

Water Resources Research®

RESEARCH ARTICLE

10.1029/2022WR032653

Efficient Optimization of Energy Recovery From Geothermal Reservoirs With Recurrent Neural Network Predictive Models

Zhen Qin¹, Anyue Jiang¹ , Dave Faulder², Trenton T. Cladouhos² , and Behnam Jafarpour¹ 

¹Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA, ²Cyrq Energy Inc, Seattle, WA, USA

Key Points:

- An efficient workflow is developed for long-term optimization of energy recovery from geothermal reservoirs using deep learning-based proxy models (a specific design of the recurrent neural network (RNN) is used in this paper)
- Using a limited set simulated data for training the RNN model improves its extrapolation power and leads to reliable long-term predictions
- During training RNN captures the dynamics in well response data and uses it to predict long-term net energy production under different well control settings (e.g., mass flow rates)

Correspondence to:

B. Jafarpour,
jafarpou@usc.edu

Citation:

Qin, Z., Jiang, A., Faulder, D., Cladouhos, T. T., & Jafarpour, B. (2023). Efficient optimization of energy recovery from geothermal reservoirs with recurrent neural network predictive models. *Water Resources Research*, 59, e2022WR032653. <https://doi.org/10.1029/2022WR032653>

Received 22 APR 2022

Accepted 20 JAN 2023

Author Contributions:

Conceptualization: Zhen Qin, Behnam Jafarpour

Data curation: Zhen Qin

Formal analysis: Zhen Qin, Anyue Jiang, Behnam Jafarpour

Investigation: Zhen Qin

Methodology: Zhen Qin, Anyue Jiang

Resources: Behnam Jafarpour

Software: Zhen Qin, Anyue Jiang

Supervision: Behnam Jafarpour

Validation: Zhen Qin, Anyue Jiang, Behnam Jafarpour

Visualization: Zhen Qin

Writing – original draft: Zhen Qin

Writing – review & editing: Behnam Jafarpour

Abstract Improving the long-term energy production performance of geothermal reservoirs can be accomplished by optimizing field development and management plans. Reliable prediction models, however, are needed to evaluate and optimize the performance of the underlying reservoirs under various operation and development strategies. In traditional frameworks, physics-based simulation models are used to predict the energy production performance of geothermal reservoirs. However, detailed simulation models are not trivial to construct, require a reliable description of the reservoir conditions and properties, and entail high computational complexity. Data-driven predictive models can offer an efficient alternative for use in optimization workflows. This paper presents an optimization framework for net power generation in geothermal reservoirs using a variant of the recurrent neural network (RNN) as a data-driven predictive model. The RNN architecture is developed and trained to replace the simulation model for computationally efficient prediction of the objective function and its gradients with respect to the well control variables. The net power generation performance of the field is optimized by automatically adjusting the mass flow rate of production and injection wells over 12 years, using a gradient-based local search algorithm. Two field-scale examples are presented to investigate the performance of the developed data-driven prediction and optimization framework. The prediction and optimization results from the RNN model are evaluated through comparison with the results obtained by using a numerical simulation model of a real geothermal reservoir.

Plain Language Summary The paper proposes the use of recurrent neural network (RNN) architectures for capturing the dynamics of historical well response data as a function of input control variables. A trained RNN is then used as an efficient input-output dynamical model for optimization of energy recovery from geothermal reservoirs. Results from time-consuming simulation-based and fast RNN prediction models are presented and evaluated to compare the optimization strategies of the two approaches, indicating their consistency. The results suggest that RNN can be used as an efficient dynamic prediction tool for decision support and management of geothermal reservoir operations and development.

1. Introduction

Sustainable development of geothermal resources can play an important role as a clean and renewable source in meeting the future energy demand (Olasolo et al., 2016). The past decade has seen a rapid increase in recovering geothermal energy for both direct-use and electrical power generation (Bertani, 2016; Lund & Toth, 2021). The development of geothermal fields includes extracting heat (hot water or steam) from the subsurface reservoir (resource) and transporting it to the surface facilities, for example, geothermal power plants, where the heat energy is transformed into electricity that is added to the power grid. The efficiency of energy production is highly dependent on the brine enthalpy and temperature (Zarrouk & Moon, 2014). Energy production performance from geothermal reservoirs can be improved through effective field development plans and reservoir management strategies. Specifically, in an optimization problem, well control variables, for example, well flow rates or bottomhole pressure (BHP), can be adjusted to maximize a user-specified objective function over the reservoir life cycle, for example, net power generation or net present value (NPV). This task typically requires a reliable predictive model to make long-term predictions over the life cycle of the project. The predictive model is combined with an optimization algorithm to develop a model-based optimization framework, also known as model predictive control. Model-based optimization can be performed using detailed full-physics numerical simulation models, which we refer to as “simulation-based optimization.” The simulation model generates long-term predictions of the optimization objective function, as well as its gradients, which are used by the optimization algorithm to search for a solution. However, detailed simulation models are not trivial to construct and entail significant

computational runtime (Franco & Vaccaro, 2014). They also require an accurate description of highly uncertain reservoir properties and involve a challenging model calibration process (Jafarpour & McLaughlin, 2008, 2009). These issues can present a hurdle for field deployment of these techniques. Therefore, efficient fit-for-purpose proxy (or surrogate) models have been proposed to replace the simulation model during optimization to alleviate its computational burden. The resulting framework is referred to as the proxy-based optimization. Examples of proxy models include data-driven statistical models, simplified physics-based models, analytical approximations, or simulation models with coarser grid systems, see (Asher et al., 2015; Razavi et al., 2012) for a review.

Optimization with simulation models is commonly used in determining the design parameters of EGS (Chen et al., 2015; Pollack & Mukerji, 2019; Samina et al., 2019), and has been widely applied to oil and gas reservoir operation (Chen et al., 2009; Jahandideh & Jafarpour, 2018, 2019; Jansen et al., 2008; Kim & Durlofsky, 2021; Kim et al., 2022; van Essen et al., 2009). For well control optimization, gradient-based algorithms have shown promise in reservoir engineering applications. A major difficulty in using these techniques, however, is the need for efficient computation of the required gradients. To reduce the computational cost, ensemble-based techniques have been proposed to approximate the gradient with fewer simulation runs than are needed by standard finite-difference approximation methods. Examples of such techniques are Ensemble Optimization (EnOpt) (Chen et al., 2009; Do & Reynolds, 2013; Fonseca et al., 2015) and Stochastic Simplex Approximate Gradient (Stowage) (Fonseca et al., 2017). However, the quality of the approximated gradients in these methods depends on the ensemble size, which determines the number of simulation runs.

The application of optimization to geothermal reservoir operations using physics-based proxy models has been investigated by several authors. Juliusson and Horne (2013) fit an empirical model to a simulation model and use it to control the production flow rate by fixing the BHP as an empirical function of the injection rate. The proxy model is then applied to maximize the NPV of the asset by optimizing the injection rate allocations. Hecht-Méndez et al. (2013) use the line source solution as a proxy model for a borehole heat-exchanger (BHE) system. They integrate the resulting model into an optimization problem to mitigate the anomaly in the temperature field. The optimization problem is defined as minimizing the maximum temperature decline by adjusting the brine extraction rate. Sigurdardottir et al. (2015) apply a lumped parameter model to maximize the project NPV by adjusting the pump use through a mixed-integer programming algorithm (continuous production rate and discrete pump options as decision variables). Patterson et al. (2020) use analytical solutions to represent the production response of an enhanced geothermal system (EGS) to optimize the economics of production operation. While the physics-based proxy models mentioned above can provide physically consistent predictions, their underlying assumptions limits the application to specific and simple problems.

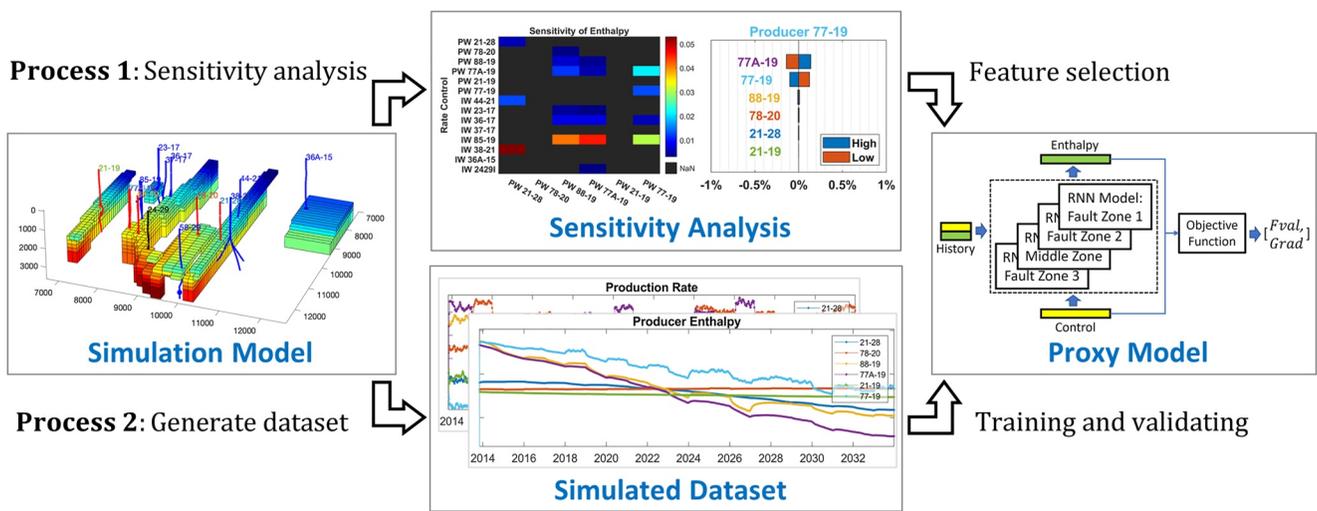
Data-driven proxy models provide fast prediction and gradient computation that can significantly improve the computational efficiency of optimization workflows. To do so, they are trained to learn the statistical input-output relationship from available data. A common type of data-driven proxy model is linear regression and its extensions that have been applied to maximize energy production from geothermal reservoirs (Chen et al., 2015; Schulte et al., 2020; Song et al., 2021). Chen et al. (2015) use multivariate adaptive regression spline (MARS) and response surface methodology (RSM) to optimize well locations in geothermal reservoirs under uncertainty. Recently, multi-objective optimization has been introduced to consider the trade-off between pressure support and thermal breakthrough (Schulte et al., 2020; Song et al., 2021). Schulte et al. (2020) apply the multi-objective particle swarm algorithm to a well placement optimization problem for a doublet geothermal system using linear regression as a proxy model (with a Gaussian process regression as the response surface model). Their work develops one proxy model for each geological realization using more than one hundred simulated input-output samples. However, the prediction from the linear regression-based proxy model shows a significant discrepancy in temperature. Song et al. (2021) integrate the genetic algorithm into multi-objective optimization to determine the operational parameters of an enhanced geothermal system (EGS). In their work, multiple linear regression (MLR) is used with simulated data to generate a proxy model to represent the objective functions. RSM has also been widely used in subsurface flow systems for optimization or model calibration purposes (Babaei et al., 2022; Chen et al., 2015, 2021; Schulte et al., 2020). RSM directly relates independent variables to the objective function and provides fast evaluation during optimization. However, RSM is typically designed as a second-order polynomial equation, which limits its application to complex large-scale problems (Baş & Boyacı, 2007). While artificial neural networks (ANN) have also been applied to response surface methodology or to replace linear regression-based proxy models, it has shown limitation in handling long-term dependencies and dynamics that are prevalent in subsurface flow system.

Different types of proxy models have also been used in the water resources literature (Razavi et al., 2012). With recent advances in machine learning, especially neural network-based deep learning models, a new class of proxy models is introduced for characterization of subsurface flow systems (Jiang & Jafarpour, 2021a, 2021b; Laloy et al., 2017; Razak & Jafarpour, 2020), and for prediction of their dynamic responses, such as time-series well response data (Gudmundsdottir & Horne, 2020; Jiang et al., 2021; Shi et al., 2021; Tian & Horne, 2017). More recently, the long-short term memory (LSTM) (Hochreiter & Schmidhuber, 1997) has been used as the proxy model with particle swarm optimization (PSO) in two-phase flow systems (Kim & Durlafsky, 2021; Kim et al., 2022). Kim and Durlafsky (2021) apply the LSTM to waterflooding optimization problems for a short-term (less than three years) operation optimization. Kim et al. (2022) also apply the LSTM to waterflooding problems under geological uncertainty. They reduce the number of independent variables by parameterizing the BHP control trajectory as cosine function. In general, deep learning models use hundreds of simulation runs to generate training samples. For example, Kim et al. (2022) run 5,000 reservoir simulations for ten geological realizations to train and test their deep learning-based proxy model. However, it is desirable to minimize the number of simulations that are needed for training a proxy model.

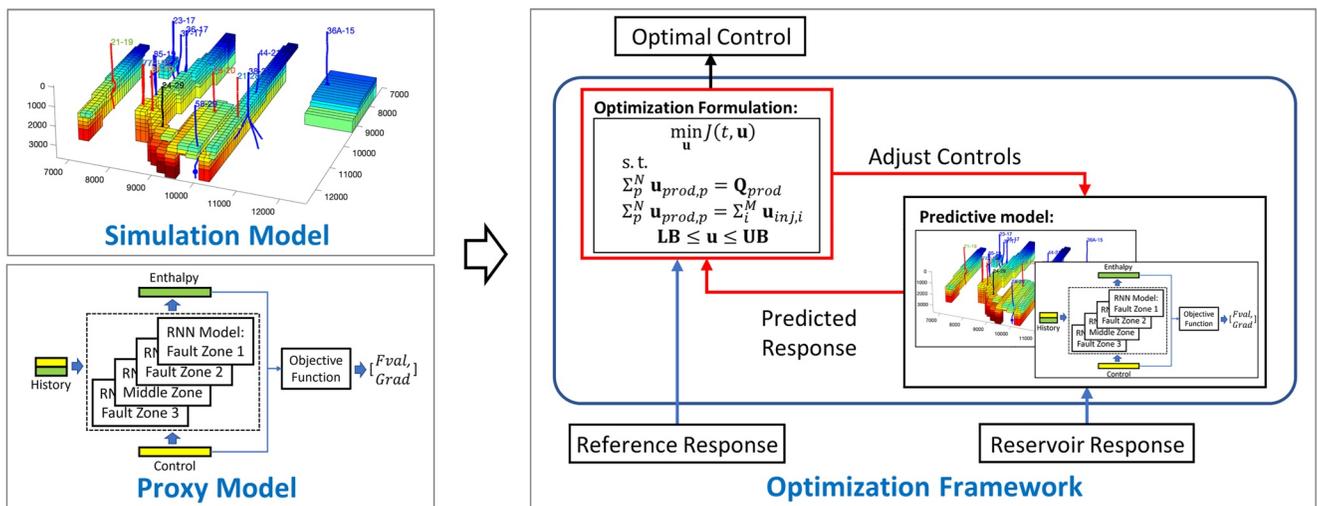
Two general limitations of the existing deep learning models that can potentially hinder their wide-spread adoption for application to subsurface flow systems are extensive training data need and weak extrapolation (long-term prediction). Typically, data-driven proxy models require extensive amount of data, especially for long-term prediction. In many scientific applications, such as geothermal systems, real data is limited and must be augmented with either simulated data or physics-based constraints to improve the reliability of predictions. However, generating extensive amounts of simulated data is computationally expensive and effective strategies must be adopted to reduce the computational demand of these methods. Another important limitation of machine learning models is their weak extrapolation power, which is particularly important in optimization problems, where exploring the solution space to find extreme values is bound to go beyond the training data range. In light of these issues, one important aspect that can aid with efficient design and implementation of data-driven proxy models is adapting them to their intended purpose/use. Given the sensitivity of machine learning models to training data, it is important to use effective sampling strategies to adapt the simulated data to the expected patterns and data ranges in the field.

Considering these important properties and the limitations of deep learning models, we propose an approach for efficient long-term prediction and optimization of geothermal energy production using RNN-type data-driven input-output proxy models. The paper demonstrates, using sensitivity analysis and comparison with simulation results, that for input-output models of sequential data, only a few simulated training samples may provide adequate data for long-term predictions. The study shows that, for long-term predictions, where the proxy model is likely to perform extrapolation over unseen control ranges, historical field data may not be sufficient. However, the addition of a few simulated data for the prediction range can significantly improve the performance of the RNN-type proxy models. Another key aspect that is the need for proper design and implementation, including effective sampling strategies, monitoring of the proxy model during optimization, and convenient retraining to integrate new data.

Two numerical experiments are performed to investigate the performance of the proxy-based optimization framework. In the first experiment, the well mass flow rates are used as control variables that are fixed over the 12-year life cycle of a geothermal project. Three sets of proxy models are trained, each using a different number of simulated data realizations (60, 100, 140). The number of realizations refers to the number of simulation runs to generate labeled data for different control inputs. In the second experiment, the well mass flow rates are allowed to change annually over the 12 years of production. Therefore, the dimension of the control variables is 12-times larger than it is in the first experiment. We investigate the computational cost of developing the proxy model in this work and perform a sensitivity analysis to assess the effect of the number of training data realizations on the prediction performance of the proxy model. The results indicate that, for training RNN models with simulated data, a modest number of simulations (e.g., 10) with different control inputs could provide the necessary training data to learn the main input-output relations of the geothermal reservoir. The trained model can then be integrated into the optimization framework to speed up the computation. The optimization results and the predictive accuracy of the proxy model are evaluated using full-physics-based simulation and optimization approach.



(a) Two processes of developing proxy model by using simulation model



(b) Process 3: Integration of predictive models into the optimization framework

Figure 1. Proposed workflow for (a) development of the proxy model and (b) development of proxy-based and simulation-based optimization framework. In the optimization framework (b), depending on whether a simulation model or a proxy model is used for prediction the workflow is referred to as simulation-based or proxy-based optimization, respectively.

2. Methodology

In this section, the optimization workflow (Figure 1) using both proxy-based and simulation-based predictions is presented. The main components of the workflow include the optimization implementation and the deep learning-based prediction model. Since simulated data is used for training the model and for validating the results, prediction with physics-based simulation models is also briefly described. We first describe the workflow that connects these components and then present a more detailed description of the three components in this section.

2.1. Overview of Workflow

The optimization goal is to maximize energy production by adjusting control variables, which are the mass flow rates of production and injection wells. The objective function is the net power generation over 12 years, which is maximized using a gradient-based optimization algorithm. The objective function and the required gradients are computed efficiently using the trained proxy model. The optimization results from

the deep learning-based proxy model are compared with those obtained using a physics-based simulation model of the geothermal reservoir. The simulation model is developed in TETRAD simulation software (Vinsome & Shook, 1993). As shown in Figure 1, the simulation model is used to (a) perform sensitivity analysis, (b) generate simulated datasets, and (c) validate the data-driven model and perform simulation-based optimization. Of these functionalities, the first two contribute to the development of the deep learning proxy model, which are denoted as “Process 1” and “Process 2” in Figure 1a, respectively. “Process 3” is called upon once the proxy model is built and used for prediction and optimization as shown Figure 1b.

The deep learning model used in this work is named CNN-RNN and was introduced in previous research (Jiang et al., 2022). The architecture of the proposed deep learning model consists of a convolutional neural network (CNN) and a recurrent neural network (RNN). The CNN-RNN model is designed to take the well mass flow rates as time-series input and predict the production enthalpy over the field life cycle (12 years). We also perform a sensitivity analysis using the simulation model to confirm the segmentation of the field into four independent divisions (fault zones). Field evidence and collected data, including tracer test results, have shown that the fault zones do not communicate with each other. The sensitivity analysis is used to investigate how strongly the inputs and outputs of all wells are correlated in the simulation model. The inputs and outputs are the production and injection mass flow rate and production enthalpy, respectively. Based on the sensitivity analysis, instead of using one proxy model to represent the whole field, we use four different models to learn the input-output relationship for each fault zone. This process serves as feature selection and reduces the complexity of the deep learning model and the amount of data required for training. Once the CNN-RNN models are trained and validated based on simulated data, they are used as proxy models to predict the energy production response of different fault zones in the reservoir. The predicted enthalpies and mass flow rates by the proxy model are then used to calculate the optimization objective function. In this paper, we refer to the individual models for each fault zone as the CNN-RNN model and denote the aggregate model for all zones as the proxy model for the field.

The simulation model is used to generate the simulated data sets for training and validation of the deep learning models for each fault zone. The control vectors are sampled using the Latin Hypercube Sampling (LHS) technique. In the second experiment of this work, we progressively increase the control samples to find the minimum sample size that provides the desired accuracy. The training is supervised and requires labels (simulated production enthalpy) for each input feature (the normalized mass flow rate). The trained models are validated by comparing their predictions against the validation data sets using the root-mean-square-error (RMSE) and R-squared (R^2) score metrics. Once the proxy model is validated, it is integrated into the proxy-based optimization framework for fast computation of the objective function and its gradient. The required gradients are automatically calculated and stored during forward prediction with the deep learning-based proxy model. To evaluate the performance of the proxy-based optimization, the physics-based simulation model is used in two different ways. The simulation model is first used to monitor the prediction accuracy of the proxy model. At each iteration of the proxy-based optimization, the simulation model takes the input control variables and returns the simulated value of the objective function (as the reference case). This monitoring process is not needed in real application of the method and is only used to illustrate the prediction fidelity of the proxy model. Second, for validating the optimization results from the proxy model, the simulation model is used to perform simulation-based optimization. The results from the simulation-based optimization are used as the reference case in evaluating the optimization performance of the proxy-based optimization.

2.2. Optimization Implementation

The proxy-based and simulation-based optimization frameworks are both implemented using a gradient-based local search algorithm. When the physics-based simulation is used to provide the predictions during optimization iterations, the computational cost can be high. The computational burden can increase significantly when the required gradient information must be computed numerically. One approach to speed up the computation is to adopt gradient approximation techniques such as the ensemble-based optimization (EnOpt). The EnOpt algorithm finds a least squares solution to the linear relationship between perturbed control vectors and corresponding objective values (Fonseca et al., 2015). It typically requires far fewer simulations than the standard finite difference approach to approximate the gradients. However, the quality of the approximated gradient in the EnOpt approach depends on the number of simulations used. Errors in approximating the gradients typically lead to additional optimization iterations to converge to a solution. In extreme cases, the approximation error may

become too large and the computed search direction may not be a descent direction, thus stalling the optimization algorithm. In this work, the simulation-based optimization is only used to validate the results of the proxy-based optimization. In the examples presented, since the number of decision variables is modest, finite-difference gradient approximation is used to compute the gradients. The implementation of the proxy-based and simulation-based optimization frameworks is based on the submodule “optimize” from the open-source SciPy software. For optimization, the built-in trust-region constrained optimization algorithm, with the implementation proposed in Lalee et al. (1998) for equality-constraint problems, is used.

2.2.1. Optimization Formulation

The well control variables of the optimization problem are time-varying mass/volumetric flow rate, or BHP. Denoting the control variables as u and the objective function as $f(\cdot)$, a simple form of the optimization problem can be formulated as:

$$\begin{aligned} u^* &= \arg \min_{u \in U} f(u) \\ \text{s.t. } g(u, m) &= 0 \end{aligned} \quad (1)$$

where $g(u, m)$ are the nonlinear equality constraints that represent the governing equations of the subsurface flow system (e.g., balance equations of mass, momentum, and energy). The notation m is used to represent the reservoir input parameters (e.g., reservoir and fluid properties). Since the governing equations constraints are automatically satisfied by running the fluid flow simulation, hereafter, this constraint is dropped for brevity.

In optimization of geothermal reservoirs, different objective functions may be used. Examples include maximization of net power generation, minimization of maximum temperature decline, and minimization of average temperature decline over the life cycle of the project. In this work, maximization of the net power generation is adopted to improve energy production sustainability by mitigating the long-term temperature decline. To convert the problem to minimization, the objective function is set to be the negative value of the net power generation from the reservoir, that is:

$$\begin{aligned} f(u) &= -\text{Net Power Generation} \\ &= -\sum_{k=1}^{N_T} \left[\left(\sum_{j=1}^{N_P} u_j^k h_j^k - \sum_{i=1}^{N_I} u_i^k h_i^k \right) \cdot \eta - \sum_{i=1}^{N_P+N_I} u_i^k r_i^k \right] \cdot \Delta t_k \end{aligned} \quad (2)$$

where u is the vector of input variable that consists of the mass flow rate of production and injection wells at each time step; u_i^k and h_i^k are the mass flow rate and specific flowing enthalpy of well i at time step k , respectively; η denotes the power plant efficiency; r_i^k represents the ratio of the pump load of well i to its mass flow rate at time step k ; Δt_k is the k^{th} time step size; N_T , N_P and N_I denote the number of time steps, production wells, and injection wells, respectively. The objective function consists of three components for each time step k , which are the produced enthalpy $\sum_{j=1}^{N_P} u_j^k h_j^k$, the injected enthalpy $\sum_{i=1}^{N_I} u_i^k h_i^k$, and the cost of running the pumps for the production and injection wells $\sum_{i=1}^{N_P+N_I} u_i^k r_i^k$. To simplify the optimization problem, the pump load ratio r is set to be constant. The mass flow rate u , as control vector, is set to honor three practical constraints: (a) mass balance equality constraint (equating the total rate of production and injection for each time step), (b) demand equality constraint, by setting the total production rate to be a fixed value, and (c) bound constraints for each control variable. The mathematical representation of the resulting constrained optimization problem is given as:

$$\begin{aligned} \min_u f(u) &= -\text{Net Power Generation} \\ \text{s.t.} & \\ \sum_i^{N_P} u_i^k &= \sum_j^{N_I} u_j^k, \text{ for } k = 1, 2, \dots, N_T \\ \sum_i^{N_P} u_i^k &= Q_{\text{prod}}^k, \text{ for } k = 1, 2, \dots, N_T \\ u^{\text{lower}} &\leq u \leq u^{\text{upper}} \end{aligned} \quad (3)$$

where Q_{prod}^k is the total production rate at time step k ; u^{lower} and u^{upper} are the lower and upper bounds of the control variable u , respectively.

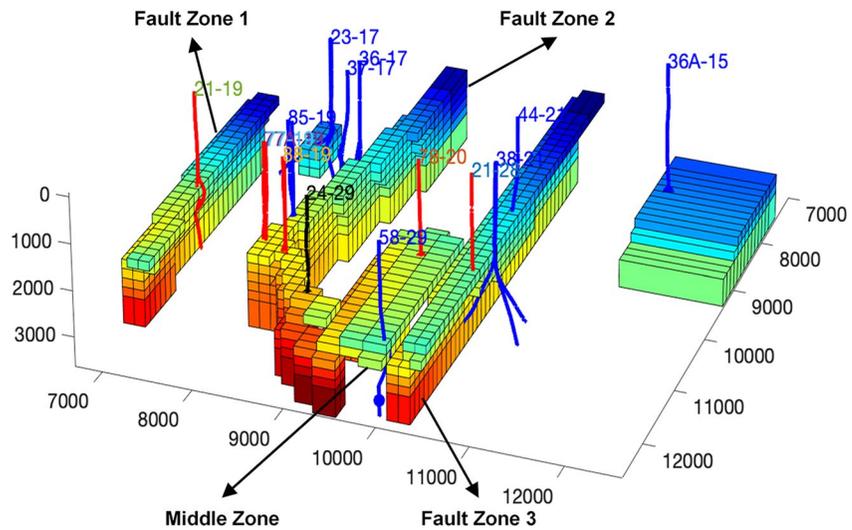


Figure 2. The main fault zones and well locations in the physics-based simulation model of the field example. The color of the grid block represents the initial temperature of the rock matrix. The scale of temperature is not shown due to data confidentiality.

2.3. Physics-Based Simulation Model

For numerical simulation of the geothermal reservoir, in this paper we use TETRAD (Vinsome & Shook, 1993). The simulation model is a three-dimensional single-phase dual-porosity numerical reservoir model consisting of roughly 35,000 grid blocks. The numerical model has dimensions of 16 km in the *X*-direction by 18 km in the *Y*-direction and extends from +1,500 to −3,000 mRSL. The geothermal reservoir has several distinct fault zones that are characterized by elevated permeability, high-productivity wells, and convective thermal gradients (Murphy et al., 2017). The subsurface system contains six production wells and eight injection wells, for which the connectivity is dominated by the existing fault zones (Cladouhos et al., 2017). A screenshot of the initial temperature of the main fault zones, as well as the well locations, in the numerical simulation model is shown in Figure 2. The six production wells are marked in red, while the eight injection wells are shown in blue. Well 24–29 was originally an observation well and changed to an active injection well in the simulation. The main fault zones are named Fault Zone 1, Fault Zone 2, Middle Zone, and Fault Zone 3, from West to East, respectively. Table 1 summarizes the wells included in each of the fault zones. Fault Zone 1 and Middle Zone have only one production well each, that is, Well 21-19 and Well 78-20, respectively. Well 58-29 is not active in this work, despite being marked as an injection well. Fault Zone 2 has Production Wells 88-19, 77-19 and 77A-19, and Injection Wells 85-19, 36-17, 37-17, 23-17, and 24-29. Well 88-19 has the strongest connection to the production wells in this compartment due to its proximity. Fault Zone 3 has Production Well 21-28 and Injection Wells 44-21 and 38-21. The differences in the connectivity and the complex geological setting result in distinct production behavior for each well.

2.3.1. Simulated Data Set

Weekly outputs from the simulation model are used as data set to train the deep learning model for long-term prediction. The weekly data set is generated over around 12 years (from 2021/05/21 to 2032/12/31). The well controls (features) are the mass flow rates of production and injection wells. The model output (label) is the specific enthalpy of produced brine. To generate the training data set, the mass flow rates are generated stochastically, while honoring the three linear constraints in Equation 3. The input variables are then randomly perturbed by adding Gaussian noise with a 2% standard deviation. Injection temperature is not controlled but is a required input into the simulation and is specified based on realistic data that follows seasonal variations.

Table 1
Separation of Wells for Each Fault Zone

Fault zone	Production wells	Injection wells
Fault Zone 1	21-19	–
Fault Zone 2	88-19, 77-19, 77A-19	85-19, 36-17, 37-17, 23-17, 24,29I
Middle Zone	78-20	–
Fault Zone 3	21-28	44-21, 38-21
Others	–	36A-15

2.3.2. Simulation Model With Annual Control Changes

The original simulation model that is used to generate weekly training data is adjusted and used with less frequent outputs for faster implementation of the long-term optimization. As stated earlier, the optimization problem with simulated prediction generated by the simulation model is used to evaluate the optimization performance of the data-driven model.

The simulation model is also applied to monitor the prediction accuracy of the proxy model during the optimization iterations. To this end, the control variables over all iterations within the proxy-based optimization are sent to the simulation model to compute the corresponding simulated values of the objective function as a reference. The comparison between the values predicted by the proxy and simulated models is used to monitor the quality of the proxy model. This monitoring step is not needed in actual implementation of the method and is only used in this paper to demonstrate the prediction performance of the proxy model throughout the iterations.

2.3.3. Sensitivity Analysis and Feature Selection

A sensitivity analysis is performed with the simulation model to assess the relationship between the inputs and outputs of the geothermal reservoir model. The analysis leads to a better understanding of the subsurface structure and informs the feature selection step of the data-driven model. The sensitivity analysis quantifies the partial derivative $\partial y/\partial x$, which measures the changes in the output variable y due to changes in the input variable x (Lenhart et al., 2002). The central difference approximation of the partial derivative can be expressed as:

$$I' = \frac{y_2 - y_1}{2\Delta x} \quad (4)$$

where I' represents the sensitivity index, y_1 and y_2 refer to the output values corresponding to the two perturbed input variables $x_0 - \Delta x$ and $x_0 + \Delta x$, respectively, with x_0 denoting a reference value for the input with a corresponding output value y_0 , and Δx denoting the perturbation. To get a dimensionless sensitivity index, I' is normalized as follows

$$I' = \frac{(y_2 - y_1)/y_0}{2\Delta x/x_0} \quad (5)$$

In our examples, the output variables y are the specific enthalpy of six production wells collected after 10 years of operation. The input variables x are the mass flow rate of six production wells and eight injection wells, which are fixed during the simulation. For each input x_0 , two simulation runs are needed for the corresponding two perturbed controls $x_0 - \Delta x$ and $x_0 + \Delta x$, respectively. Therefore, the sensitivity analysis for the 14 variables involves 28 simulation runs. The perturbation size Δx is set to be 25% of the feasible ranges of x . Based on the results from sensitivity analysis, the simulation model of the geothermal reservoir is divided into four fault zones. We develop and train four separate deep learning models to represent each of the four fault zones in the numerical simulation model (Figure 2). The feature selection for deep learning models follows the separation presented in Table 1. Since Injection Well 36A-15 shows no effect on any of production wells, it is not included in any of the proxy models. The sensitivity analysis is used to reduce the dimension of the input and output variables in developing deep learning models. In machine learning, this process is known as feature selection, which aims to reduce the model complexity (e.g., the number of trainable parameters) and data needs.

2.4. Deep Learning-Base Proxy Model

In this section, we briefly discuss the role of the deep learning model (Figure 3) and the proxy model (Figure 4) to give a clear problem formulation. A detailed description and development of the CNN-RNN architecture can be found in our previous work (Jiang et al., 2022). The use of CNN-RNN architecture in this work is primarily motivated by easy access to this model from our prior work. Other implementations of RNN may also be used in the presented workflow. The CNN-RNN model consists of one CNN encoder, one RNN encoder, and one RNN decoder. The CNN encoder is based on the one-dimension (1D) convolution operation for time-series data. The CNN encoder takes a short sequence of historical data, $\{\mathbf{x}, \mathbf{y}\} \in \mathbb{R}^{D \times m}$, and maps it to the low-dimensional latent variables, $\mathbf{z} \in \mathbb{R}^{D_z \times 1}$. For each prediction step, starting from timestep t , the data consists of the historical well control trajectories (i.e., mass flow rate), $\mathbf{x} \in \mathbb{R}^{D_x \times m}$, and the corresponding enthalpy outputs, $\mathbf{y} \in \mathbb{R}^{D_y \times m}$, over the past m timesteps. The superscripts D , D_x , D_y , and D_z refer to the dimensions of historical data $\{\mathbf{x}, \mathbf{y}\}$, control

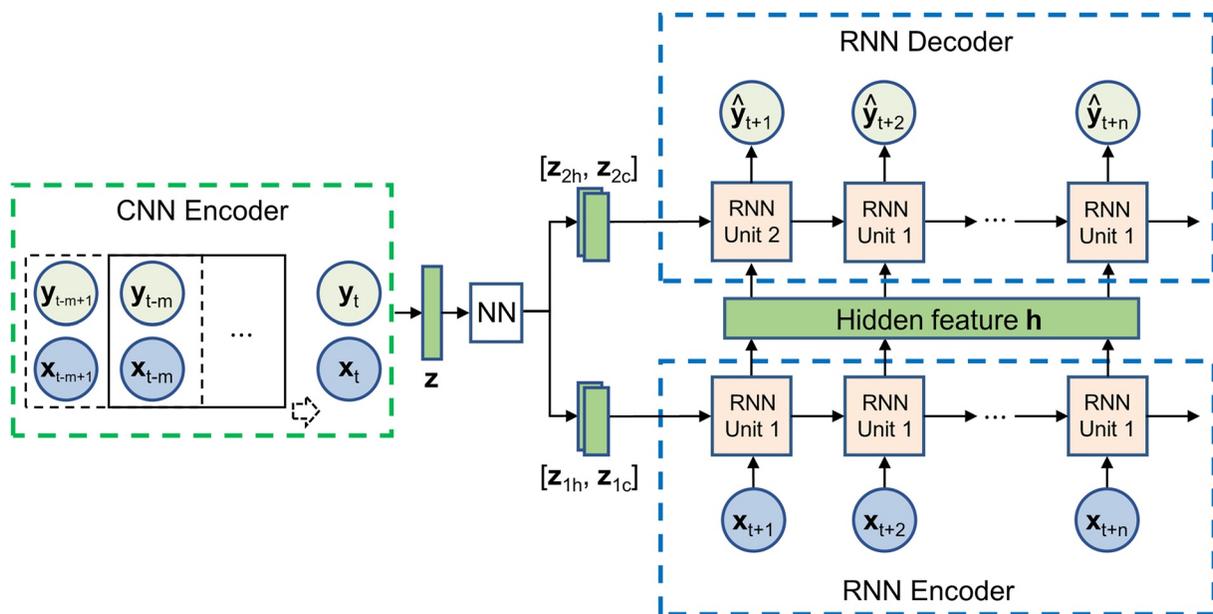


Figure 3. The architecture of the convolutional neural network-recurrent neural network (CNN-RNN) model.

variables x , enthalpy outputs y , and the latent variables z , respectively. The superscript m represents the length of historical data that is used in the CNN encoder. The output from the CNN encoder, z , is sent to four different fully connected feedforward neural networks that generate the initial states for the following RNN encoder and decoder structures.

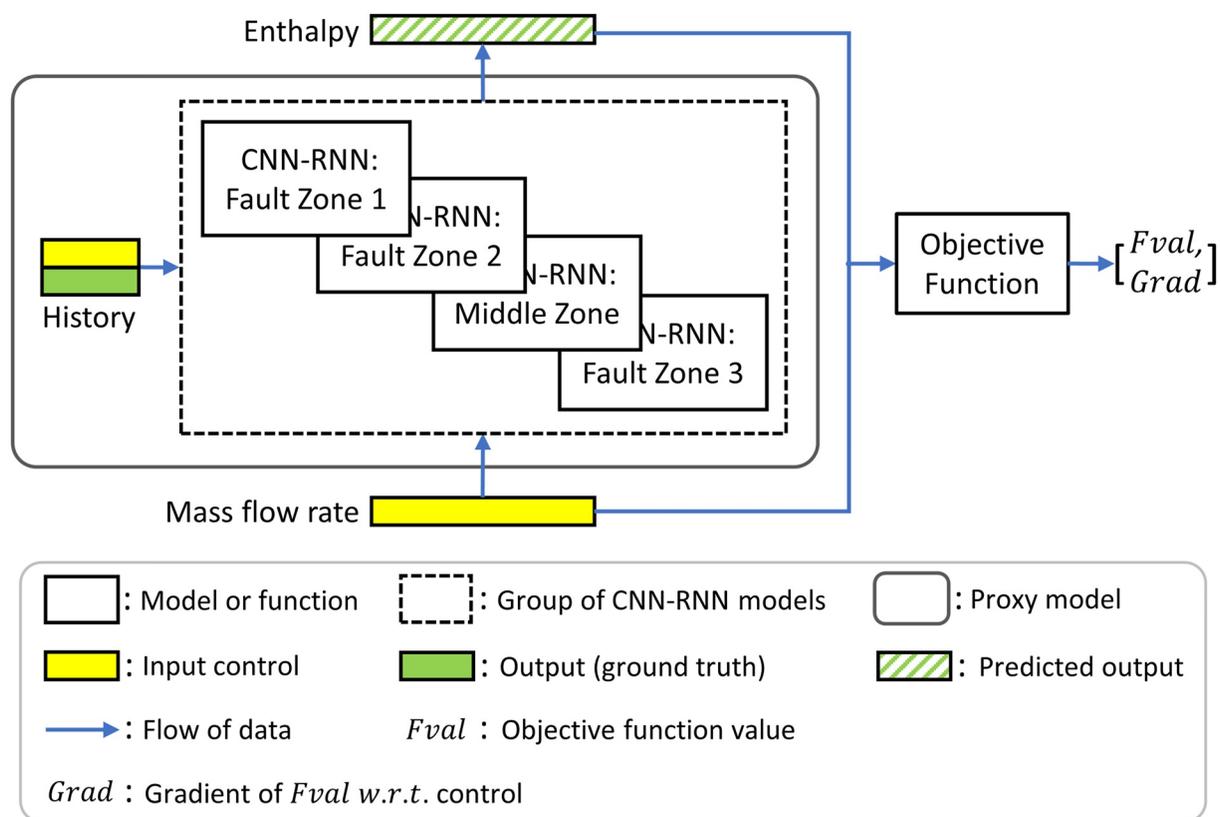


Figure 4. A demonstration of the proxy model and its use in calculating the objective function and its gradient.

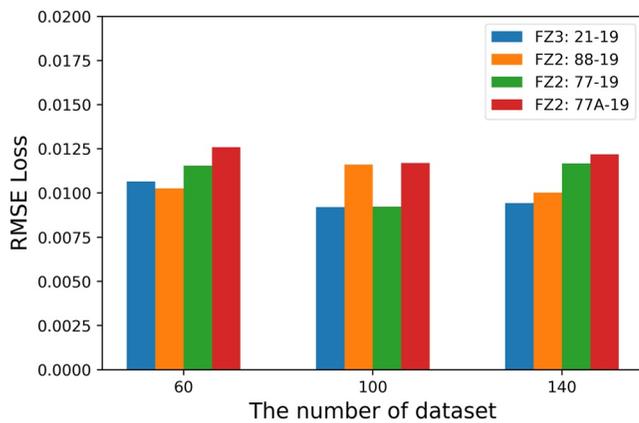


Figure 5. Sensitivity analysis showing the root-mean-square-error loss based on test data sets as a function of the number of training datasets.

An LSTM unit is used in both encoder and decoder. As a popular variant of RNN, the LSTM is developed for handling the long-term dependencies between the current states and previous states. For the RNN encoder, we use two different neural networks to generate the initial hidden state, $\mathbf{z}_{1h} \in \mathbb{R}^{D_h}$, and the initial cell state, $\mathbf{z}_{1c} \in \mathbb{R}^{D_h}$. The subscript 1 denotes the first RNN encoder layer. The notations h and c represent the hidden and cell states, respectively. The superscript D_h denotes the dimension of hidden state. For the RNN decoder, another set of two neural networks is used to generate the hidden state, $\mathbf{z}_{2h} \in \mathbb{R}^{D_y}$, and the cell state, $\mathbf{z}_{2c} \in \mathbb{R}^{D_y}$. The RNN encoder takes an arbitrary length n of future controls, $\mathbf{x} \in \mathbb{R}^{D_x \times n}$, and compresses them into a hidden feature sequence, $\mathbf{h} \in \mathbb{R}^{D_h \times n}$. Then, the RNN decoder takes the hidden state as input and generates the predicted output $\hat{\mathbf{y}} \in \mathbb{R}^{D_y \times n}$ as time-series data.

Since the input x to the CNN-RNN model are the well mass flow rates and the outputs y are the specific enthalpies of the produced brine the values of D_x and D_y are the total number of wells and the number of production wells within a fault zone, respectively. The data are normalized using

the 0-1 min-max normalization. The length of the historical data m and the dimension of the hidden feature D_h are the tuning parameters that must be specified by the user. The length of the future controls, n , is a user-specified integer. The CNN-RNN implementation is based on the open-source machine learning package TensorFlow (Abadi et al., 2016). The training process is implemented by minimizing an objective function that represents the mismatch between predicted and observed data. The Adam optimizer is used to train the resulting architectures (Kingma & Ba, 2014).

The four trained CNN-RNN models for different fault zones are combined and used as a proxy model to predict the field energy production performance. For the trained proxy model, the input is the sequence of future controls (i.e., mass flow rates) of all production and injection wells $\mathbf{x} \in \mathbb{R}^{D_x \times n}$ and the outputs are the specific enthalpies of the production wells, $\hat{\mathbf{y}} \in \mathbb{R}^{D_y \times n}$. The predicted enthalpies, along with the mass flow rates, are used to calculate the net power generation. As shown in Figure 4, the proxy model is then connected to the objective-function layer through a computational graph in TensorFlow, which leverages the automatic differentiation (AD) technique for the gradient calculation. During optimization, the gradient will be automatically calculated and stored when the whole computational graph (i.e., the proxy model combined with the objective function) is used to predict the enthalpy and calculate the objective function.

3. Field-Scale Numerical Experiments

In this section, two numerical experiments are presented to demonstrate the performance of the proposed proxy-based optimization framework. The deep-learning proxy models in the two experiments are trained and validated using the simulated data set generated by the field model. The proxy-based prediction and optimization results are validated by replacing the proxy model with the simulation model for the same problem setup.

3.1. Experiment 1: Fixed Rate Optimization

In the first experiment, the control vector (mass flow rate) is fixed over the life cycle of the project, which is around 12 years (from 2021/05/21 to 2032/12/31). There are 14 control variables corresponding to the 14 wells in the reservoir. To investigate the sensitivity of the prediction performance to the number of training datasets, three proxy models are trained using datasets with 60, 100, and 140 simulated cases, and the results are validated on an additional set of 100 realizations. The enthalpy responses of the four different fault zones to changes in the control variables are distinct. Compared with the insignificant enthalpy variations in Fault Zone 1 and the Middle Zone, the enthalpy outputs from Fault Zones 2 and 3 are sensitive to the injection rates, mainly due to the proximity of the wells. Figure 5 shows the RMSE loss of the three sets of proxy models (trained on 60, 100, and 140 data sets) for the validation set with the 100 unseen samples. The RMSE losses are calculated for the predicted normalized enthalpy of four production wells from Fault Zones 2 (88-19, 77-19, and 77A-19) and Fault Zone 3 (21-28). All three proxy models show small losses. A non-monotonic behavior is observed in the RMSE

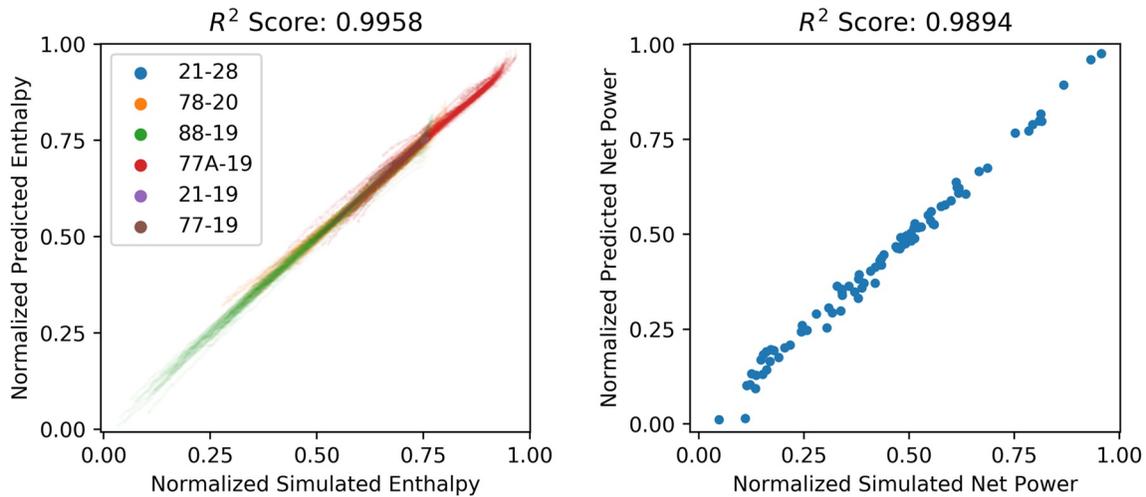


Figure 6. Scatter plots of predictions from the simulation model and the proxy model trained on 60 data sets.

losses for different sizes of training data from 60 to 140. For example, the RMSE loss for Well 21-19 decreases (slightly) when the data size is increased from 60 to 100 and increases (slightly) when the data size increases from 100 to 140. This non-monotonic behavior is not significant and is mainly attributed to the randomness in the training of a deep learning model (e.g., initialization of weights, local minima, stochastic gradient algorithms, etc.). Therefore, for the sample sizes used in the first experiment (60, 100, and 140), the prediction performance of the proxy model is acceptable.

Figure 6 shows the Quantile-Quantile (Q-Q) plot to compare the predictions from the proxy model (y-axis) with the corresponding simulated values (x-axis) over the validation data set. Specifically, we investigate the normalized values of the specific enthalpy of production wells and the net power generation of the project. The proxy model used for comparison is trained using 60 samples. As is reflected by the R^2 scores that are close to 1, the predictions of enthalpy and net power generation match the simulated values.

Each of the three proxy models is used in the optimization framework, with the results denoted as Proxy 60, Proxy 100, and Proxy 140. As an additional demonstration step, we run the simulation model to monitor the prediction accuracy of the proxy model during optimization (note that the optimization algorithm is implemented based on the proxy model and the physics-based monitoring simulations are performed to validate the predictions made by the proxy model). The corresponding monitoring cases are referred to as Sim 60, Sim 100, and Sim 140, respectively. Although this comparison is not necessary in practical application of the proposed approach, it is performed in this paper to examine the performance of the model. As shown in Figure 7(left), the objective function in all three proxy-based cases shows approximately 9% improvement. The monitoring plots, however,

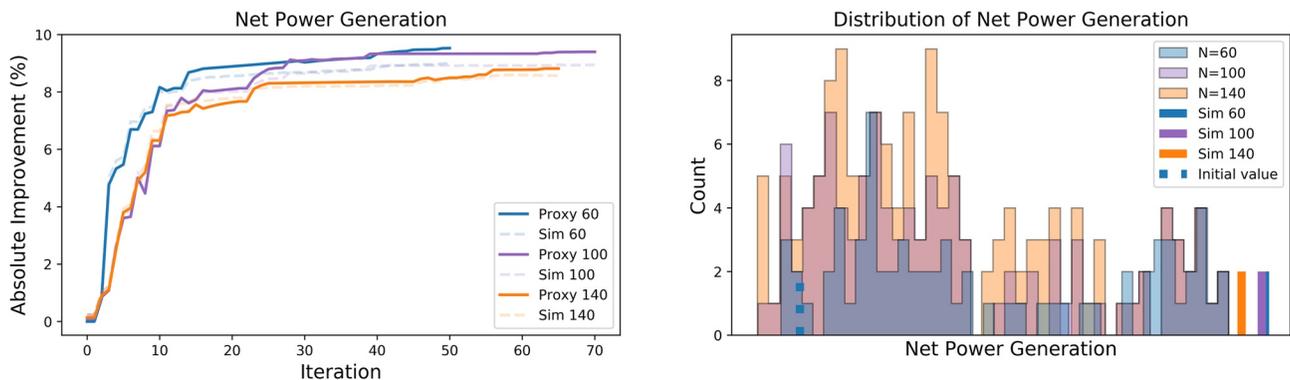


Figure 7. Objective function value for proxy-based optimization: (left) evolution and monitoring values versus iterations, (right) the distribution of net power generation for the training data set and monitored values.

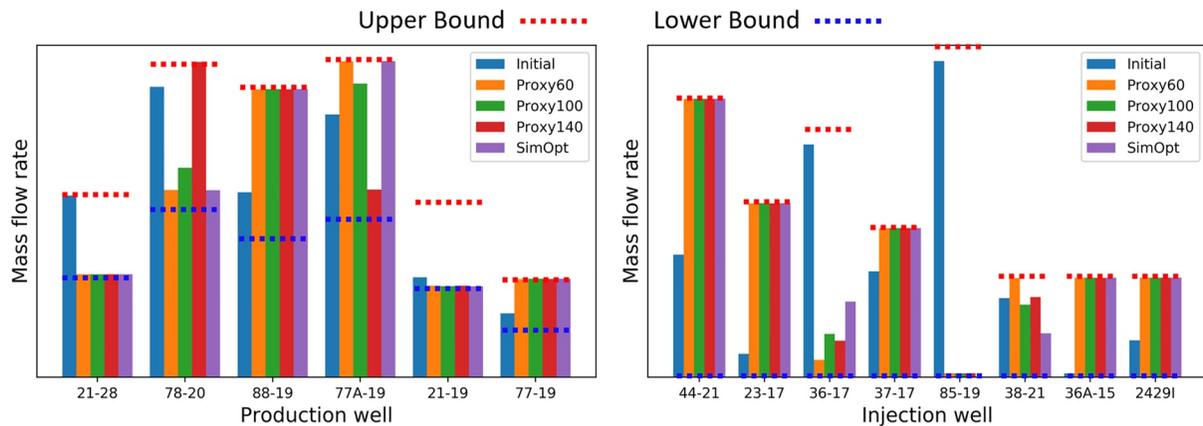


Figure 8. Comparison of optimal production and injection controls for proxy-based and simulation-based optimization.

show slightly lower improvements at the end of the iterations, indicating that the proxy-based optimizations overestimate the optimal net power generation. The discrepancy between the proxy and simulation models during the iterations suggests that the control inputs begin to fall outside the regions covered by the training datasets. Figure 7(right) depicts the distribution of the net power generation for the training datasets, as well as the distribution of the monitored objective function values for three proxy-based optimization cases (i.e., Sim 60, Sim 100, and Sim 140). All three monitored values, with the prefix “Sim” in the legend, surpass the net power generation in the training data set. The results indicate that the proxy model can identify search directions that improve the value of the objective function.

To further investigate the reliability of the proxy-based optimization framework, simulation-based optimization is implemented for the same setup. Figure 8 shows the comparison between optimal controls of each production and injection well for three proxy-based optimization cases along with the simulation-based case. The results confirm an overall good consistency between the well control solutions that are obtained using the simulation and proxy models. The production wells with higher production capacity are assigned higher rates (e.g., Wells 88-19 and 77-19), and vice versa (e.g., Wells 21-28 and 21-19). The allocation of injection rate is mainly based on the mass-balance constraint. The Injection Well 85-19 has an obvious negative effect on the production in Fault Zone 2 and, therefore, its injection rates is decreased to the lower bound.

Table 2 summarizes the improvements in the net power generation as evaluated by the proxy-based optimization, the monitoring cases, and the simulation-based optimization. The absolute improvement achieved by simulation-based optimization is reasonably close to the monitoring values for proxy-based cases. However, the simulation-based optimization requires 24 iterations to reach the optimal control solution, with a total of 360 simulation runs to complete the optimization problem (using finite-different gradient calculation). In contrast, proxy-based optimization requires fewer simulations during the training phase (60, 100, and 140 simulation runs). In the next experiment, we further decrease the number of simulation runs that are required for training a proxy model and investigate the efficiency of the proxy-based optimization.

3.2. Experiment 2: Dynamic Rate Control Optimization

In the second experiment, a more complex optimization problem is performed, where the control variables are allowed to change annually over the 12 years of simulation. Therefore, the length of the control vector is $12 \times 14 = 168$ (12 control steps and 14 wells). To investigate the accuracy and computational efficiency of the proxy model, we progressively add control samples based on the Latin Hypercubic Sampling (LHS) and generate the corresponding simulated samples for a set of given controls. The sampling strategy is designed based on the approach in (Sheikholeslami & Razavi, 2017). Then, the procedure of developing the proxy model (Figure 9) starts with N_1 samples as training set and N_2 samples as validation set, where

Table 2			
Summary of Absolute Improvement in Net Power Generation in Percentage (%)			
The number of data sets	$N = 60$	$N = 100$	$N = 140$
Proxy-based optimization	9.52	9.39	8.81
Monitoring using simulation	8.99	8.94	8.55
Simulation-based optimization	8.69 (24 iterations)		

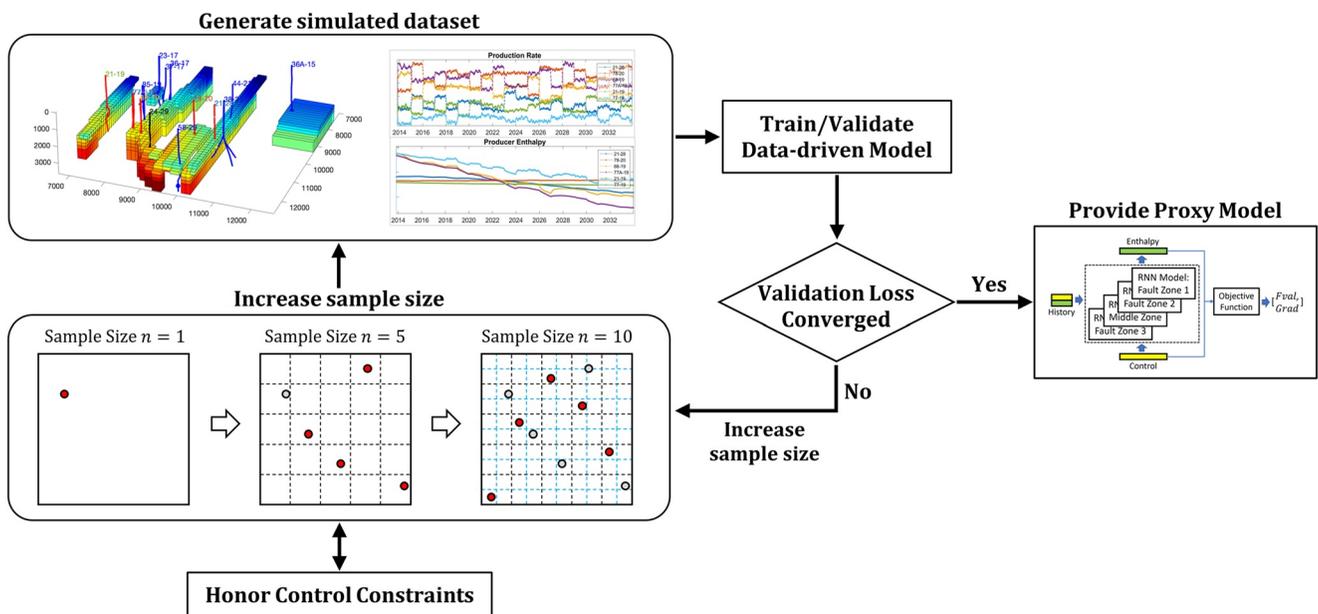


Figure 9. A progressive sampling procedure for the development of the data-driven proxy model. The sampling strategy gradually increases the sample size by progressive Latin Hypercube Sampling and is terminated by a converged validation loss and acceptable R^2 scores.

$(N_1 + N_2)$ control vectors are sampled using the LHS approach. The CNN-RNN models are trained and validated with the validation losses saved. Next, N_3 validation samples are generated using the new control vectors that are sampled progressively using LHS. The controls for the validation samples are generated to randomly fill in the space between previously sampled $(N_1 + N_2)$ control vectors for training. The new validation loss on N_3 samples for CNN-RNN models are saved and compared with the training loss to ensure consistency. This procedure is repeated until the validation loss converges and the R^2 score gets close to 1. The resulting deep learning models are used for optimization. The selection of the set of sample sizes $\{N_1, N_2, \dots, N_k\}$ in this procedure can be defined by the user. In this work, CNN-RNN models were built with 10 training samples and 10 validation samples. Also, a sensitivity analysis was performed in (Jiang et al., 2022) to assess the impact of the training data size on the prediction performance of the deep learning models. Figure 10 shows an example of predicting the specific enthalpy with the CNN-RNN models trained only on 10 samples and validated on another 10 samples. The enthalpy predictions from the CNN-RNN models are compared with those obtained from the simulation models. It is observed that the changes in the enthalpy for the six production wells are different. Optimization is then used to adjust the well controls to improve the energy production performance and to avoid early thermal breakthrough.

In this example, the proxy model is built using four CNN-RNN models that are trained on 10 simulated cases within the period 2021/05/21 and 2032/12/31. To ensure that the proxy model follows the historical data before

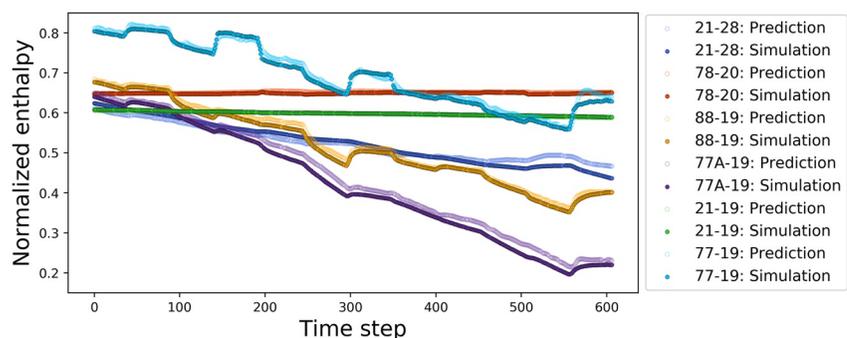


Figure 10. An example of the long-term prediction of specific enthalpy using CNN-RNN models trained on 10 samples.

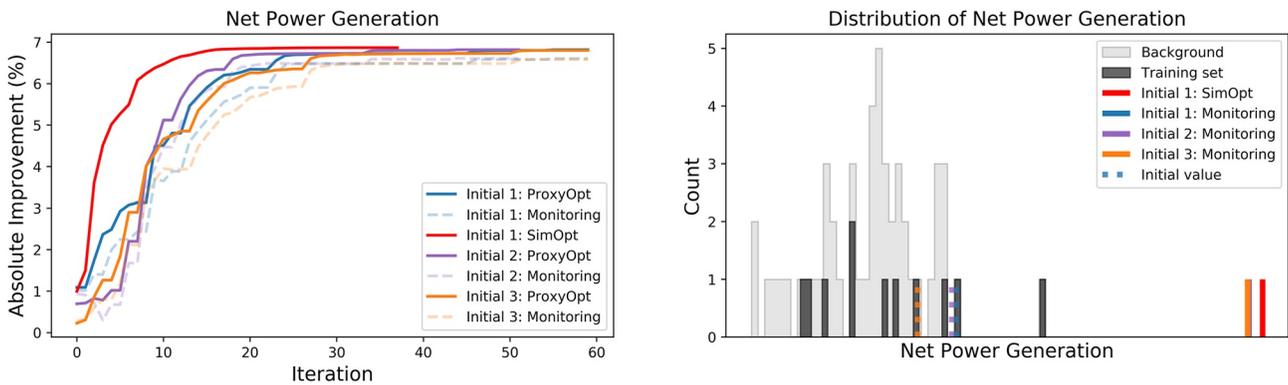


Figure 11. Objective function value for proxy-based and simulation-based optimization: (left) evolution and monitoring values versus iteration number, (right) distribution of net power generation for the training data set and monitored optimal values.

2021/05/21, each simulated sample includes the historical data ranging from 2013/10/17 to 2021/05/21. The optimization is initialized using three separate vectors of initial controls \mathbf{u} , which include the field control vector (Initial Control 1) and two random control vectors, denoted as Initial Control 2 and 3. The Initial Control 1 is the control from historical field data at the last timestep 2021/05/21. One of the two random control vectors (Initial Control 2) is not seen by the proxy model, while the other (Initial Control 3) is picked from the training samples. For validation, simulation-based optimization with the same setup is also initialized using the field control vector (Initial Control 1).

Figure 11(left) plots the objective function values versus the iteration number for the proxy-based (“ProxyOpt”) and simulation-based (“SimOpt”) optimization. The monitoring cases (“Monitoring”) show that the three initial control cases tend to have a slight discrepancy during the iterations. This behavior is in part due to a larger number of control variables (degrees of freedom) and complexity for of the CNN-RNN model. The SimOpt result also outperforms the ProxyOpt model for the Initial Control 1 by showing a faster improvement within fewer iterations. However, the simulation-based optimization using finite-difference gradient requires 169 simulation runs to calculate the gradient, making the approach computationally expensive for large-scale models. A detailed discussion of the computational cost is presented in Section 4.3. In the first few iterations, an obvious discrepancy exists in the Initial Control 2 since the control values are not seen by the proxy model. In Figure 11(right), the net power generation labels in the training set are compared with those obtained after optimization. The net power generation outputs in the validation and testing sets are also shown in the background. The results show that the proxy-based optimized values exceed the range of values that are present in the training data, suggesting that the proxy model can extrapolate beyond the training set. The result is promising since the RNN proxy is trained using data from only 10 simulation cases.

Figure 12 shows that the production enthalpy of Fault Zone 2 (88-19, 77-19, and 77A-19) is increased throughout the iterations and is higher than those in the other zones. Figure 13 shows the normalized rate controls of all production wells for proxy-based and simulation-based optimization, including the initial and optimal solutions as well as the control vectors in the training set. The enthalpy from Fault Zone 1 (21-19) is the lowest, and the related control reaches its lower bound over the 12 steps for all the three cases with different initial controls. For Fault Zone 2, however, the control rates of the three production wells with higher capacity reach their upper bounds. Figures 14 and 15 show the normalized rate of injection wells in Fault Zone 2 and 3, respectively. The Injection Well 85-19, on the other hand, is closed by the optimization, as it shows a strong connection to Wells 88-19, 77-19, and 77A-19 and can cause early thermal breakthrough for continuous injection. The shutdown of Injection Well 85-19 is also the reason for the increase in the production enthalpy from Fault Zone 2. For Middle Zone and Fault Zone 3, the controls for Wells 78-20 and 21-28 are decreased to their lower bounds. For both ProxyOpt and SimOpt, a portion of the rate allocation is switched from Well 21-28 to Well 78-20, as the specific enthalpy from Well 78-20 surpasses that of Well 21-28 after around 6 years of operation (Figure 12). For the SimOpt, the timing of the switch in control solution fits the “switch” that happens in the production enthalpy, which is around the sixth step. However, the control solution with the ProxyOpt provides an earlier switch between these two wells, which is the reason for the objective function being different in SimOpt versus ProxyOpt (Figure 11). The reason for an early switch for ProxyOpt might be the mismatch between the

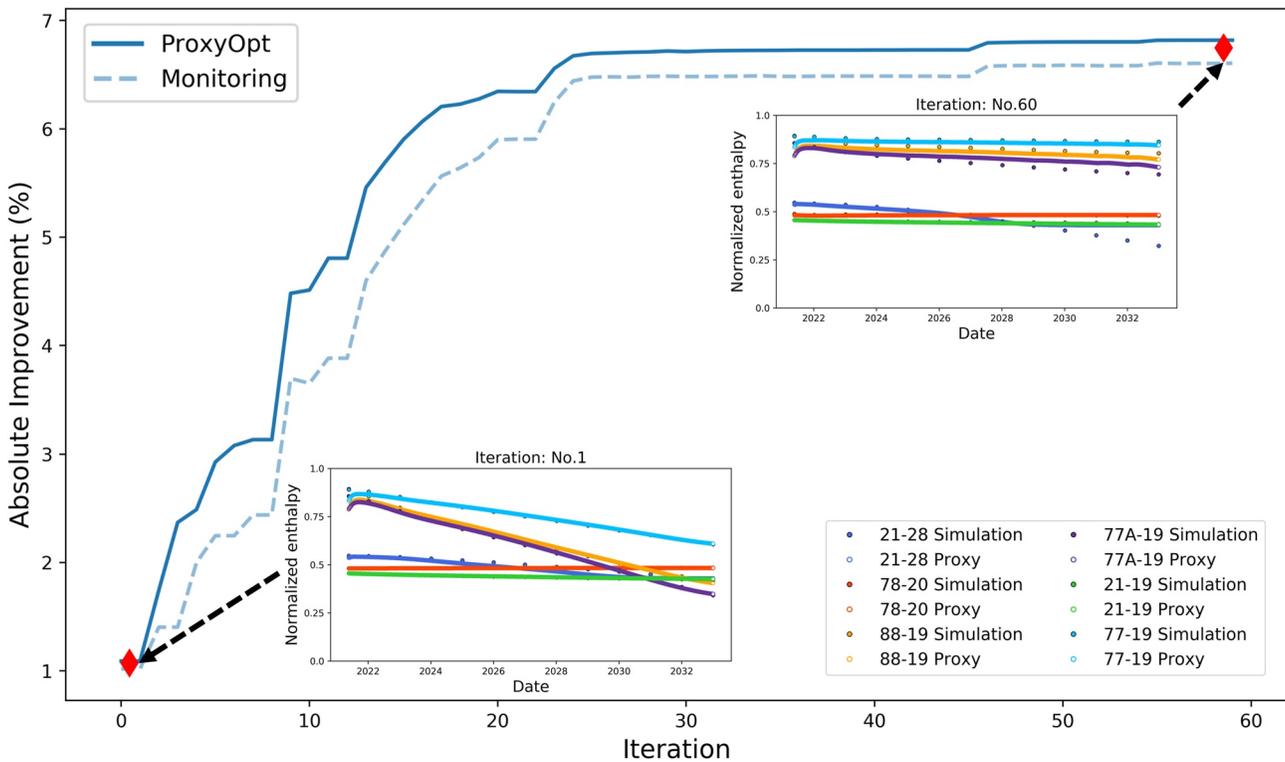


Figure 12. Specific enthalpy from the proxy model (solid lines) and monitored values (dots) shown at the first and last iterations of the optimization case with “Initial Control 1”.

predicted enthalpy by RNN and the simulated enthalpy at the last iteration (Figure 12). For wells that show high sensitivity of enthalpy to control inputs (e.g., Wells 85-19, 88-19, 77-19, 77A-19), the control solutions from simulation-based optimization are consistent with those from the proxy-based models. However, the optimal injection rates show less consistency and are more affected by constraints since the specific enthalpy is relatively insensitive to most of them.

4. Discussion

This study presents an optimization framework for geothermal reservoir operations and management by using deep learning-based predictive models. The proxy model is based on encoder-decoder CNN-RNN architectures that are trained using time series measurements collected at the well locations. The trained model is then used to predict the production enthalpy to calculate the net power generation. Sensitivity analysis is used to perform feature selection and to choose the major input-output variables of the deep learning model. The paper presents the design, construction, training, and validation of the proxy-based optimization framework to improve the efficiency of energy production from geothermal reservoirs. The feature selection step is used to generate four independent CNN-RNN models, one per each fault zone in the reservoir, to predict the specific enthalpy for the corresponding fault zones. The proxy-based optimization is applied to two field-scale numerical experiments to maximize the net power generation by adjusting the mass flow rate of production and injection wells.

4.1. Feature Learning

The results from the proxy-based optimization are validated against those obtained from a physics-based simulation model. The prediction accuracy and optimization performance of the proxy model are evaluated, respectively, by monitoring the prediction errors and comparing the results with those from a simulation-based optimization.

As demonstrated in the two examples, throughout the iterations, Figures 7(left) and Figure 11(left), the objective function values obtained by the proxy-based optimization are close to those predicted by the physics-based

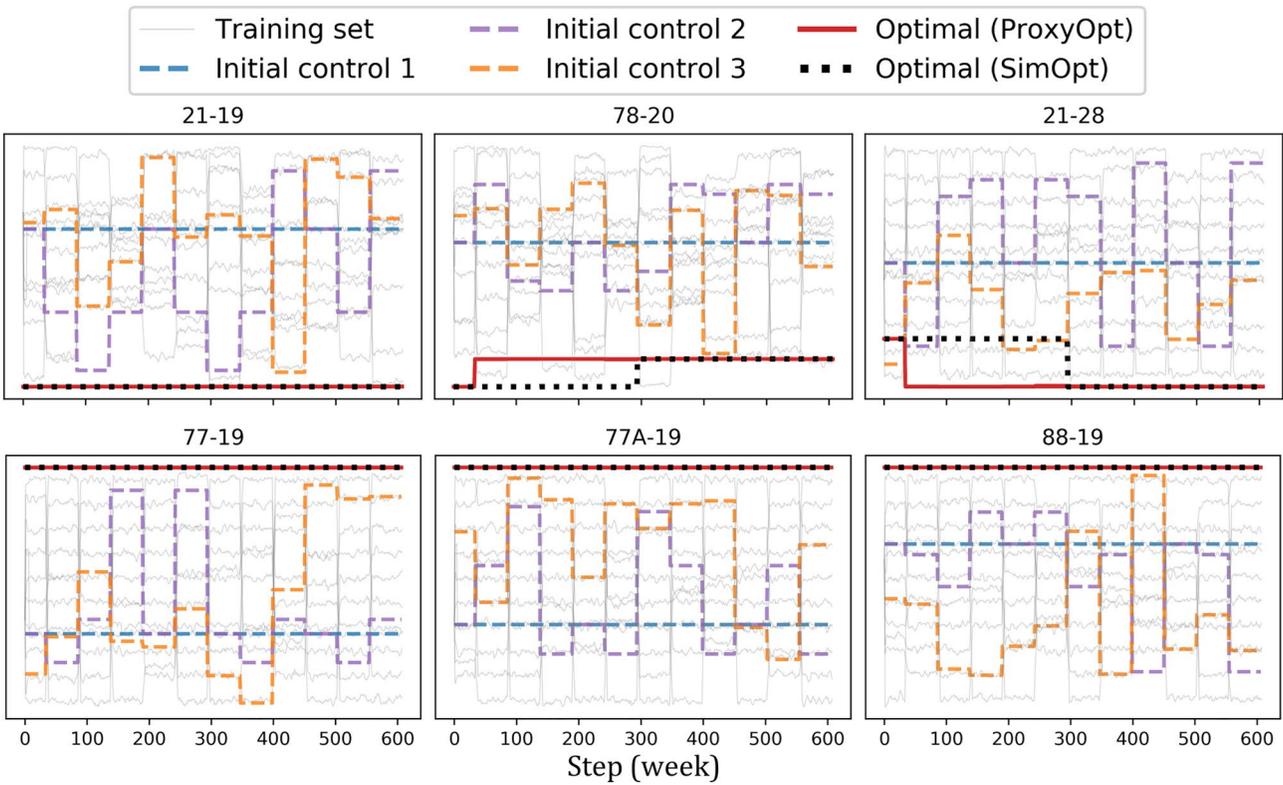


Figure 13. Normalized mass flow rates of all production wells. Gray lines are the rate distribution in the training set; colored dash lines are three initial controls, and the red line is the optimal controls for the three optimization cases. The black dotted line shows the optimal control solution for the simulation-based optimization.

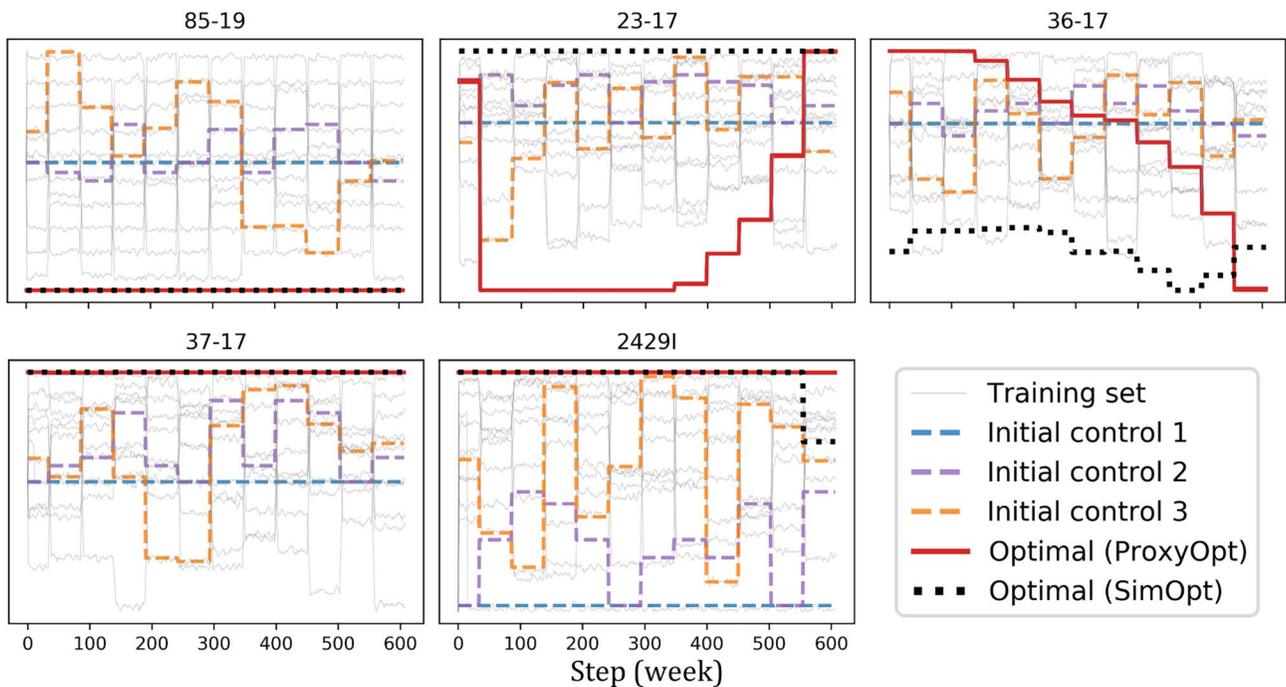


Figure 14. Normalized mass flow rates of the injection wells in Fault Zone 2.

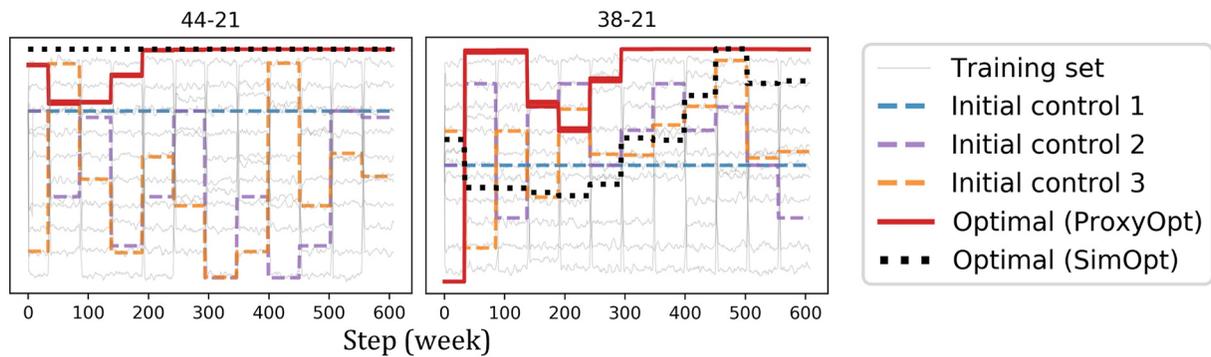


Figure 15. Normalized mass flow rates of the injection wells in Fault Zone 3.

simulation model. This indicates that, once trained, the CNN-RNN model can provide accurate predictions during optimization. Furthermore, the CNN-RNN models in the second example are trained by only using simulated data from 10 simulated samples. With modest computational and data requirements for training, the proxy model can improve the computational efficiency of the field-scale optimization problems without a significant loss in accuracy. The simulation-based optimization approach is implemented to provide a reference case to compare the final values of the objective functions and the optimization solutions. As shown in Table 2 and Figure 11(left), the improvements in net power generation that are achieved by the proxy-based optimization are as high as those obtained by the simulation-based approach. Another important observation is the extrapolation power of the proxy model. As shown in Figures 7(right) and Figure 11(right), the proxy models in the two examples can successfully extrapolate the net power generation for control strategies that fall beyond those used for training. This suggests that the CNN-RNN model can learn the input-output relationship after being trained on a small number of simulated datasets. However, this extrapolation property depends on the distance from the training data set and may not generalize in other examples.

4.2. Retraining Process

An important observation from this study is that the optimization process is likely to expose the proxy model to extrapolation. That is, after a few optimization iterations, the control variables are likely to stray outside the range of input-output values in the training data set. Extrapolation is generally viewed as a major difficulty in application of machine learning algorithms. One approach to deal with extrapolation is retraining, where the model is retained after including additional simulated realizations (labels) that correspond to control variables in the proximity of the current iterate. The retraining process is expected to improve the accuracy of the proxy model. Retraining involves several hyperparameters, including the number of additional samples, the sampling range (radius), and the initialization of the retraining process, among others. In this section, a simple retraining strategy is performed and discussed without highlighting the effects of hyperparameters. The goal is to show the potential improvement that can be achieved through the retraining process.

Figure 16 visualizes the distribution of the simulated data by projecting the high-dimensional input arrays onto only three leading Principal Components (PCs). The training samples and background (mostly validation data and unused data) are distributed evenly in the reduced dimensions. The optimization iteration and control solutions are derived from the optimization case “Initial Control 1.” The sampling range and the new sample size are set to be 0.08 and 10, respectively. That is, 10 new samples are generated by randomly adding a perturbation component of no more than 0.08 to the control solutions in the last iteration. The additional samples are added to the 10 initial samples and used to train the RNN models for Fault Zones 2 and 3. After training, the models are used to continue the optimization from the last iteration.

Figure 17 shows the values of the objective function before and after the retraining process, along with the predicted and simulated specific enthalpy after retraining and at the last iteration. The results show that after retraining the proxy model, the predicted values provide closer matches to the simulated values. The retraining process also results in a slight improvement in the net power generation. However, the changes in the control variables after retraining are not significant as the control variables are already close to the minimum values obtained

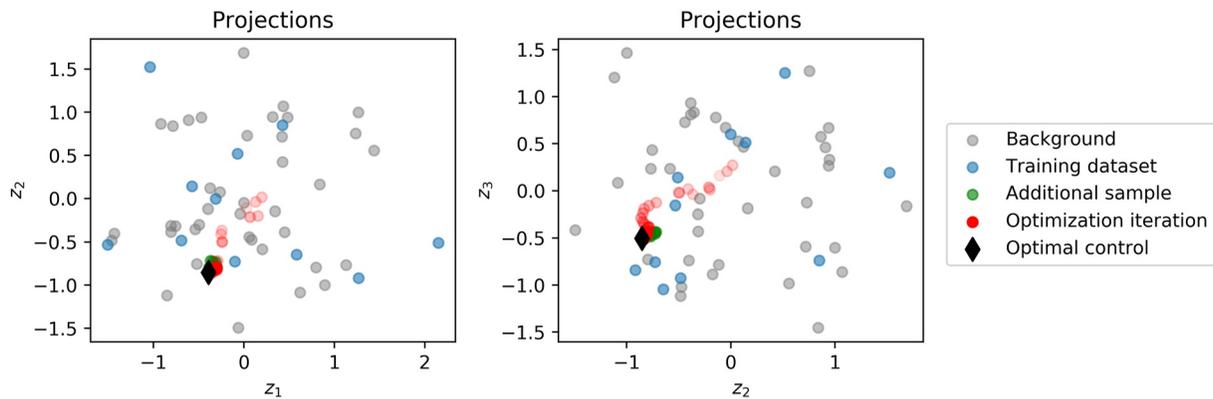


Figure 16. Visualization of the input control samples by projecting on the first three principal components using PCA. The sampling range and size for the additional samples are 0.08 and 10, respectively.

from the simulation-based optimization. It is important to note that these observations are limited to the examples performed in this work and do not necessarily generalize. In general, the need for retraining and its impact on the prediction performance are problem-specific. In this work, the data for retraining are generated after examining different ratios of new to initial sample sizes. In general, when the number of new samples is not sufficient, the retraining process may not be effective. On the other hand, an insufficient number of initial samples might lead to overfitting and poor prediction performance. A more detailed investigation of the retraining process and effective implementation strategies is needed to provide additional insight about the necessity and importance of this step.

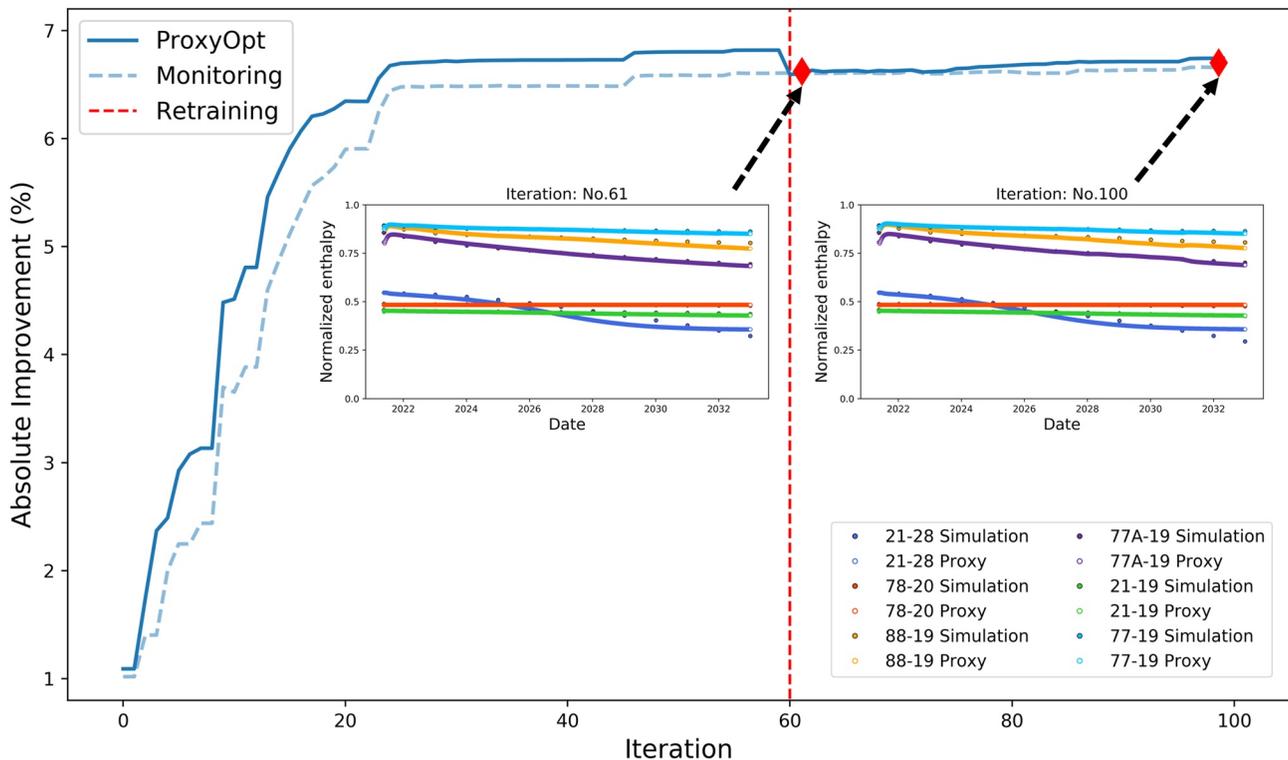


Figure 17. Objective function values during optimization iterations with retraining process using the CNN-RNN proxy model. The vertical dashed line (shown in red) indicates the retraining step, which is introduced at the 60th iteration after the first round of optimization. Two subplots show the comparisons between predicted enthalpy from proxy model and simulated enthalpy from simulation model. Two comparisons (red diamonds) are made at the 61st iteration (right after retraining) and at 100th iteration (last iteration of second-round optimization), respectively.

Table 3
Summary of the Average Computational Cost of Gradient Calculation and Forward Simulation Run

Model	Simulation model (weekly time size)	Simulation model (larger time size)	RNN proxy
Forward run (CPU time, secs)	2,143	271	0.411
Gradient Calculation (Simulation runs)	–	169	0
Gradient Calculation (CPU time, secs)	–	45,799	0.865

Note. The simulation model with weekly time size is not used in the gradient calculation. Therefore, the corresponding data is blank.

4.3. Computational Cost

The main computational cost of the proposed workflow is related to the generation of the simulated data set and the calculation of the required gradients. The cost of training the CNN-RNN models compared to the cost of a forward simulation run is typically negligible. Table 3 lists the average computational costs of running simulation models and generating the gradient information for the examples used in this paper. Once the proxy model is built, the computational costs associated with the prediction and gradient calculation are insignificant, which is one of the main motivations for developing the proposed proxy models. Each iteration of the proxy-based optimization typically takes less than one second. The simulation-based optimization, on the other hand, uses several simulation runs with the finite-difference approximation framework to estimate the required gradients. In our examples, the computational cost of finite-difference gradient calculation is 12.72 CPU hr. For comparison, each simulation run to generate the simulated data set for training the proxy model takes approximately 45 min.

The simulation software in this work is TETRAD, which was run on a PC with 8 CPU cores as it does not support parallelization on HPC. Therefore, gradient calculation for simulation-based optimization is processed in parallel and using only 8 CPUs. The simulation-based optimizations for Experiment 2 are completed using 8 CPU cores in 60 hr (6,084 simulation runs), or 457.99 CPU hr equivalently. Table 4 summarizes the computational cost of the simulation-based and proxy-based optimization approaches for the second experiment. The training of the proposed proxy model only requires a small number of training samples. Even though we generated 61 simulated samples for the second experiment, only 21 samples are used for training (10 random samples and 1 historical sample) and validating (10 random samples). The remaining 40 simulated samples are not used but are shown as the background in Figure 16. In general, the proxy-based optimization can be used to significantly reduce the computational cost of the optimization workflows.

5. Conclusion

In summary, computationally expensive simulation models pose a barrier for implementing field-scale optimization, especially when the input control variables are high-dimensional. In this paper, a workflow is proposed to develop deep learning proxy models to enable efficient field-scale optimization implementation to maximize energy production from geothermal fields. The workflow is presented and evaluated using field-scale examples. The results suggest that the proposed model can learn input-output patterns and relations from training datasets to provide reliable and efficient model predictions for performance optimization. In some optimization cases, the proxy model may need to extrapolate beyond the training data set, an issue that may prompt the use of retraining before completing the optimization process. Another important extension of the current workflow is related to the treatment of geologic uncertainty in the prediction by the deep learning proxy models, which can have important implications for optimization. One straightforward way to address the geologic uncertainty is to train multiple CNN-RNN proxy models using simulated data from different simulation models with different geologic realizations. This can be followed by a robust optimization problem formulation using an ensemble of proxy models for prediction.

The approaches described here resemble the works of Kim et al. (2022) and Schulte et al. (2020), which involve many simulation runs to generate training data. An alternative approach is to treat the geologic uncertainty as part of the inputs into a proxy model (Kim & Durlofsky, 2022), which will require the proxy model to have a more complex architecture with geologic maps as additional inputs. In this approach, one can train a single, and more complex, proxy model to represent different realizations instead of introducing multiple proxy models to capture the behavior of different realizations. However, additional complexity can lead to learning in high-dimensional spaces, which can expose the deep learning proxy model to challenging extrapolation tasks (Balestrieri et al., 2021). These potential

Table 4
Summary of the Computational Cost of the Simulation-Based and Proxy-Based Optimization in the Second Experiment

Model	Simulation run per gradient	Total simulation run	CPU time (hours)
Proxy-based Optimization	–	21	12.50
Simulation-based Optimization	169	6,084	457.99

Note. The simulation model used for simulation-based optimization is the model with larger time size of 180 days. The simulation run for proxy-based optimization is used to generate simulated samples for training and validating.

complications pose important questions that warrant further investigation of deep learning-based models for more advanced workflows that are needed in practical settings.

Data Availability Statement

Upon publication of this study, a link to the data, codes, and examples used in this study will be posted to the FAIR-compliant Zenodo online repository as well as our research website at <http://sees.usc.edu>. During the review process, the current versions of the files are posted to GitHub on <https://github.com/ZhenQin-USC/ProxyBasedOptimization>.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Geothermal Technologies Office award number DE-EE0008765. The corresponding author acknowledges the Energy Simulation support through an Industrial Research Chair Program on Subsurface Energy Data Science at USC. The authors also thank Cyrq Energy for supplying TETRAD software and numerical simulation for this work.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265–283). USENIX Association. Retrieved from <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>
- Asher, M. J., Croke, B. F. W., Jakeman, A. J., & Peeters, L. J. M. (2015). A review of surrogate models and their application to groundwater modeling: Surrogates of groundwater models. *Water Resources Research*, *51*(8), 5957–5973. <https://doi.org/10.1002/2015WR016967>
- Babaei, M., Norouzi, A. M., Nick, H. M., & Gluyas, J. (2022). Optimisation of heat recovery from low-enthalpy aquifers with geological uncertainty using surrogate response surfaces and simple search algorithms. *Sustainable Energy Technologies and Assessments*, *49*, 101754. <https://doi.org/10.1016/j.seta.2021.101754>
- Balestriero, R., Pesenti, J., & LeCun, Y. (2021). Learning in high dimension always amounts to extrapolation. arXiv. <https://doi.org/10.48550/ARXIV.2110.09485>
- Baş, D., & Boyacı, İ. H. (2007). Modeling and optimization I: Usability of response surface methodology. *Journal of Food Engineering*, *78*(3), 836–845. <https://doi.org/10.1016/j.jfoodeng.2005.11.024>
- Bertani, R. (2016). Geothermal power generation in the world 2010–2014 update report. *Geothermics*, *60*, 31–43. <https://doi.org/10.1016/j.geothermics.2015.11.003>
- Chen, M., Al-Maktoumi, A., Mahdi Rajabi, M., Izady, A., Al-Mamari, H., & Cai, J. (2021). Evaluation of CO₂ sequestration and circulation in fault-bounded thin geothermal reservoirs in north oman using response surface methods. *Journal of Hydrology*, *598*, 126411. <https://doi.org/10.1016/j.jhydrol.2021.126411>
- Chen, M., Tompson, A. F. B., Mellors, R. J., & Abdalla, O. (2015). An efficient optimization of well placement and control for a geothermal prospect under geological uncertainty. *Applied Energy*, *137*, 352–363. <https://doi.org/10.1016/j.apenergy.2014.10.036>
- Chen, Y., Oliver, D. S., & Zhang, D. (2009). Efficient ensemble-based closed-loop production optimization. *SPE Journal*, *14*(04), 634–645. <https://doi.org/10.2118/112873-PA>
- Cladouhos, T. T., Uddenberg, M. W., Swyer, M. W., Nordin, Y., & Garrison, G. H. (2017). Patua geothermal geologic conceptual model. *GRC Transactions*, *41*.
- Do, S. T., & Reynolds, A. C. (2013). Theoretical connections between optimization algorithms based on an approximate gradient. *Computational Geosciences*, *17*(6), 959–973. <https://doi.org/10.1007/s10596-013-9368-9>
- Fonseca, R. M., Chen, B., Jansen, J. D., & Reynolds, A. (2017). A stochastic simplex approximate gradient (StoSAG) for optimization under uncertainty: A stochastic simplex approximate gradient (StoSAG). *International Journal for Numerical Methods in Engineering*, *109*(13), 1756–1776. <https://doi.org/10.1002/nme.5342>
- Fonseca, R. M., Leeuwenburgh, O., Van den Hof, P. M. J., & Jansen, J. D. (2015). Improving the ensemble-optimization method through covariance-matrix adaptation. *SPE Journal*, *20*(01), 155–168. <https://doi.org/10.2118/163657-PA>
- Franco, A., & Vaccaro, M. (2014). Numerical simulation of geothermal reservoirs for the sustainable design of energy plants: A review. *Renewable and Sustainable Energy Reviews*, *30*, 987–1002. <https://doi.org/10.1016/j.rser.2013.11.041>
- Gudmundsdottir, H., & Horne, R. N. (2020). Prediction modeling for geothermal reservoirs using deep learning. In *PROCEEDINGS, 45th Workshop on Geothermal Reservoir Engineering, Stanford University*.
- Hecht-Méndez, J., De Paly, M., Beck, M., & Bayer, P. (2013). Optimization of energy extraction for vertical closed-loop geothermal systems considering groundwater flow. *Energy Conversion and Management*, *66*, 1–10. <https://doi.org/10.1016/j.enconman.2012.09.019>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Jafarpour, B., & McLaughlin, D. B. (2008). History matching with an ensemble Kalman filter and discrete cosine parameterization. *Computational Geosciences*, *12*(2), 227–244. <https://doi.org/10.1007/s10596-008-9080-3>
- Jafarpour, B., & McLaughlin, D. B. (2009). Reservoir characterization with the discrete cosine transform. *SPE Journal*, *14*(01), 182–201. <http://doi.org/10.2118/106453-PA>
- Jahandideh, A., & Jafarpour, B. (2018). Reservoir performance optimization under uncertain future development plans. D041S010R001. SPE. <https://doi.org/10.2118/190139-MS>
- Jahandideh, A., & Jafarpour, B. (2019). Closed-loop stochastic oilfield optimization under uncertainty in geologic description and future development plans. D010S001R006. SPE. <https://doi.org/10.2118/193856-MS>
- Jansen, J.-D., Bosgra, O. H., & Van den Hof, P. M. J. (2008). Model-based control of multiphase flow in subsurface oil reservoirs. *Journal of Process Control*, *18*(9), 846–855. <https://doi.org/10.1016/j.jprocont.2008.06.011>
- Jiang, A., & Jafarpour, B. (2021a). Deep convolutional autoencoders for robust flow model calibration under uncertainty in geologic continuity. *Water Resources Research*, *57*(11). <https://doi.org/10.1029/2021WR029754>
- Jiang, A., & Jafarpour, B. (2021b). Inverting subsurface flow data for geologic scenarios selection with convolutional neural networks. *Advances in Water Resources*, *149*, 103840. <https://doi.org/10.1016/j.advwatres.2020.103840>
- Jiang, A., Qin, Z., Cladouhos, T. T., Faulder, D., & Jafarpour, B. (2021). Recurrent neural networks for prediction of geothermal reservoir performance. In *Proceedings, 46th workshop on geothermal reservoir engineering, Stanford University*.
- Jiang, A., Qin, Z., Faulder, D., Cladouhos, T. T., & Jafarpour, B. (2022). Recurrent neural networks for short-term and long-term prediction of geothermal reservoirs. *Geothermics*, *104*, 102439. <https://doi.org/10.1016/j.geothermics.2022.102439>

- Juliusson, E., & Horne, R. N. (2013). Optimization of injection scheduling in fractured geothermal reservoirs. *Geothermics*, 48, 80–92. <https://doi.org/10.1016/j.geothermics.2013.05.004>
- Kim, J., Choe, J., & Lee, K. (2022). Sequential field development plan through robust optimization coupling with CNN and LSTM-based proxy models. *Journal of Petroleum Science and Engineering*, 209, 109887. <https://doi.org/10.1016/j.petrol.2021.109887>
- Kim, Y. D., & Durlafsky, L. J. (2021). A recurrent neural network–based proxy model for well-control optimization with nonlinear output constraints. *SPE Journal*, 26(04), 1837–1857. <https://doi.org/10.2118/203980-PA>
- Kim, Y. D., & Durlafsky, L. J. (2022). Convolutional-recurrent neural network proxy for robust optimization and closed-loop reservoir management. (Version 1). <https://doi.org/10.48550/ARXIV.2203.07524>
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv. <https://doi.org/10.48550/ARXIV.1412.6980>
- Lalee, M., Nocedal, J., & Plantenga, T. (1998). On the implementation of an algorithm for large-scale equality constrained optimization. *SIAM Journal on Optimization*, 8(3), 682–706. <https://doi.org/10.1137/S1052623493262993>
- Laloy, E., Héroult, R., Lee, J., Jacques, D., & Linde, N. (2017). Inversion using a new low-dimensional representation of complex binary geological media based on a deep neural network. *Advances in Water Resources*, 110, 387–405. <https://doi.org/10.1016/j.advwatres.2017.09.029>
- Lenhart, T., Eckhardt, K., Fohrer, N., & Frede, H.-G. (2002). Comparison of two different approaches of sensitivity analysis. *Physics and Chemistry of the Earth, Parts A/B/C*, 27(9–10), 645–654. [https://doi.org/10.1016/S1474-7065\(02\)00049-9](https://doi.org/10.1016/S1474-7065(02)00049-9)
- Lund, J. W., & Toth, A. N. (2021). Direct utilization of geothermal energy 2020 worldwide review. *Geothermics*, 90, 101915. <https://doi.org/10.1016/j.geothermics.2020.101915>
- Murphy, J., Holt, R., & Morrison, M. (2017). A numerical model case study of the patua geothermal field. *GRC Transactions*, 41.
- Olasolo, P., Juárez, M. C., Morales, M. P., D'Amico, S., & Liarte, I. A. (2016). Enhanced geothermal systems (EGS): A review. *Renewable and Sustainable Energy Reviews*, 56, 133–144. <https://doi.org/10.1016/j.rser.2015.11.031>
- Patterson, J. R., Cardiff, M., & Feigl, K. L. (2020). Optimizing geothermal production in fractured rock reservoirs under uncertainty. *Geothermics*, 88, 101906. <https://doi.org/10.1016/j.geothermics.2020.101906>
- Pollack, A., & Mukerji, T. (2019). Accounting for subsurface uncertainty in enhanced geothermal systems to make more robust techno-economic decisions. *Applied Energy*, 254, 113666. <https://doi.org/10.1016/j.apenergy.2019.113666>
- Razak, S. M., & Jafarpour, B. (2020). Convolutional neural networks (CNN) for feature-based model calibration under uncertain geologic scenarios. *Computational Geosciences*, 24(4), 1625–1649. <https://doi.org/10.1007/s10596-020-09971-4>
- Razavi, S., Tolson, B. A., & Burn, D. H. (2012). Review of surrogate modeling in water resources. *Water Resources Research*, 48(7). <https://doi.org/10.1029/2011WR011527>
- Samin, M. Y., Faramarzi, A., Jefferson, I., & Harireche, O. (2019). A hybrid optimisation approach to improve long-term performance of enhanced geothermal system (EGS) reservoirs. *Renewable Energy*, 134, 379–389. <https://doi.org/10.1016/j.renene.2018.11.045>
- Schulte, D. O., Arnold, D., Geiger, S., Demyanov, V., & Sass, I. (2020). Multi-objective optimization under uncertainty of geothermal reservoirs using experimental design-based proxy models. *Geothermics*, 86, 101792. <https://doi.org/10.1016/j.geothermics.2019.101792>
- Sheikholeslami, R., & Razavi, S. (2017). Progressive latin hypercube sampling: An efficient approach for robust sampling-based analysis of environmental models. *Environmental Modelling and Software*, 93, 109–126. <https://doi.org/10.1016/j.envsoft.2017.03.010>
- Shi, Y., Song, X., & Song, G. (2021). Productivity prediction of a multilateral-well geothermal system based on a long short-term memory and multi-layer perceptron combinatorial neural network. *Applied Energy*, 282, 116046. <https://doi.org/10.1016/j.apenergy.2020.116046>
- Sigurdardottir, S. R., Valfell, A., Pálsson, H., & Stefánsson, H. (2015). Mixed integer optimization model for utilizing a geothermal reservoir. *Geothermics*, 55, 171–181. <https://doi.org/10.1016/j.geothermics.2015.01.006>
- Song, G., Song, X., Li, G., Shi, Y., Wang, G., Ji, J., et al. (2021). An integrated multi-objective optimization method to improve the performance of multilateral-well geothermal system. *Renewable Energy*, 172, 1233–1249. <https://doi.org/10.1016/j.renene.2021.03.073>
- Tian, C., & Horne, R. N. (2017). *Recurrent neural networks for permanent downhole gauge data analysis*. SPE. D011S008R007. <https://doi.org/10.2118/187181-MS>
- Van Essen, G., Zandvliet, M., Van Den Hof, P., Bosgra, O., & Jansen, J.-D. (2009). Robust waterflooding optimization of multiple geological scenarios. *SPE Journal*, 14(01), 202–210. <https://doi.org/10.2118/102913-PA>
- Vinsome, P. K. W., & Shook, G. M. (1993). Multi-purpose simulation. *Journal of Petroleum Science and Engineering*, 9(1), 29–38. [https://doi.org/10.1016/0920-4105\(93\)90026-B](https://doi.org/10.1016/0920-4105(93)90026-B)
- Zarrouk, S. J., & Moon, H. (2014). Efficiency of geothermal power plants: A worldwide review. *Geothermics*, 51, 142–153. <https://doi.org/10.1016/j.geothermics.2013.11.001>