networks parametrized by θ [36].

To train the neural network, we select the loss function as a mean-squared error between the actual noise $\boldsymbol{\epsilon}$ added during the forward process and the noise $\boldsymbol{\epsilon}_\theta(\cdot, t)$ predicted by the neural network:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, t) \right\|^2, \tag{5}$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ and $\bar{\alpha}_t$ as defined before. Algorithm 1 summarizes the implementation of the training process [36].

Once the neural network is trained, new data points are generated using the predicted clean sample $\hat{\mathbf{x}}_0$ at each step t via Tweedie's formula:

$$\hat{\mathbf{x}}_0(\mathbf{x}_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right), \tag{6}$$

and then

$$\mathbf{x}_{t-1} = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{x}}_0 + \sigma_t \mathbf{z}, \tag{7}$$

where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, $\sigma_t = \beta_t (1-\bar{\alpha}_{t-1}) / (1-\bar{\alpha}_t)$, and t ranges from T (starting with the pure Gaussian noise) to 1 (generated sample). Algorithm 2 summarizes the implementation of the sampling process [36], which returns the synthetic sample $\tilde{\mathbf{x}}_0$ statistically consistent with the original sample $\mathbf{x}_0$.

B. Power Flow Constraints

Figure 2 illustrates the grid topology and notation used throughout this section. Let $\mathcal{B} = \{1, \cdots, B\}$ denote the set of buses and $\mathcal{L} = \{1, \cdots, L\}$ denote the set of transmission lines in a power grid. Moreover, let elements of power injection vectors $\mathbf{p}$ and $\mathbf{q}$ be indexed as p_b and q_b, and let elements of voltage vectors $\mathbf{v}$ and $\boldsymbol{\theta}$ be indexed as v_b and θ_b, $\forall b \in \mathcal{B}$. In the interest of presentation, we omit shunt admittances in the formulation, though they are included in our numerical results.

Algorithm 1 : Training the diffusion model

Inputs: initialized neural network $\boldsymbol{\epsilon}_\theta$, noise schedule $\{\alpha_t\}_{t=1}^{T}$, dataset of $\mathbf{x}_0$'s sampled from q_0

Outputs: trained neural network $\boldsymbol{\epsilon}_\theta$

1: **repeat**
2: $\quad \mathbf{x}_0 \sim q_0(\mathbf{x}_0)$
3: $\quad t \sim \text{Uniform}(\{1, \ldots, T\})$
4: $\quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$
5: $\quad$ Take gradient descent step on

$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t \right) \right\|^2$$

6: **until** converged

Algorithm 2 : Sampling new data points

Inputs: trained neural network $\boldsymbol{\epsilon}_\theta$, noise schedule $\{\alpha_t\}_{t=1}^{T}$, noise scale σ_t

Outputs: new data point $\tilde{\mathbf{x}}_0$

1: $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \hat{\mathbf{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right)$
4: $\quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = 0$
5: $\quad \mathbf{x}_{t-1} \leftarrow \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{x}}_0 + \sigma_t \mathbf{z}$
6: **return** $\tilde{\mathbf{x}}_0$

1) Power Flow Equality Constraints: For each bus $b \in \mathcal{B}$, power flow equality constraints can be represented as

$$p_b - \sum_{l \in \mathcal{L}: i=b} f^p_{l,i\to j} - \sum_{l \in \mathcal{L}: j=b} f^p_{l,j\to i} = 0, \tag{8a}$$

$$q_b - \sum_{l \in \mathcal{L}: i=b} f^q_{l,i\to j} - \sum_{l \in \mathcal{L}: j=b} f^q_{l,j\to i} = 0, \tag{8b}$$

where constraints (8a) and (8b) enforce the active and reactive nodal power balance. The explicit expression for active power flows $f^p_{l,i\to j}$ and reactive power flows $f^q_{l,i\to j}$ on each transmission line $l \in \mathcal{L}$ from node i to node j are given by:

$$f^p_{l,i\to j} = v_i v_j \left[g_l \cos(\theta_i - \theta_j) + b_l \sin(\theta_i - \theta_j) \right], \tag{9a}$$

$$f^q_{l,i\to j} = v_i v_j \left[g_l \sin(\theta_i - \theta_j) - b_l \cos(\theta_i - \theta_j) \right], \tag{9b}$$

where $g_l = G_{ij}$ and $b_l = B_{ij}$ are the real and imaginary parts of the grid admittance matrix $Y = G + jB$. Note that due to line power losses, $f^p_{l,i\to j} \neq f^p_{l,j\to i}$ and $f^q_{l,i\to j} \neq f^q_{l,j\to i}$ [37].

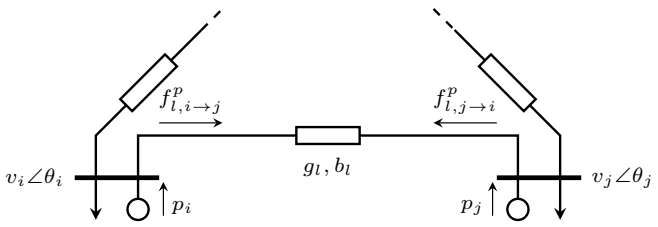
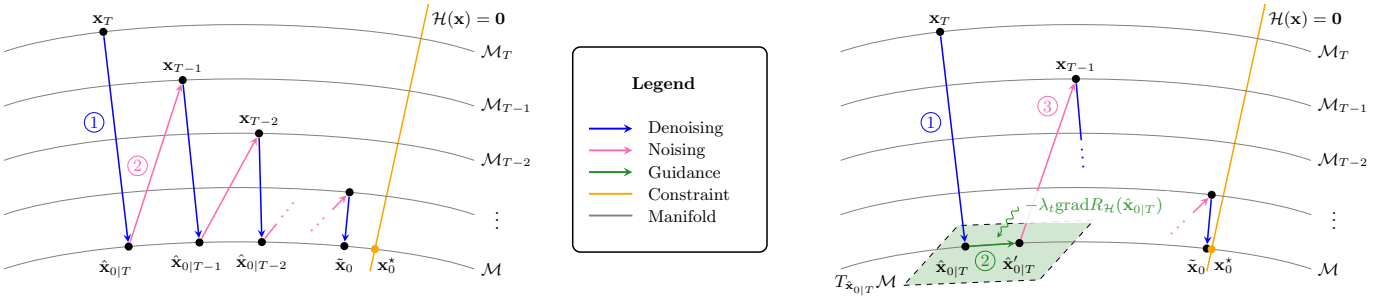


Fig. 2. Schematic diagram of the power grid.



(a) Sampling without guidance.

(b) Sampling with manifold-constrained gradient guidance.

Fig. 3. Schematic overview of the geometry of sampling (a) without guidance and (b) with manifold-constrained gradient guidance. In sampling without guidance (a), at each step t, we have a 2-stage reverse diffusion step: (1) we do a denoising step based on $\mathbf{x}_t$ and estimate the clean data $\hat{\mathbf{x}}_{0|t}$, and (2) by adding noise with respect to the corresponding noise schedule, we obtain $\mathbf{x}_{t-1}$. In sampling with guidance (b), we have a 3-stage reverse diffusion step: (1) we do a denoising step based on $\mathbf{x}_t$ and estimate the clean data $\hat{\mathbf{x}}_{0|t}$, (2) we add the guidance term based on the gradient of the constraints residual function $R_{\mathcal{H}}(\hat{\mathbf{x}}_{0|t})$ and obtain $\hat{\mathbf{x}}'_{0|t}$, and (3) by adding noise with respect to the corresponding noise schedule, we obtain $\mathbf{x}_{t-1}$.

2) Power Flow Inequality Constraints: Power flow inequality constraints can be represented as follows:

$$p_b^{min} \le p_b \le p_b^{max}, \quad \forall b \in \mathcal{B}, \tag{10a}$$
$$q_b^{min} \le q_b \le q_b^{max}, \quad \forall b \in \mathcal{B}, \tag{10b}$$
$$v_b^{min} \le v_b \le v_b^{max}, \quad \forall b \in \mathcal{B}, \tag{10c}$$
$$(f_{l,i\to j}^p)^2 + (f_{l,i\to j}^q)^2 \le (s_l^{max})^2, \quad \forall l \in \mathcal{L}. \tag{10d}$$

where constraints (10a) and (10b) impose limits on the active and reactive power injections, and constraints (10c) and (10d) do the same for nodal voltages and apparent power flows.

IV. Diffusion Guidance based on Power Flow Constraints

In theory, a diffusion model trained on feasible power flow data should satisfy constraints (8)–(10), as they are implicitly encoded in the training dataset. However, in practice, the training and sampling errors may lead to a different outcome [38], [39]. Although these errors enables the generative power of diffusion models to synthesize new yet statistically consistent samples, the generated power flow samples may not be feasible. In this section, we propose a guidance term for diffusion sampling that preserves the statistical properties of the learned distribution while steering the sampling trajectory toward physically meaningful power flow samples.

Figure 3a illustrates the geometry of standard sampling in a diffusion model. This geometry is characterized by a sequence of manifolds $\{\mathcal{M}_i\}_{i=0}^T$. At the bottom, there exists a clean data manifold $\mathcal{M} = \mathcal{M}_0$ surrounded by noisier manifolds according to the noise schedule, where the noisy data resides. Furthermore, let $\mathcal{H}(\mathbf{x}) = 0$ represent the power flow equations (8), where their intersection with the clean data manifold $\mathcal{M}$ is the ideal sample $\mathbf{x}_0^\star$. Accordingly, reverse diffusion steps can be characterized as mere transitions from manifold $\mathcal{M}_i$ to $\mathcal{M}_{i-1}$. At each step t, we have a 2-stage reverse diffusion step. First, we do a denoising step based on $\mathbf{x}_t$ and estimate the clean data $\hat{\mathbf{x}}_{0|t}$. Due to the geometric interpretation of diffusion models, a single denoising step at t from manifold $\mathcal{M}_t$ can be viewed as an orthogonal projection onto the clean data manifold $\mathcal{M}$ [40]. Then, by adding noise with respect to the corresponding noise schedule, we obtain $\mathbf{x}_{t-1}$. As shown in Fig. 3a, the standard sampling process is oblivious to the power flow constraints $\mathcal{H}(\mathbf{x}) = 0$. That is, no information about these constraints is incorporated into the sampling process.

To address this issue, we propose to incorporate a guidance term during the sampling process to encourage constraint satisfaction. The proposed guidance term, inspired by manifold-constrained gradients, incorporates the constraint information. Specifically, we define a data consistency loss function as a residual of power flow constraints $\mathcal{H}(\mathbf{x})$:

$$R_{\mathcal{H}}(\mathbf{x}) = \|\mathcal{H}(\mathbf{x})\|_2^2, \tag{11}$$

and aim to minimize this loss over the clean data manifold $\mathcal{M}$, which is implicitly learned by the diffusion model:

$$\min_{\mathbf{x}\in\mathcal{M}} R_{\mathcal{H}}(\mathbf{x}). \tag{12}$$

To guide sampling trajectories at each denoising step, we apply a Riemannian gradient descent with respect to (12):

$$\hat{\mathbf{x}}'_{0|t} = \hat{\mathbf{x}}_{0|t} - \lambda_t \text{ grad } R_{\mathcal{H}}(\hat{\mathbf{x}}_{0|t}), \tag{13}$$

where grad $R_{\mathcal{H}}(\hat{\mathbf{x}}_{0|t})$ denotes the Riemannian gradient of $R_{\mathcal{H}}$ at $\hat{\mathbf{x}}_{0|t}$, defined as the projection of the Euclidean gradient onto the tangent space of the manifold $\mathcal{M}$ at $\hat{\mathbf{x}}_{0|t}$, i.e., $T_{\hat{\mathbf{x}}_{0|t}}\mathcal{M}$ [41]:

$$\text{grad } R_{\mathcal{H}}(\hat{\mathbf{x}}_{0|t}) = \mathcal{P}_{T_{\hat{\mathbf{x}}_{0|t}}\mathcal{M}}\left(\nabla_{\mathbf{x}_t} R_{\mathcal{H}}(\hat{\mathbf{x}}_{0|t})\right). \tag{14}$$

However, since the clean data manifold $\mathcal{M}$ is not known explicitly, this makes it intractable to compute the projection operator $\mathcal{P}_{T_{\hat{\mathbf{x}}_{0|t}}\mathcal{M}}$ directly. Fortunately, under a local linearity assumption [40], it can be shown that the Euclidean gradient of $R_{\mathcal{H}}$ at $\hat{\mathbf{x}}_{0|t}$ is already aligned with the tangent space of $\mathcal{M}$ at $\hat{\mathbf{x}}_{0|t}$, making the projection step unnecessary. Leveraging the main theorem from [40] on manifold-constrained gradients, we formalize this insight in the following theorem.

Theorem 1. Let $\mathcal{M}$ denote the clean data manifold, and assume that in a local neighborhood of $\hat{\mathbf{x}}_{0|t}$, $\mathcal{M}$ is well approximated by an affine subspace. Then, the gradient of the residual function $R_{\mathcal{H}}(\hat{\mathbf{x}}_{0|t})$ is tangential to $\mathcal{M}$, i.e.,

$$\mathcal{P}_{T_{\hat{\mathbf{x}}_{0|t}}\mathcal{M}}\left(\nabla_{\mathbf{x}_t} R_{\mathcal{H}}(\hat{\mathbf{x}}_{0|t})\right) = \nabla_{\mathbf{x}_t} R_{\mathcal{H}}(\hat{\mathbf{x}}_{0|t}). \tag{15}$$

Proof. Let $\mathbf{x}_t$ denote a noisy sample at diffusion step t, and let $Q : \mathbb{R}^{4B} \to \mathbb{R}^{4B}$ denote the function that maps $\mathbf{x}_t$ to its corresponding clean estimate $\hat{\mathbf{x}}_{0|t}$:

$$\hat{\mathbf{x}}_{0|t} = Q(\mathbf{x}_t) = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\ \epsilon_\theta(\mathbf{x}_t, t) \right). \tag{16}$$

Recall the data consistency loss function

$$R_{\mathcal{H}}(Q(\mathbf{x}_t)) = \|\mathcal{H}(Q(\mathbf{x}_t))\|_2^2, \tag{17}$$

which evaluates the violation of the constraint function $\mathcal{H}$ on the clean estimate $\hat{\mathbf{x}}_{0|t} = Q(\mathbf{x}_t)$.

We are interested in computing the gradient of (17) with respect to the noisy input $\mathbf{x}_t$, i.e., $\nabla_{\mathbf{x}_t} R_{\mathcal{H}}(Q(\mathbf{x}_t))$. Applying the chain rule yields:

$$\begin{aligned}
\nabla_{\mathbf{x}_t} R_{\mathcal{H}}(Q(\mathbf{x}_t)) &= \nabla_{\mathbf{x}_t} \|\mathcal{H}(Q(\mathbf{x}_t))\|_2^2 \\
&= \nabla_{\mathbf{x}_t} \left(\mathcal{H}(Q(\mathbf{x}_t))^\top \mathcal{H}(Q(\mathbf{x}_t)) \right) \\
&= 2 \left(\nabla_{\mathbf{x}_t} \mathcal{H}(Q(\mathbf{x}_t)) \right)^\top \mathcal{H}(Q(\mathbf{x}_t)) \\
&= 2 \left(J_{\mathcal{H}}(Q(\mathbf{x}_t))\ J_Q(\mathbf{x}_t) \right)^\top \mathcal{H}(Q(\mathbf{x}_t)) \\
&= 2 J_Q(\mathbf{x}_t)^\top J_{\mathcal{H}}(Q(\mathbf{x}_t))^\top \mathcal{H}(Q(\mathbf{x}_t)),
\end{aligned} \tag{18}$$

where $J_Q(\mathbf{x}_t)$ is the Jacobian of the map Q and $J_{\mathcal{H}}(Q(\mathbf{x}_t))$ is the Jacobian of the constraint function $\mathcal{H}$ evaluated at the clean data estimate $\hat{\mathbf{x}}_{0|t} = Q(\mathbf{x}_t)$.

Now, according to Proposition 2 in [40], the map Q behaves locally as an orthogonal projection onto the clean data manifold $\mathcal{M}$:

$$Q(\mathbf{x}_t) \in \mathcal{M}, \tag{19}$$

$$J_Q(\mathbf{x}_t) = J_Q(\mathbf{x}_t)^\top = J_Q(\mathbf{x}_t)^2, \tag{20}$$

which implies that $J_Q(\mathbf{x}_t)$ is an orthogonal projection that projects onto the tangent space $T_{Q(\mathbf{x}_t)}\mathcal{M}$ at $Q(\mathbf{x}_t)$. As a result, the gradient

$$\nabla_{\mathbf{x}_t} R_{\mathcal{H}}(Q(\mathbf{x}_t)) = J_Q(\mathbf{x}_t)^\top v, \tag{21}$$

where $v = 2J_{\mathcal{H}}(Q(\mathbf{x}_t))^\top \mathcal{H}(Q(\mathbf{x}_t))$ due to (18), is already in the tangent space $T_{Q(\mathbf{x}_t)}\mathcal{M}$. Therefore, projecting it onto the tangent space does not change it:

$$\mathcal{P}_{T_{Q(\mathbf{x}_t)}\mathcal{M}} \left(\nabla_{\mathbf{x}_t} R_{\mathcal{H}}(Q(\mathbf{x}_t)) \right) = \nabla_{\mathbf{x}_t} R_{\mathcal{H}}(Q(\mathbf{x}_t)). \tag{22}$$

Hence, the gradient of the constraint residual function $R_{\mathcal{H}}(Q(\mathbf{x}_t))$ with respect to the noisy sample $\mathbf{x}_t$ lies in the tangent space of the clean data manifold $\mathcal{M}$ at $\hat{\mathbf{x}}_{0|t}$; thus, no projection is needed. □

Substituting the result from Theorem 1 in (13) yields the following practical correction rule for gradient guidance:

$$\hat{\mathbf{x}}'_{0|t} = \hat{\mathbf{x}}_{0|t} - \lambda_t\ \nabla_{\mathbf{x}_t} R_{\mathcal{H}}(\hat{\mathbf{x}}_{0|t}), \tag{23}$$

where λ_t is a hyperparameter controlling the strength of the guidance at step t. Although a moderate λ_t encourages the generation of feasible samples, excessively large values can distort the sampling trajectory, pushing samples off the data manifold or even causing instability. This occurs because large values of λ_t violate the affine subspace assumption in Theorem 1, thereby undermining the validity of the guidance direction. Conversely, small λ_t results in samples that violate the constraints. Hence, λ_t should be carefully tuned in practice to balance constraint satisfaction and statistical representation.

Algorithm 3 : Sampling with gradient guidance

Inputs: trained neural network $\boldsymbol{\epsilon}_\theta$, noise schedule $\{\alpha_t\}_{t=1}^T$, noise scale σ_t, guidance scale λ_t
Outputs: new data point $\tilde{\mathbf{x}}_0$

$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I}_{4B})$
for $t = T-1$ to 0 **do**
  $\hat{\mathbf{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right)$
  $\hat{\mathbf{x}}'_0 \leftarrow \hat{\mathbf{x}}_0 - \lambda_t\ \left(\nabla_{\mathbf{x}_t} R_{\mathcal{H}}(\hat{\mathbf{x}}_0) + \nabla_{\mathbf{x}_t} R_{\mathcal{G}}(\hat{\mathbf{x}}_0) \right)$
  $z \sim \mathcal{N}(0, \mathbf{I}_{4B})$
  $\mathbf{x}_{t-1} \leftarrow \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \hat{\mathbf{x}}'_0 + \sigma_t \mathbf{z}$
return $\tilde{\mathbf{x}}_0$

Figure 3b illustrates the geometry of sampling with the manifold-constrained gradient guidance. Unlike standard sampling, an additional step is incorporated based on the gradient of the constraint residual function $R_{\mathcal{H}}(\hat{\mathbf{x}}_{0|t})$. The guidance term steers the sampling trajectory toward the intersection $\mathbf{x}_0^\star$ of the constraint $\mathcal{H}(\mathbf{x}) = 0$ and the clean data manifold $\mathcal{M}$. Since the guidance term is tangential to the clean data manifold $\mathcal{M}$, $\hat{\mathbf{x}}'_{0|t}$ remains on the clean data manifold $\mathcal{M}$, ensuring that the final sample is both feasible and statistically representative.

To enforce inequality constraints (10), we implement a similar approach. Consider the following inequality constraints

$$\mathcal{G}(\mathbf{x}) \le 0, \tag{24}$$

for which the residual function $R_{\mathcal{G}}(\cdot)$ is defined as

$$R_{\mathcal{G}}(\mathbf{x}) = \|\max(\mathcal{G}(\mathbf{x}), 0)\|_2^2. \tag{25}$$

The correction rule is similar to (23), with the gradient guidance term defined as

$$-\nabla_{\mathbf{x}_t} R_{\mathcal{G}}(\hat{\mathbf{x}}_{0|t}) = -\nabla_{\mathbf{x}_t} \|\max(\mathcal{G}(\hat{\mathbf{x}}_{0|t}), 0)\|_2^2. \tag{26}$$

The gradient guidance terms are incorporated into the sampling process as shown in Algorithm 3. The full expressions of the guidance terms, specific to the AC power flow constraints, are provided in the e-companion of this paper [42]. Due to Step 4, the guidance terms modify the sampling path at each reverse diffusion step. We also omit the subscript t from the estimated clean data $\hat{\mathbf{x}}_{0|t}$ and denote it by $\hat{\mathbf{x}}_0$.

V. Practical Implementation via Variable Decoupling and Normalization

We present two practical techniques that leverage domain knowledge in power systems to improve (i) computational efficiency of the proposed constrained diffusion model and (ii) scale consistency of the gradient guidance during sampling.

A. Variable Decoupling for Computational Efficiency

In high-voltage transmission systems, active power injection $\mathbf{p}$ highly correlates with $\boldsymbol{\theta}$ and less so with voltage magnitude $\mathbf{v}$, while reactive power injection $\mathbf{q}$ primarily correlates with $\mathbf{v}$ and weakly correlates with phase angle $\boldsymbol{\theta}$ [43], [44]. This observation underlies the classical fast decoupled power flow

Algorithm 4 : Training the diffusion model under variable decoupling

Inputs: initialized neural networks $\boldsymbol{\epsilon}_{\theta_1}$ and $\boldsymbol{\epsilon}_{\theta_2}$, noise schedule $\{\alpha_t\}_{t=1}^T$, dataset of $\mathbf{x}_0$'s sampled from q_0
Outputs: trained neural networks $\boldsymbol{\epsilon}_{\theta_1}$ and $\boldsymbol{\epsilon}_{\theta_2}$

repeat
  $\mathbf{x}_0 \sim q_0(\mathbf{x}_0)$
  $\mathbf{x}_0^{(1)}, \mathbf{x}_0^{(2)} \leftarrow \mathbf{x}_0$ ▷ split vector
  $t \sim \text{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2 \sim \mathcal{N}(0, \mathbf{I}_{2B})$
  Take gradient descent step on

$$\mathbb{E}_{\mathbf{x}_0^{(1)}, \boldsymbol{\epsilon}_1, t} \left\| \boldsymbol{\epsilon}_1 - \boldsymbol{\epsilon}_{\theta_1}(\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0^{(1)} + \sqrt{1-\bar{\alpha}_t}\, \boldsymbol{\epsilon}_1, t) \right\|^2 + \mathbb{E}_{\mathbf{x}_0^{(2)}, \boldsymbol{\epsilon}_2, t} \left\| \boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_{\theta_2}(\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0^{(2)} + \sqrt{1-\bar{\alpha}_t}\, \boldsymbol{\epsilon}_2, t) \right\|^2$$

until converged

method and motivates our variable decoupling strategy. We split the full vector $\mathbf{x}_0 = (\mathbf{p}, \mathbf{q}, \mathbf{v}, \boldsymbol{\theta}) \in \mathbb{R}^{4B}$ into two lower-dimensional vectors $\mathbf{x}_0^{(1)} = (\mathbf{p}, \boldsymbol{\theta}) \in \mathbb{R}^{2B}$ and $\mathbf{x}_0^{(2)} = (\mathbf{q}, \mathbf{v}) \in \mathbb{R}^{2B}$. The diffusion loss $\mathcal{L}_{\text{diff}}$ in (5) is thus split between two denoiser neural networks:

$$\mathcal{L}_{\text{diff}} = \mathcal{L}_{\text{diff}}^{(1)} + \mathcal{L}_{\text{diff}}^{(2)}, \tag{27}$$

where $\mathcal{L}_{\text{diff}}^{(1)}$ and $\mathcal{L}_{\text{diff}}^{(2)}$ correspond to $\mathbf{x}_0^{(1)}$ and $\mathbf{x}_0^{(2)}$, respectively. Due to (5), $\mathcal{L}_{\text{diff}}^{(1)}$ is defined as

$$\mathcal{L}_{\text{diff}}^{(1)} = \mathbb{E}_{\mathbf{x}_0^{(1)}, \boldsymbol{\epsilon}_1, t} \left\| \boldsymbol{\epsilon}_1 - \boldsymbol{\epsilon}_{\theta_1}(\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0^{(1)} + \sqrt{1-\bar{\alpha}_t}\, \boldsymbol{\epsilon}_1, t) \right\|^2, \tag{28}$$

where $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_{\theta_1}$ are the actual and predicted noise at time step t of the forward process. Similarly, $\mathcal{L}_{\text{diff}}^{(2)}$ is defined as

$$\mathcal{L}_{\text{diff}}^{(2)} = \mathbb{E}_{\mathbf{x}_0^{(2)}, \boldsymbol{\epsilon}_2, t} \left\| \boldsymbol{\epsilon}_2 - \boldsymbol{\epsilon}_{\theta_2}(\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0^{(2)} + \sqrt{1-\bar{\alpha}_t}\, \boldsymbol{\epsilon}_2, t) \right\|^2, \tag{29}$$

where $\boldsymbol{\epsilon}_2$ and $\boldsymbol{\epsilon}_{\theta_2}$ are the actual and predicted noise. Algorithm 4 demonstrates the training of the diffusion model under variable decoupling. Similarly, to generate new data points, Algorithm 2 can be adapted based on the variable decoupling approach, resulting in Algorithm 5.

B. Normalization for Scale-Consistent Gradient Guidance

Normalization in power systems is traditionally achieved via p.u. transformation [45], bringing all variables to a common basis. However, p.u. transformation alone is insufficient for diffusion, as it does not normalize the variables to a unified numerical range. In fact, the power flow variables $\mathbf{x}_0 = (\mathbf{p}, \mathbf{q}, \mathbf{v}, \boldsymbol{\theta})$ in p.u. still have different numerical ranges and scales. Hence, when computing the gradient guidance term, the difference in scales becomes problematic. Specifically, the elements of the resulting guidance vector inherit the magnitudes of the corresponding variables. As a result, a single scalar guidance scale λ may have inconsistent effects, hindering both the convergence and constraint satisfaction during sampling.

To address this issue, we propose a new normalization of the power flow variables to ensure that the guidance vector has a comparable scale across all the variable types. First, we apply the min-max normalization to the real data prior to training the denoiser, ensuring that all the power flow variables are mapped to the range $[-1, 1]$. Specifically, for each data point $\mathbf{x}_0$, its i-th variable $\mathbf{x}_{0,i}$ is transformed as

$$\mathbf{x}_{0,i}^{\text{norm}} = 2\frac{\mathbf{x}_{0,i} - \mathbf{x}_{0,i}^{\min}}{\mathbf{x}_{0,i}^{\max} - \mathbf{x}_{0,i}^{\min}} - 1, \quad \forall i = 1, \cdots, 4B, \tag{30}$$

where $\mathbf{x}_{0,i}^{\min}$ and $\mathbf{x}_{0,i}^{\max}$ denote the minimum and maximum values of the i-th variable in the dataset. The denoiser is then trained using this normalized dataset. Consequently, the entire sampling process is carried out in the normalized space.

To compute the gradient guidance term, we first denormalize the current estimate of the clean sample $\hat{\mathbf{x}}_0$ using the denormalization function $f_{\text{de}}(\cdot)$:

$$f_{\text{de}}(\hat{\mathbf{x}}_{0,i}^{\text{norm}}) = \left(\frac{\hat{\mathbf{x}}_{0,i}^{\text{norm}} + 1}{2}\right)\left(\mathbf{x}_{0,i}^{\max} - \mathbf{x}_{0,i}^{\min}\right) + \mathbf{x}_{0,i}^{\min}. \tag{31}$$

The residuals are then evaluated based on actual values. Yet, the derivative in the gradient guidance term is taken with respect to normalized values. By chain rule, we thus have

$$\nabla_{\hat{\mathbf{x}}_0^{\text{norm}}} R_{\mathcal{H}}(\hat{\mathbf{x}}_0^{\text{norm}}) = \frac{\partial R_{\mathcal{H}}(f_{\text{de}}(\hat{\mathbf{x}}_0^{\text{norm}}))}{\partial f_{\text{de}}(\hat{\mathbf{x}}_0^{\text{norm}})} \frac{\partial f_{\text{de}}(\hat{\mathbf{x}}_0^{\text{norm}})}{\partial \hat{\mathbf{x}}_0^{\text{norm}}}. \tag{32}$$

This approach ensures numerical stability during sampling, while enabling scale-consistent guidance.

Algorithm 5 : Sampling with gradient guidance under variable decoupling

Inputs: trained neural networks $\boldsymbol{\epsilon}_{\theta_1}$ and $\boldsymbol{\epsilon}_{\theta_2}$, noise schedule $\{\alpha_t\}_{t=1}^T$, noise scale σ_t, guidance scale λ_t
Outputs: new data point $\tilde{\mathbf{x}}_0$

$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I}_{4B})$
$\mathbf{x}_T^{(1)}, \mathbf{x}_T^{(2)} \leftarrow \mathbf{x}_T$ ▷ split vector
for $t = T-1$ to 0 **do**
  $\hat{\mathbf{x}}_0^{(1)} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t^{(1)} - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\theta_1}(\mathbf{x}_t^{(1)}, t)\right)$
  $\hat{\mathbf{x}}_0^{(2)} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t^{(2)} - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\theta_2}(\mathbf{x}_t^{(2)}, t)\right)$
  $\hat{\mathbf{x}}_0 \leftarrow \hat{\mathbf{x}}_0^{(1)} \,\|\, \hat{\mathbf{x}}_0^{(2)}$ ▷ concatenate vectors
  $\hat{\mathbf{x}}_0' \leftarrow \hat{\mathbf{x}}_0 - \lambda_t\ (\nabla_{\mathbf{x}_t} R_{\mathcal{H}}(\hat{\mathbf{x}}_0) + \nabla_{\mathbf{x}_t} R_{\mathcal{G}}(\hat{\mathbf{x}}_0))$
  $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_{4B})$
  $\mathbf{x}_{t-1} \leftarrow \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{x}}_0' + \sigma_t \mathbf{z}$
  $\mathbf{x}_{t-1}^{(1)}, \mathbf{x}_{t-1}^{(2)} \leftarrow \mathbf{x}_{t-1}$ ▷ split vector
return $\tilde{\mathbf{x}}_0$

VI. Numerical Results

We evaluate the performance of the proposed constrained diffusion model for synthesizing power flow datasets. We run experiments on three benchmark systems: PJM 5-bus system, IEEE 24-bus system, and IEEE 118-bus system [46]. The effectiveness of our approach is evaluated through three analyses: (i) comparing the statistical properties of the synthesized data with those of the ground truth data (Sec. VI-B), (ii) analyzing constraint satisfaction (Sec. VI-C), and (iii) evaluating the utility of the synthesized data in a ML application (Sec. VI-D).

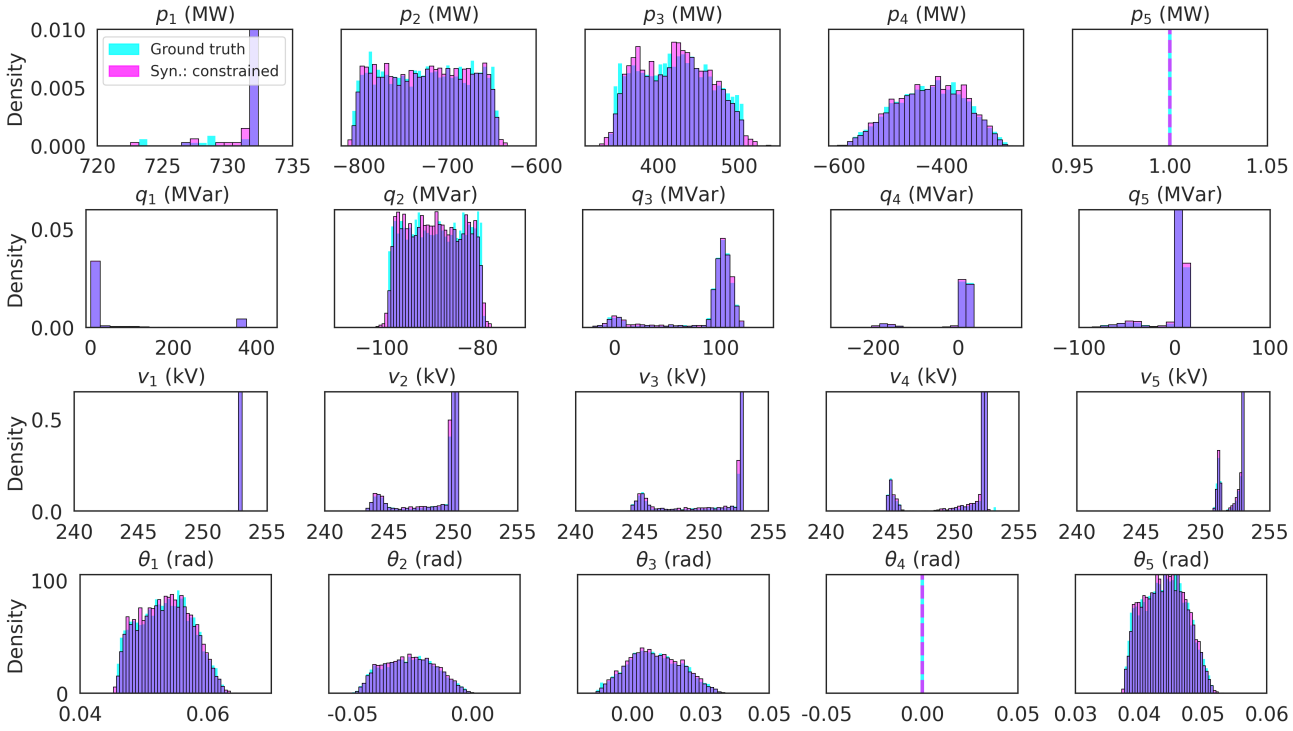


Fig. 4. Histograms of the ground truth versus synthetic power flow data points for active power injections (first row), reactive power injections (second row), voltage magnitudes (third row), and phase angles (forth row) at each bus in the PJM 5-bus system.

A. Experimental Setup

Since real-world power flow datasets are not publicly available, we generate the ground truth dataset by applying random perturbations around the nominal load $s_b^{\text{nom}} = (p_{d,b}^{\text{nom}}, q_{d,b}^{\text{nom}})$ at each bus $b \in \mathcal{B}$ of the benchmark systems. We uniformly sample active and reactive power demands at each bus $s_b = (p_{d,b}, q_{d,b}) \sim \text{Uniform}\,(0.8\, s_b^{\text{nom}}, s_b^{\text{nom}})$, and then solve the AC-OPF problem to extract the feasible solutions $\mathbf{p}, \mathbf{q}, \mathbf{v}, \boldsymbol{\theta}$. By stacking them together into a single data point $(\mathbf{p}_i, \mathbf{q}_i, \mathbf{v}_i, \boldsymbol{\theta}_i)$, we obtain the ground truth dataset $\mathcal{D} = \{(\mathbf{p}_i, \mathbf{q}_i, \mathbf{v}_i, \boldsymbol{\theta}_i)\}_{i=1}^N$.

B. Statistical Similarity

The histograms of the ground truth and synthetic power flow variables for the PJM 5-bus system are given in Fig. 4. The synthesized variables capture the underlying distribution of the ground truth data. For each class of variables, the synthetic data not only aligns well with the support of the ground truth data but also successfully captures the modes. The proposed diffusion model also ensures similarity of the joint probability distributions, as shown in Fig. 5a and Fig. 5b depicting the joint distributions of the active and reactive power injections, and voltage magnitude and phase angel, respectively. If the ground truth data exhibits a multi-modal structure, the synthetic data points successfully capture all modes. Moreover, in regions where the ground truth data is denser, the synthetic data points are also more concentrated, whereas in regions where the density of the ground truth data points decreases, the synthetic data points become more sparse. Another important property is the coverage capability, as shown in Fig. 5a and Fig. 5b, where the synthesized data points closely span the entire domain of the ground truth data.

TABLE I
WASSERSTEIN DISTANCES BETWEEN THE GROUND TRUTH $\mathcal{D}$ AND SYNTHETIC DATA $\widetilde{\mathcal{D}}$

Distance between...	5-Bus	24-Bus	118-Bus
...$\mathcal{D}$ and $\widetilde{\mathcal{D}}$ w/o guidance	0.442	0.607	0.622
...$\mathcal{D}$ and $\widetilde{\mathcal{D}}$ w/ guidance	0.382	0.585	0.597

To quantify the similarity of the ground truth and synthetic datasets, we use the type-1 Wasserstein distance between these datasets, defined as

$$W_1(\mathcal{D}, \widetilde{\mathcal{D}}) = \min_{\gamma \in \Gamma(\mathcal{D}, \widetilde{\mathcal{D}})} \sum_{i=1}^{N} \sum_{j=1}^{M} \gamma_{ij} \|\mathbf{x}_i - \tilde{\mathbf{x}}_j\|_2, \tag{33}$$

where $\Gamma(\mathcal{D}, \widetilde{\mathcal{D}})$ represents the set of all valid ways to assign mass between the ground truth and synthetic samples [47]. The results for synthetic datasets obtained with and without gradient guidance are summarized in Table I. Lower Wasserstein distances indicate closer alignment between synthetic and ground truth distributions. Across all the test systems, the distances remain low, showing that the synthesized power flow data closely mirrors the ground truth data. Enforcing the constraints during sampling further reduces the Wasserstein distance consistently. Thus, we validate that constraint enforcement not only promotes feasibility but also enhances the statistical similarity of the ground truth and synthetic datasets.

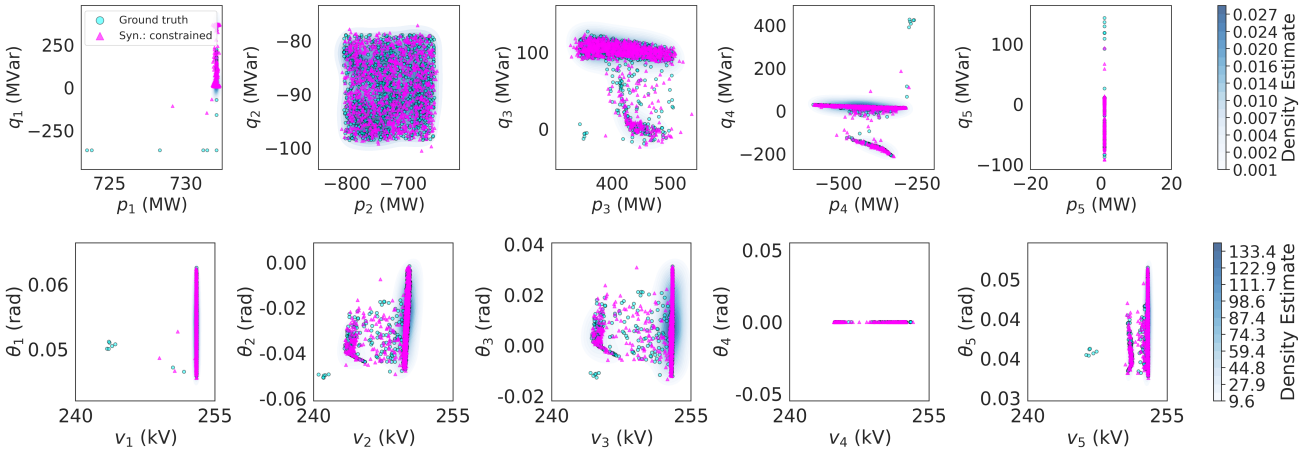


Fig. 5. 2D scatter plots with density estimates of the active and reactive power injection (top row), and voltage magnitude and phase angle (bottom row) at each bus in the PJM 5-bus system, comparing the joint distributions of the ground truth and constrained synthetic datasets. The plots highlight the ability of the diffusion model to replicate the underlying pattern, domain, and multi-modal structure of the ground truth data.

C. Constraint Satisfaction

We evaluate constraint satisfaction of the synthetic datasets generated with and without gradient guidance, specifically focusing on the active and reactive power balance at each bus. Figure 6 presents the histograms of violation magnitudes in the PJM 5-bus system: without the guidance mechanism, a significant portion of the generated samples exhibit non-negligible violations at buses 1, 2, 3, and 5 for the active power balance constraints. With guidance, the vast majority of the samples show near-zero active power mismatch at all buses, indicating strong constraint satisfaction. Similar observations hold for the reactive power balance constraints.

For the IEEE 24-bus system, Table II reports the mean and variance of the active and reactive power mismatches at each bus. Similar to the PJM 5-bus system, gradient guidance significantly reduces both the mean and variance of constraint violations across most buses. For the IEEE 118-bus system, the results are visualized in Fig. 7 comparing the mean and variance of the power mismatches with and without guidance. Guidance leads to a noticeable improvement for some buses, particularly buses 4 and 5, where the baseline violation is relatively large. For most other buses, the initial violation magnitude is small, which limits the impact of the guidance mechanism. Furthermore, for some buses, the mismatch mean under the constrained sampling is slightly larger. However, this does not contradict the overall effectiveness of the guidance mechanism. We observe that variance plays a more critical role than the mean in determining the quality of constraint satisfaction. That is, a model with a low mismatch mean but high variance may still generate many samples with large constraint violations. The proposed guidance mechanism, instead, ensures that most generated samples remain close to the full satisfaction of the physical constraints.

TABLE II
MEAN AND STANDARD DEVIATION OF POWER MISMATCHES OF SYNTHESIZED DATA POINTS FOR THE IEEE 24-BUS SYSTEM UNDER CONSTRAINED AND UNCONSTRAINED SAMPLING (λ=10^{-4} VS λ=0).

Bus	Δp (MW)		Δq (MVar)	
	λ=0	λ=10^{-4}	λ=0	λ=10^{-4}
1	-12.50 ± 207.10	1.03 ± 4.80	-1.20 ± 59.80	-0.60 ± 5.50
2	10.10 ± 153.00	-1.50 ± 4.90	-2.56 ± 56.20	0.77 ± 5.40
3	0.24 ± 7.10	-0.15 ± 2.80	-0.06 ± 2.60	-0.08 ± 1.00
4	-0.30 ± 9.10	-0.03 ± 2.80	0.04 ± 1.40	-0.03 ± 1.00
5	0.53 ± 6.10	0.05 ± 4.00	-0.04 ± 2.80	-0.04 ± 1.40
6	0.04 ± 26.40	-0.85 ± 3.20	-0.19 ± 12.80	0.11 ± 1.00
7	-0.94 ± 17.80	-0.41 ± 3.50	0.30 ± 6.60	0.07 ± 1.40
8	1.00 ± 22.60	0.39 ± 4.60	-0.89 ± 24.80	-0.15 ± 1.70
9	0.33 ± 8.20	0.41 ± 3.90	-0.36 ± 8.30	-0.14 ± 1.00
10	0.85 ± 7.30	0.92 ± 4.90	-0.74 ± 17.50	-0.21 ± 1.00
11	-0.38 ± 7.30	-0.21 ± 2.60	0.01 ± 2.20	0.05 ± 1.40
12	-0.92 ± 24.80	-0.15 ± 2.20	-0.06 ± 5.00	-0.02 ± 1.00
13	0.02 ± 2.20	0.02 ± 1.40	0.01 ± 4.50	0.16 ± 2.60
14	-0.65 ± 19.0	-0.09 ± 3.50	0.02 ± 2.20	-0.17 ± 1.70
15	-0.28 ± 71.80	-0.34 ± 4.90	-0.42 ± 14.70	-0.11 ± 1.70
16	6.97 ± 189.00	0.22 ± 3.90	-1.84 ± 54.70	0.11 ± 1.70
17	-1.04 ± 19.60	-0.23 ± 4.40	-0.48 ± 19.40	0.07 ± 1.00
18	-1.35 ± 94.20	0.48 ± 3.00	-0.28 ± 3.60	-0.10 ± 0.00
19	-2.17 ± 78.00	-0.17 ± 4.80	-0.20 ± 5.70	0.05 ± 1.00
20	-3.14 ± 55.60	-0.55 ± 4.00	0.30 ± 3.70	0.03 ± 1.00
21	2.00 ± 85.80	0.16 ± 3.90	-0.64 ± 21.30	0.07 ± 1.00
22	-0.98 ± 26.90	-0.22 ± 3.30	-0.13 ± 4.00	0.00 ± 0.00
23	3.23 ± 68.60	0.51 ± 4.10	-0.52 ± 14.60	-0.04 ± 1.00
24	0.30 ± 7.80	0.06 ± 2.80	0.09 ± 4.70	0.04 ± 1.00

D. Utility of Synthetic Data in Downstream ML Task

As an additional measure of quality, we study how well the synthesized power flow data performs in a downstream learning task. Specifically, we examine whether constrained synthetic data better supports the learning of efficient warm-start for the Newton–Raphson power flow solver [37]. We train a neural network $f : \mathbb{R}^{d_k} \to \mathbb{R}^{d_u}$ that maps the known inputs $\mathbf{x} \in \mathbb{R}^{d_k}$ (e.g., $\mathbf{p}$, $\mathbf{q}$, $\mathbf{v}$, or $\boldsymbol{\theta}$, depending on bus types) to the corresponding unknown outputs $\mathbf{y} \in \mathbb{R}^{d_u}$. The dimensions d_k and d_u depend on the specific test case. We generate two synthetic training datasets of equal size under constrained and unconstrained sampling. Using each dataset, we train a separate neural network to predict $\mathbf{y}$ from $\mathbf{x}$. We use a fully connected feedforward neural network architecture. More

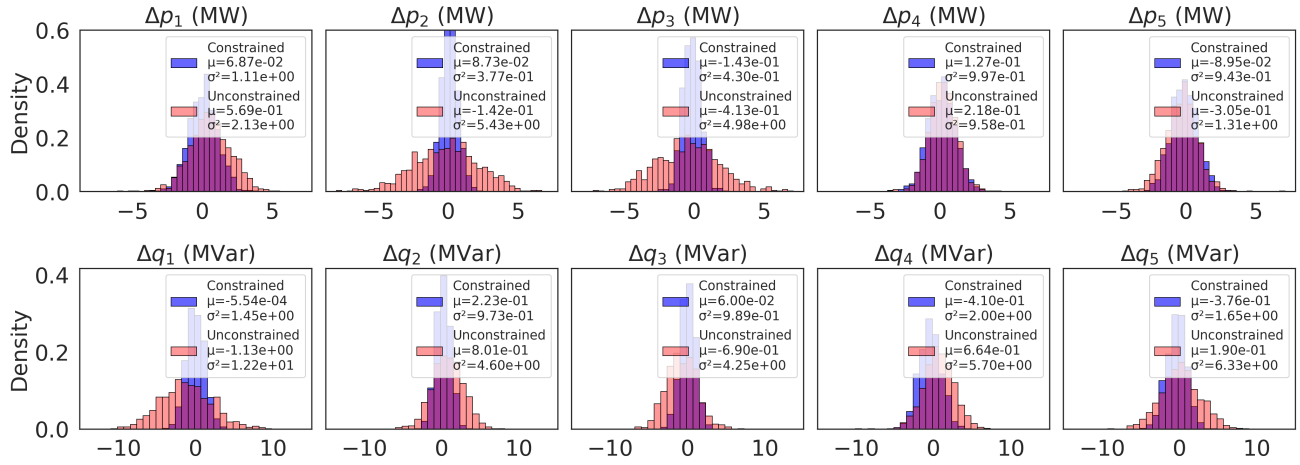


Fig. 6. Histograms of violation magnitudes for the active (top row) and reactive (bottom row) power balance constraints in the PJM 5-bus system, comparing synthesized data points under constrained and unconstrained sampling (λ=10^{-2} vs λ=0).

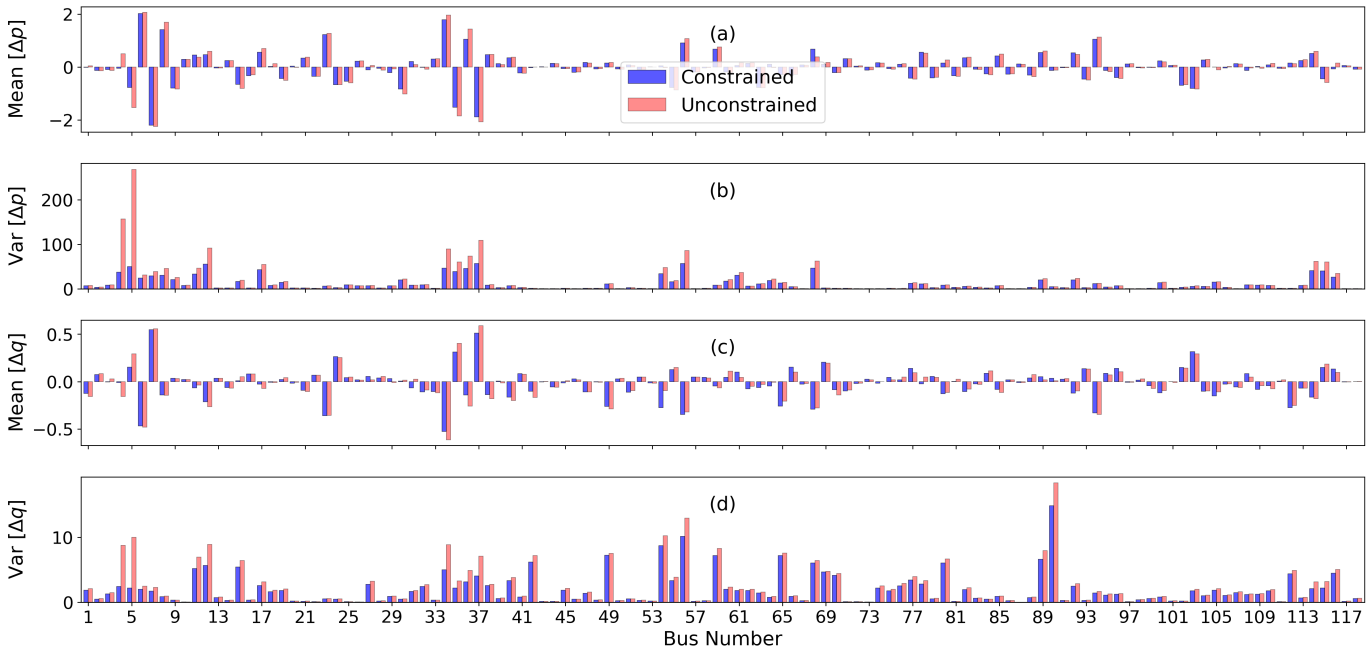


Fig. 7. Comparison of per-bus (a) mean (MW) and (b) variance (MW^2) of the active power mismatches, and (c) mean (MVar) and (d) variance (MVar^2) of the reactive power mismatches for the synthesized data on the IEEE 118-bus system under constrained and unconstrained sampling (λ=5×10^{-4} vs λ=0).

advanced architectures could improve accuracy but would not change the relative comparison between datasets. The performance of the models is then evaluated on a common test dataset. The results in Table III show that models trained on constrained synthetic data consistently yield smaller active and reactive power mismatches than those trained on unconstrained data across all the test cases. Although the gap between synthetic and ground-truth data remains, enforcing the power flow constraints during data generation clearly leads to predictions that better respect the underlying physics of power systems.

VII. Conclusion

This paper aims to generate statistically representative and physically consistent synthetic power flow datasets. We develop a diffusion model that integrates the AC power flow constraints into the data generation process through manifold-constrained gradient guidance. Numerical experiments on IEEE benchmark systems show that the model produces synthetic datasets with high statistical similarity to real data while achieving high physical feasibility. The proposed method can serve as a practical tool for system operators to generate high-quality power flow data suitable for public release and capable of supporting a wide range of downstream ML applications.

References

[1] M. Klamkin *et al.*, "PGLearn–an open-source learning toolkit for optimal power flow," *arXiv preprint arXiv:2505.22825*, 2025.

[2] M. Gillioz *et al.*, "A large synthetic dataset for machine learning applications in power transmission grids," *Scientific Data*, vol. 12, no. 1, p. 168, 2025.

TABLE III
MEAN AND STANDARD DEVIATION OF TOTAL ACTIVE AND REACTIVE POWER MISMATCHES ACROSS ALL BUSES FOR MODELS TRAINED ON GROUND TRUTH, CONSTRAINED SYNTHETIC, AND UNCONSTRAINED SYNTHETIC DATASETS (P.U.).

Test Case	Training Data	Mean ± Std. $\lvert\Delta p\rvert$	Mean ± Std. $\lvert\Delta q\rvert$
5-Bus	Ground Truth	0.0124 ± 0.0173	0.0242 ± 0.0469
	Constrained Syn.	0.0200 ± 0.0173	0.0434 ± 0.0944
	Unconstrained Syn.	0.0293 ± 0.0282	0.0557 ± 0.0916
24-Bus	Ground Truth	0.1733 ± 0.0068	0.0415 ± 0.0003
	Constrained Syn.	0.3425 ± 0.0143	0.3093 ± 0.0376
	Unconstrained Syn.	0.4522 ± 0.0292	0.3444 ± 0.0827
118-Bus	Ground Truth	1.6425 ± 0.0273	0.5838 ± 0.0066
	Constrained Syn.	3.9106 ± 0.0991	1.5219 ± 0.0311
	Unconstrained Syn.	4.0596 ± 0.1117	1.6318 ± 0.0405

[3] A. Venzke *et al.*, "Efficient creation of datasets for data-driven power system applications," *Electr. Power Syst. Res*, vol. 190, p. 106614, 2021.

[4] P. Van Hentenryck, "Machine learning for optimal power flows," *Tutorials in Operations Research: Emerging Optimization Methods and Modeling Techniques with Applications*, pp. 62–82, 2021.

[5] L. Pagnier and M. Chertkov, "Embedding power flow into machine learning for parameter and state estimation," *arXiv preprint arXiv:2103.14251*, 2021.

[6] T. Joswig-Jones *et al.*, "OPF-Learn: An open-source framework for creating representative AC optimal power flow datasets," *2022 IEEE PES ISGT*, pp. 1–5, 2022.

[7] A.-A. B. Bugaje *et al.*, "Generating quality datasets for real-time security assessment: Balancing historically relevant and rare feasible operating conditions," *IJEPES*, vol. 154, p. 109427, 2023.

[8] M. Hoseinpour *et al.*, "Privacy-preserving and approximately truthful local electricity markets: A differentially private VCG mechanism," *IEEE Trans. on Smart Grid*, vol. 15, no. 2, pp. 1991–2003, 2023.

[9] V. Dvorkin and A. Botterud, "Differentially private algorithms for synthetic power system datasets," *IEEE Control Systems Letters*, vol. 7, pp. 2053–2058, 2023.

[10] S. Wu and V. Dvorkin, "Synthesizing grid data with cyber resilience and privacy guarantees," *IEEE Control Systems Letters*, 2025.

[11] S. Zhang *et al.*, "Generating synthetic net load data with physics-informed diffusion model," *arXiv preprint arXiv:2406.01913*, 2024.

[12] C. Wang *et al.*, "Generating multivariate load states using a conditional variational autoencoder," *Electr. Power Syst. Res.*, vol. 213, p. 108603, 2022.

[13] Z. Pan *et al.*, "Data-driven EV load profiles generation using a variational auto-encoder," *Energies*, vol. 12, no. 5, p. 849, 2019.

[14] C. Wang *et al.*, "Generating contextual load profiles using a conditional variational autoencoder," in *2022 IEEE PES ISGT-Europe*. IEEE, 2022, p. 1–6.

[15] A. B. L. Larsen *et al.*, "Autoencoding beyond pixels using a learned similarity metric," in *ICML*. PMLR, 2016, pp. 1558–1566.

[16] C. Zhang *et al.*, "Generative adversarial network for synthetic time series data generation in smart grids," in *2018 IEEE SmartGridComm*. IEEE, 2018, pp. 1–6.

[17] Y. Gu *et al.*, "GAN-based model for residential load generation considering typical consumption patterns," in *2019 IEEE PES ISGT*. IEEE, 2019, pp. 1–5.

[18] M. N. Fekri *et al.*, "Generating energy data for machine learning with recurrent generative adversarial networks," *Energies*, vol. 13, no. 1, p. 130, 2019.

[19] S. El Kababji and P. Srikantha, "A data-driven approach for generating synthetic load patterns and usage habits," *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 4984–4995, 2020.

[20] Y. Chen *et al.*, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3265–3275, 2018.

[21] M. Hoseinpour and V. Dvorkin, "Domain-constrained diffusion models to synthesize tabular data: A case study in power systems," *arXiv preprint arXiv:2506.11281*, 2025.

[22] M. K. Singh *et al.*, "Learning to solve the AC-OPF using sensitivity-informed deep neural networks," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 2833–2846, 2021.

[23] F. Fioretto *et al.*, "Predicting AC optimal power flows: Combining deep learning and lagrangian dual methods," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 01, pp. 630–637, 2020.

[24] A. S. Zamzam and K. Baker, "Learning optimal solutions for extremely fast AC optimal power flow," in *2020 IEEE SmartGridComm*. IEEE, 2020, pp. 1–6.

[25] I. V. Nadal and S. Chevalier, "Scalable bilevel optimization for generating maximally representative OPF datasets," in *2023 IEEE PES ISGT EUROPE*. IEEE, 2023, pp. 1–6.

[26] N. Popli *et al.*, "On the robustness of machine-learnt proxies for security constrained optimal power flow solvers," *Sustain. Energy Grids Netw.*, vol. 37, p. 101265, 2024.

[27] S. Lovett *et al.*, "OPFData: Large-scale datasets for AC optimal power flow with topological perturbations," *arXiv preprint arXiv:2406.07234*, 2024.

[28] S. Kiyani *et al.*, "Decision theoretic foundations for conformal prediction: Optimal uncertainty quantification for risk-averse agents," *arXiv preprint arXiv:2502.02561*, 2025.

[29] G. Shafer and V. Vovk, "A tutorial on conformal prediction." *Journal of Machine Learning Research*, vol. 9, no. 3, 2008.

[30] Y. Bengio *et al.*, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[31] J. Stiasny *et al.*, "Closing the loop: A framework for trustworthy machine learning in power systems," *arXiv preprint arXiv:2203.07505*, 2022.

[32] A. Jabbar *et al.*, "A survey on generative adversarial networks: Variants, applications, and training," *ACM CSUR*, vol. 54, no. 8, pp. 1–49, 2021.

[33] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 8780–8794, 2021.

[34] X. Dong *et al.*, "Short-term wind power scenario generation based on conditional latent diffusion models," *IEEE Trans. Sustain. Energy*, vol. 15, no. 2, pp. 1074–1085, 2023.

[35] S. Li *et al.*, "Diffcharge: Generating EV charging scenarios via a denoising diffusion model," *IEEE Trans. Smart Grid*, vol. 15, no. 4, pp. 3936–3949, 2024.

[36] J. Ho *et al.*, "Denoising diffusion probabilistic models," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 6840–6851, 2020.

[37] D. K. Molzahn *et al.*, "A survey of relaxations and approximations of the power flow equations," *Found. Trends Electr. Energy Syst.*, vol. 4, no. 1-2, pp. 1–221, 2019.

[38] B. T. Feng *et al.*, "Neural approximate mirror maps for constrained diffusion models," *arXiv preprint arXiv:2406.12816*, 2024.

[39] G. Daras *et al.*, "Consistent diffusion models: Mitigating sampling drift by learning to be consistent," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 42 038–42 063, 2023.

[40] H. Chung *et al.*, "Improving diffusion models for inverse problems using manifold constraints," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 25 683–25 696, 2022.

[41] N. Boumal, *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

[42] M. Hoseinpour and V. Dvorkin, "Supplemental materials," Available online: https://github.com/miladpourv/constrained-diffusion-powerflow/, 2025.

[43] R. K. Portelinha *et al.*, "Fast-decoupled power flow method for integrated analysis of transmission and distribution systems," *Electr. Power Syst. Res.*, vol. 196, p. 107215, 2021.

[44] A. Monticelli, *State estimation in electric power systems: a generalized approach*. Springer Science & Business Media, 2012.

[45] J. D. D. Glover and M. S. Sarma, *Power system analysis and design*. Brooks/Cole Publishing Co., 2001.

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

[47] G. Peyré and M. Cuturi, "Computational optimal transport," 2020.