Optimizing Steam Distribution in SAGD

Jamal Bajwa, Nexen Energy ULC

November 21st, 2018

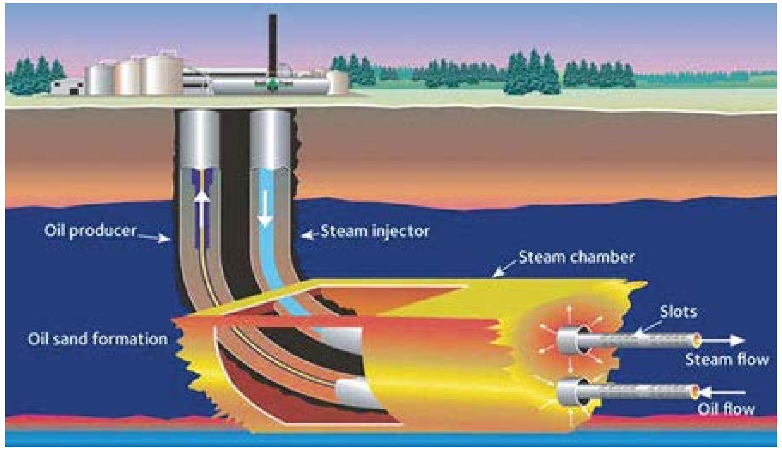
A CNOOC LIMITED COMPANY

A New Energy

Table of Contents

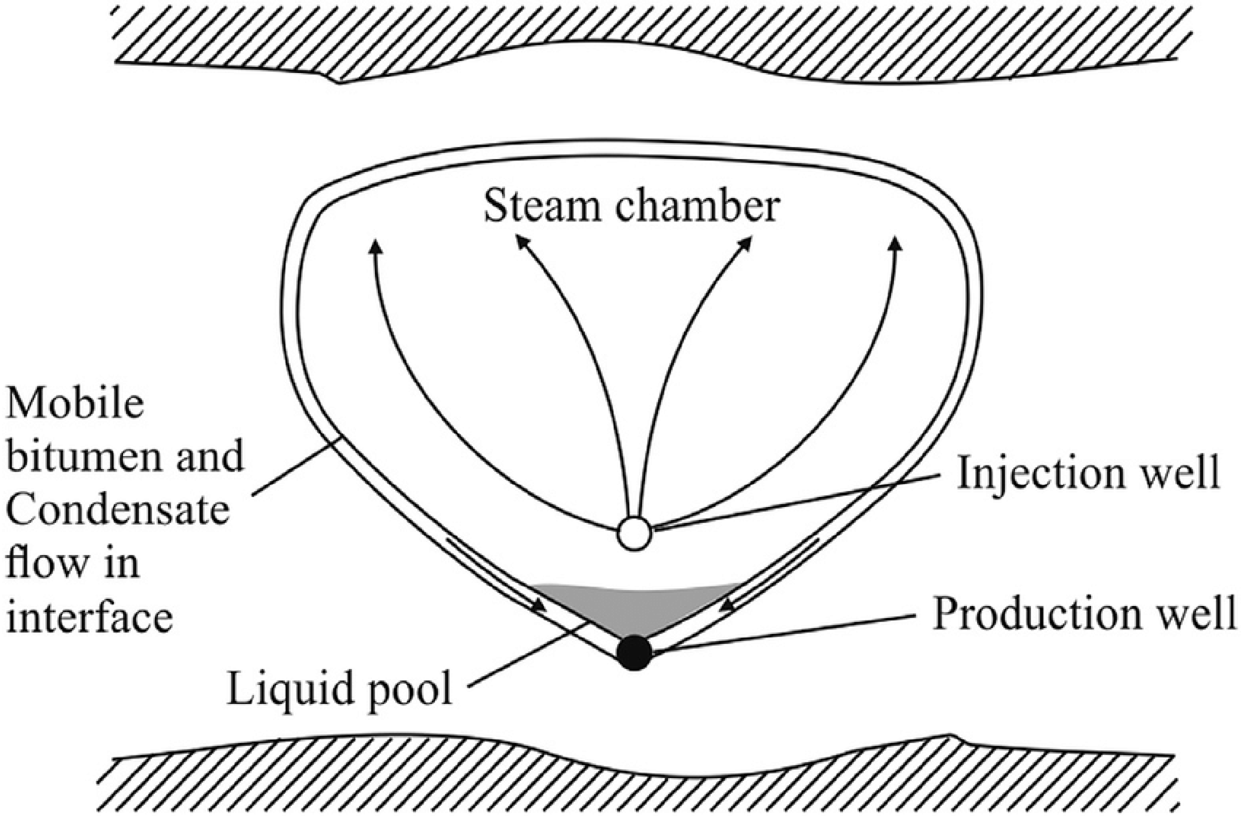
Overview of SAGD

- Conventional Reservoirs contain oil that is very mobile (low viscosity)
- Unconventional Reservoirs contain oil that is not easily accessible through conventional recovery methods
- **Oil Sands** is a type of unconventional reservoir that contains Bitumen (very low viscosity)
- Bitumen, at reservoir conditions, has extremely high viscosity and is not mobile
- One must heat up Bitumen to reduce its viscosity and allow it to become mobile
- SAGD = **Steam Assisted Gravity Drainage**
  - Steam is injected into the reservoir through **injector wells** to heat the Bitumen and reduce its viscosity
  - Heated bitumen drains down to the **producer well** because of gravity and is pumped to surface



https://www.upstreampumping.com/article/production/high-temperature-electric-submersible-pumps-effective-oil-sands-production

First Principles Methods (Butler)



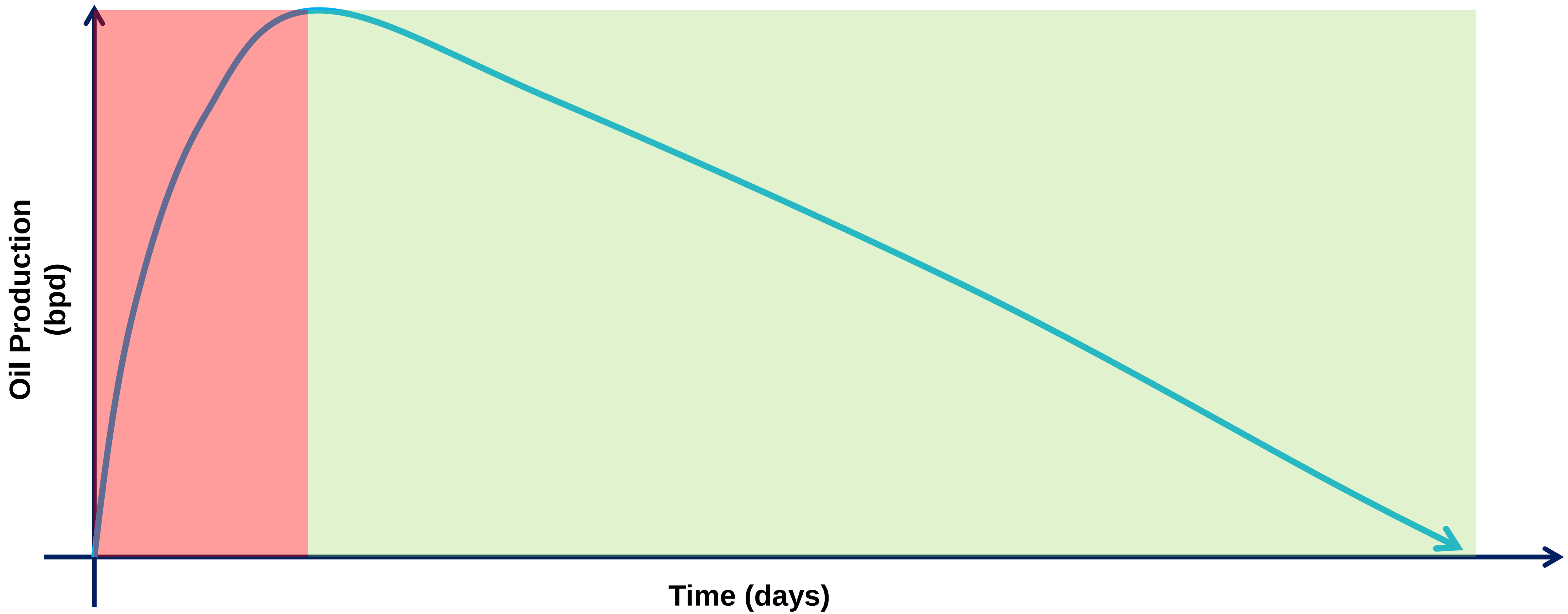
- SAGD is a recovery method that was proposed by Roger M. Butler in the 1980s

- Butler published his estimation methods in an industry classic book called *“Thermal Recovery of Oil and Bitumen”* in 1991

- In this book, Butler proposed a set of methods to predict the oil production rate from each SAGD well using a hybrid first-principles and empirical method (from his lab experiments & some pilots)

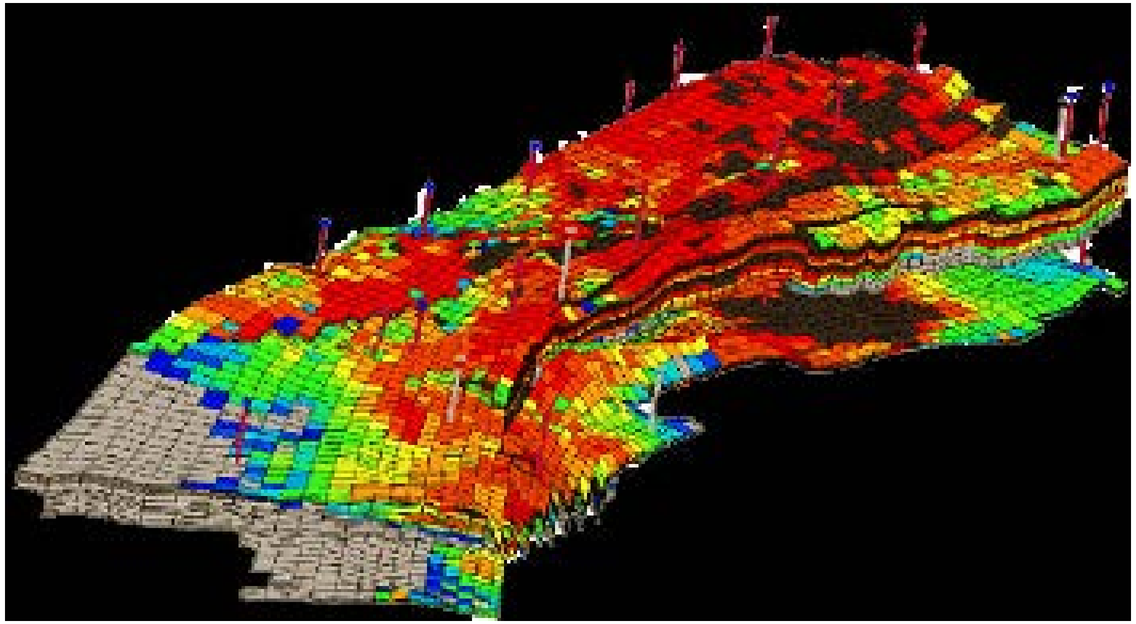
- Most companies and auditors continue to use a variant of the original Butler model today for fast approximation

First Principles Methods (Butler)



- The Butler Model predicts two regions: The Rising Steam Chamber (Red) and The Depletion Phase (Green)
- Qualitatively, the production rate in this model is a function of geological properties of the reservoir (porosity, permeability, water saturation, residual oil saturation etc.), heat transfer properties of the surrounding rock and reservoir oil properties (viscosity)
- Key Point: **The Butler Model only implicitly accounts for Steam Injection as a causal variable in Oil Production**

First Principles Methods (Thermal Simulation)



https://wiki.seg.org/wiki/Reservoir_simulation

Diffusion Equation (Flow)

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial p}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 p}{\partial \theta^2}+\frac{\partial^2 p}{\partial z^2}=\frac{\phi\mu c_t}{k}\frac{\partial p}{\partial t}$$

Fourier Equation (Heat Transfer)

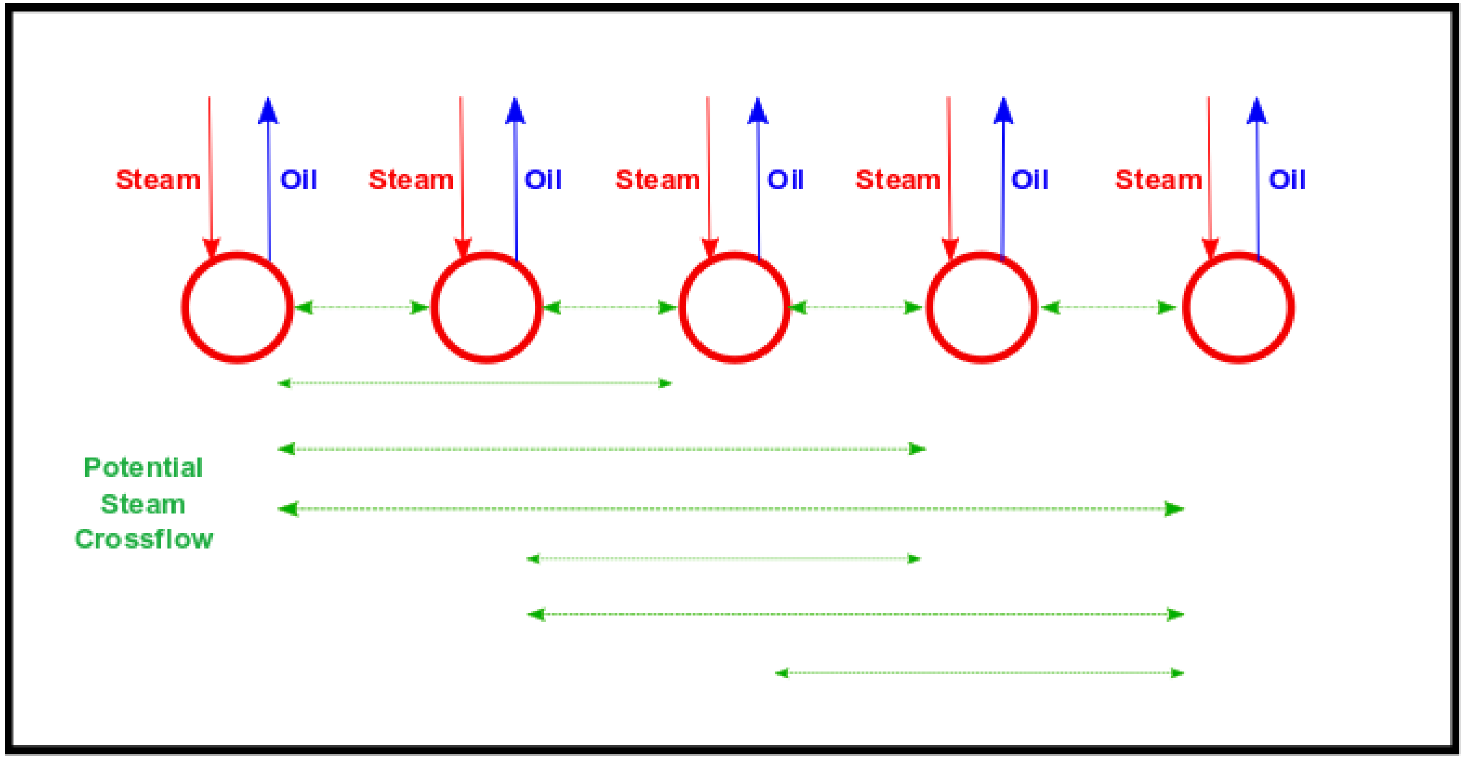
$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)+q_V=\rho c_p\frac{\partial T}{\partial t}$$

Introduce geological map and boundary conditions (etc.)

- One can write down partial differential equations that can best describe the physics of the reservoir
- Given a grid of the reservoir with geological parameters (which is often a solution to some PDEs!) and sensible boundary conditions, these partial differential equations can be solved numerically (reservoir simulators such as CMG do this for us and cost a lot of money)
- This is a step improvement from Butler because now **you can test how your steam injection (strategy) causally affects oil production**
- But numerical methods to solve these simulations are very slow…

Empirical Methods

- In day to day operation of the reservoir, we want to understand how injecting steam into the reservoir affects oil production
- The reservoir contains significant amount of interaction – steam injection in one well can affect oil production in other wells
- Being able to model the production of oil as a function of causal steam variables can help guide optimization strategy



- The Butler Model (or Modified Butler) cannot predict these interactions directly (you would have to handcraft the features)
- Simulations are too slow to "tune" to observed data in order to get accurate predictions

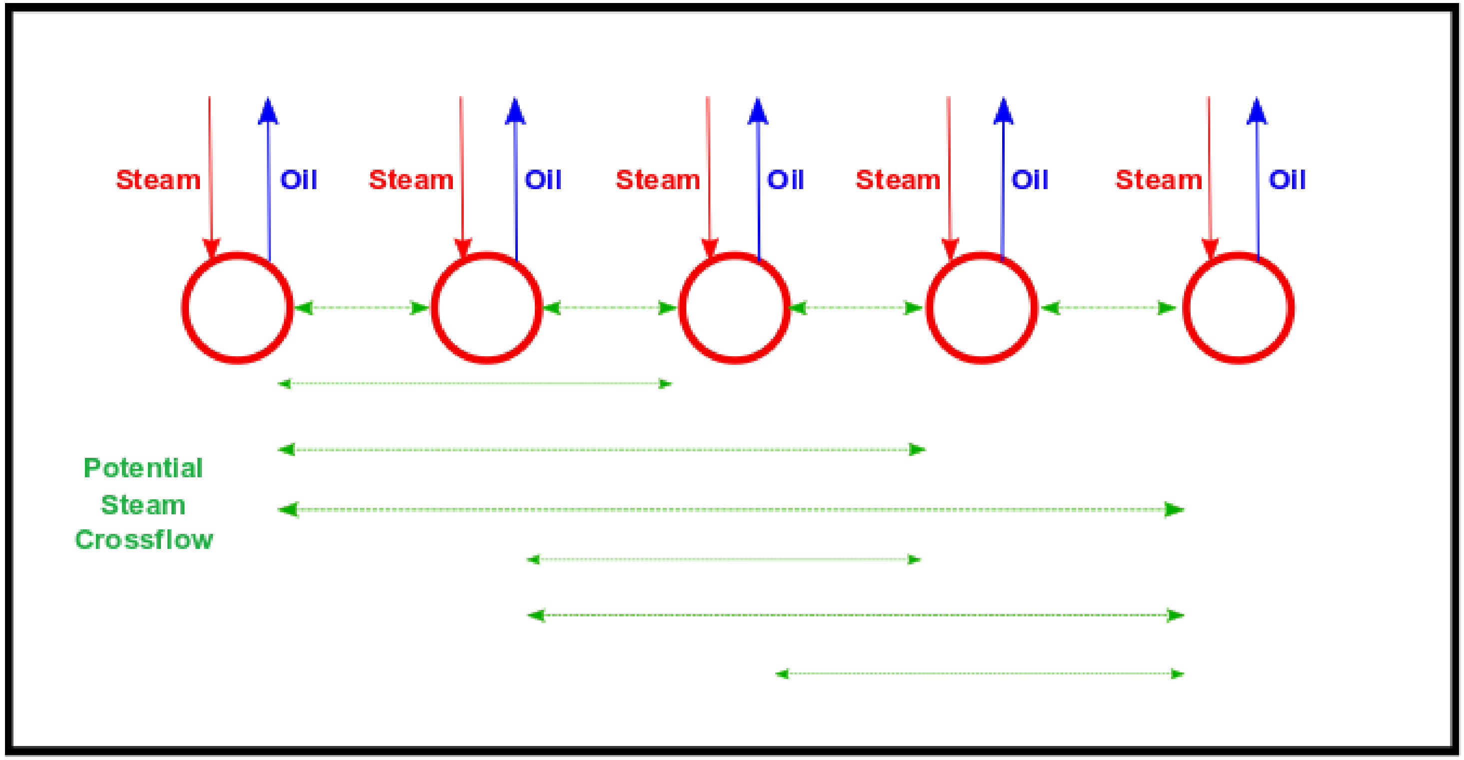
Philosophy for Field Optimization

- **Optimal Steam Injection** in the short and long run is critical for maximizing Bitumen Production
- Significant interaction between wellpairs and their configuration makes it difficult to understand the true SOR per well
- There are two ways of attacking this problem: **Bottom-Up** and **Top-Down**

Bottom Up: First Principles Modelling	Top Down: Empirical Modelling
• Solutions to Dynamic Partial Differential Equations allow us to simulate the physical behavior of the reservoir • Precise seismic, geologic and drilling information required with a complete understanding of the inherently time-variant reservoir properties • Precise information is not available and the solutions are probabilistic at best • Simulations are extremely expensive computationally	• Develop predictive models for well performance using predictive field variables that we **control** • Multivariate, non-linear models are able to capture the complex interactions between different wellpairs • The production models above can be passed through an optimization model to maximize Bitumen Production • **Empirical Models are constrained by the data used to construct them (major shortcoming)**

Mathematical Formulation of the Problem

Predictive Modelling

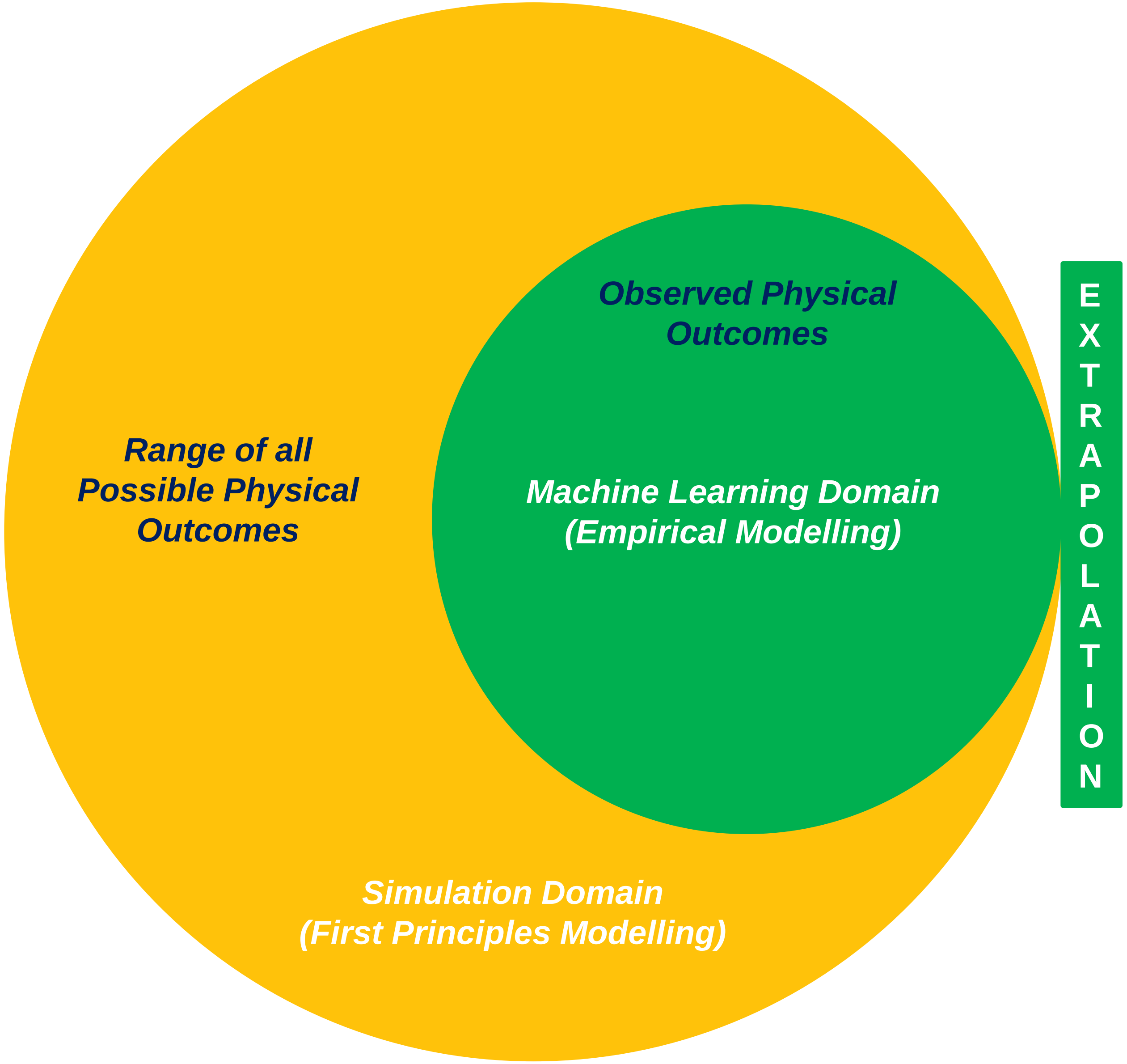


$$Oil_i = \boldsymbol{f}^*\left(\left(Steam_{heel_{1:k}}, Steam_{toe_{1:k}}\right)_{1:45\ days}\right) \quad \forall \ i = 1{:}\, N_{wells}\ \&\ k \in N_{wells}$$

$\boldsymbol{f}^*_{......}$ **denotes an arbitrary regression function that we need to solve for**

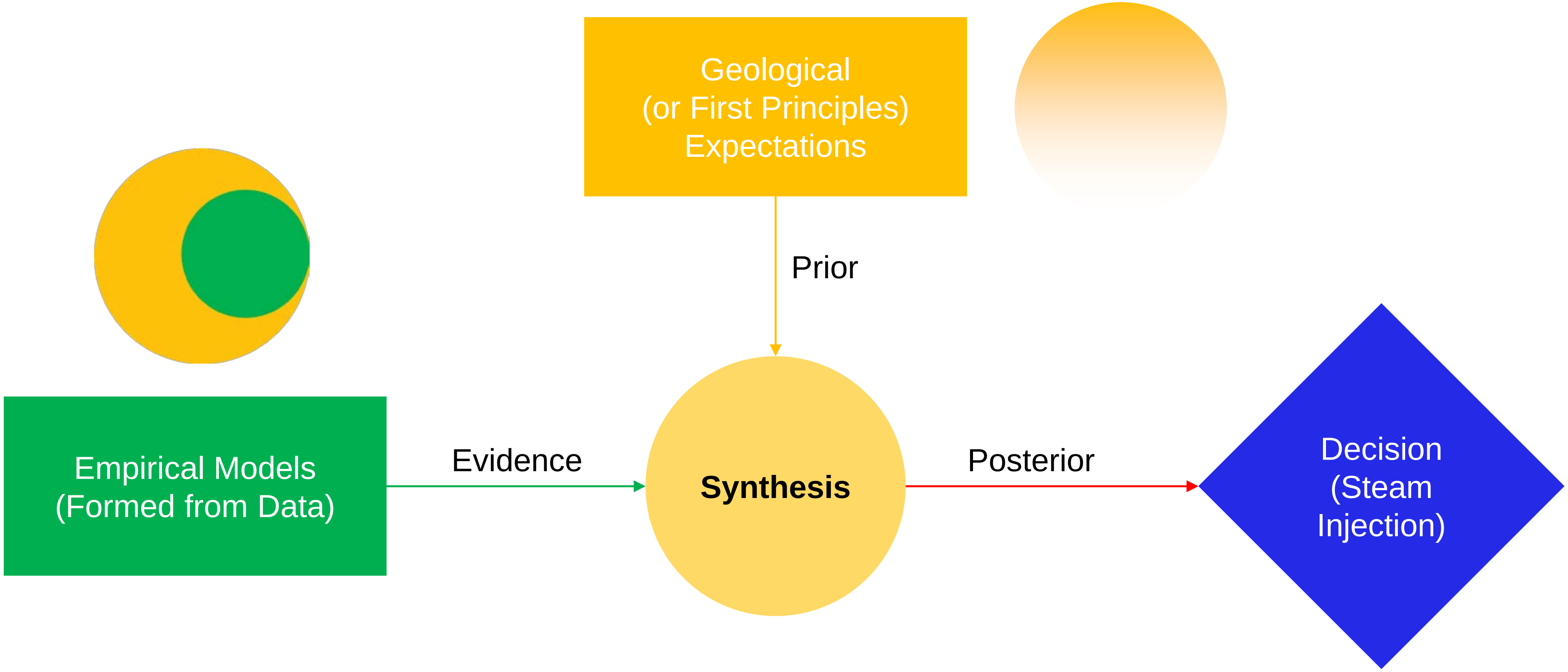
- Effectively, we need to solve for a function f* that uses causal steam variables as inputs from the recent history, and predicts expected oil tomorrow (or some days in the future)
- What is the best way to obtain f*? **We use modern deep learning techniques to obtain f*.**

Philosophy for Field Optimization



- The orange circle represents all the outcomes that are physically possible
- The green circle represents what has actually been observed (a small portion of all physical possibilities)
- The empirical modelling approach learns from observed outcomes and tries to infer causal relationships between Steam Injection and Oil Production in this space
  - In the absence of a deterministic simulation, leveraging empirical models to optimize field production is a viable option
- **We should use empirical models cautiously and constrain their usage to things that make physical sense!**

Short-Term Integration of Empirical Methods



- The synthesis stage can be thought of as a Bayesian Update of prior expectations **<u>given empirical evidence</u>**
- Our focus is on creating the best Empirical Models that can augment our decision-making process

Workflow

nexen
A CNOOC LIMITED COMPANY

Back-End Analytics

Front-End UI for Production Engineers

Historical Production Data

Short-Term Predictive Models for Reservoir Performance
(Production rate as a function of Steam Injection)

Reservoir Simulator
(Digital Twin to study perturbations and interactions, **the what-ifs**?)

Empirical Field Models serve as constraints

Objective Function

Optimization Model for Multi-Purpose Objectives

(Optimal Steam Distribution to Maximize Oil Rate, Marginal Cost/Benefit Analysis etc.)

LLK Production Recommendation Engine

(An intelligent agent that can recommend optimal operation of Well Pads using empirical performance models)

Physical Constraints, Tuning Parameters, Novel Algorithms

We are also deeply interested in understanding causal forces for oil production

Assumptions for Empirical Modelling

The assumptions below are critical to fully understand the use case for our approach:

1. **Steam Injection causes Bitumen (Oil) production**
   - Steam is responsible for supplying heat to the reservoir
   - This heat is responsible for reducing the viscosity and density of the Bitumen (oil), allowing it to drain to a Producer Well
   - We do not fully understand the interactions between steam injection from one well and Bitumen production from another well

2. **Bitumen Production is a combination of:**
   - ***Long Term Reservoir Response*** (natural response of steam heating the reservoir) and,
   - ***Short Term Production Process*** (removal of oil from the subsurface as it is produced)
   - *We assume that the wells are, on average, optimized and neglect using an ESP information in our current approach*

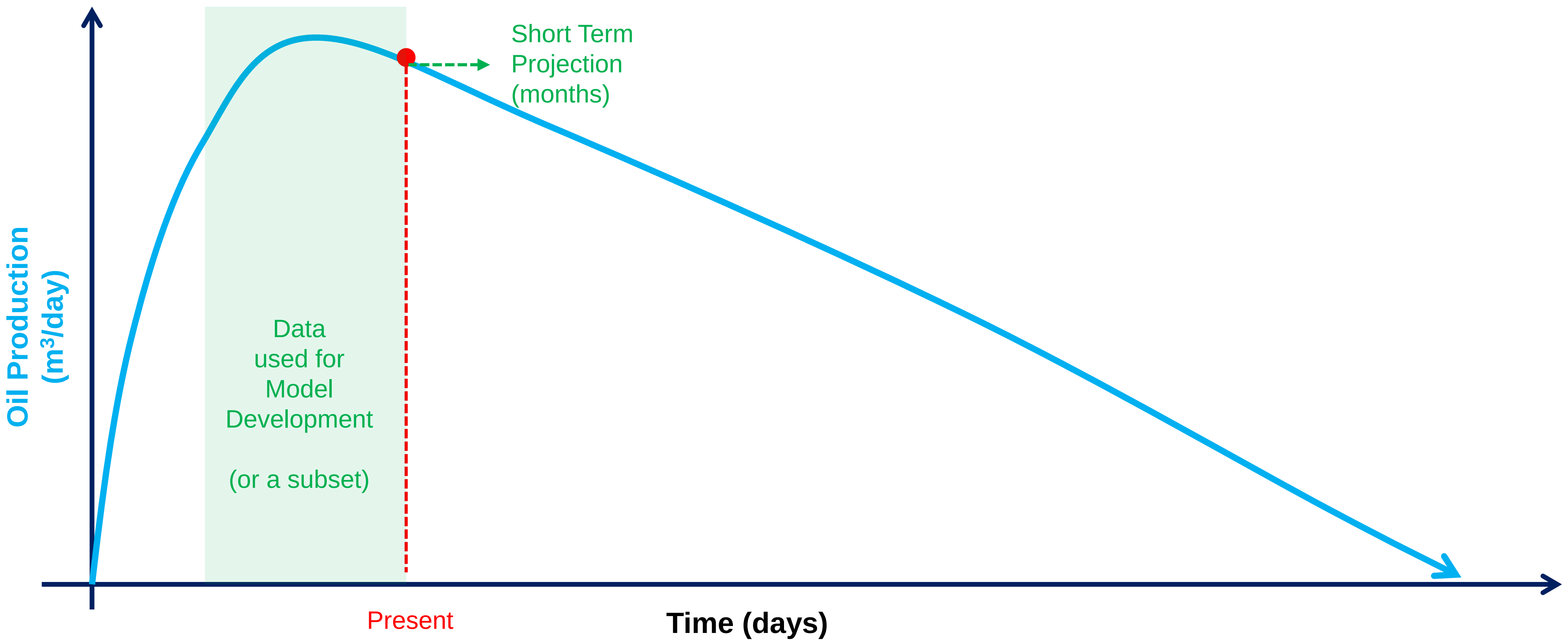
3. **Oil Production is a weak-stationary process**
   - *Details of this assumption are explained in the next slide.*

4. **Steam Injection from the last 45 days plays a larger causal role than the Steam Injection prior to the last 45 days**

5. **Production Data (Oil Cut, Steam Injection and Emulsion Produced) is reliable**

Assumption 3: Weak Stationary Process

Consider the following idealized Expected Production Profile (from Butler's Heuristics):



- Oil Production is changing as a function of time, which means the production process is non-stationary
  - *Non-Stationarity implies that the relationships between the causal variable (steam injection) and dependent variable (oil production) are changing with time*
- Because the reservoir decline observed to date is rather slow, we can make the assumption of weak stationarity and create stationary models for short-term optimization (subject to a long-term constraint, like Total Reservoir Pressure)

Predictive Models for Reservoir Performance

Steam to Heel

ESP Speed

Reservoir Cross-Section

Reservoir Outputs:

Subcool Per Well $(T_s)_i$

Reservoir Pressure $(P_{res})_i$

Oil Rate Per Well $(Oil)_i$

Water Rate Per Well $(Water)_i$

Steam to Toe

<u>Note:</u> This diagram shows the measured physical parameters that are controlled and influence reservoir performance

Empirical Reservoir Simulator

<u>Inputs (Predictors/Control Variables)</u>

Steam to Heel
Steam to Toe
ESP Speeds (Emulsion Produced)

<u>Outputs (Responses)</u>

Key Reservoir Properties (Pressure/Subcool)
Oil Rate per Well
Water Rate per Well

Predictive Models for Reservoir Performance

- The aforementioned regression problem can be written mathematically as:

$$T_{s_i} = \boldsymbol{f_k^*}\left(Steam_{heel_i}, Steam_{toe_i}, ESPSpeed_i\right) \quad \forall \quad i = 1{:}N_{wells}$$

Subcool

$$Oil_i = \boldsymbol{f_l^*}\left(Steam_{heel_i}, Steam_{toe_i}, ESPSpeed_i\right) \quad \forall \quad i = 1{:}N_{wells}$$

Oil

$$Water_i = \boldsymbol{f_m^*}\left(Steam_{heel_i}, Steam_{toe_i}, ESPSpeed_i\right) \quad \forall \quad i = 1{:}N_{wells}$$

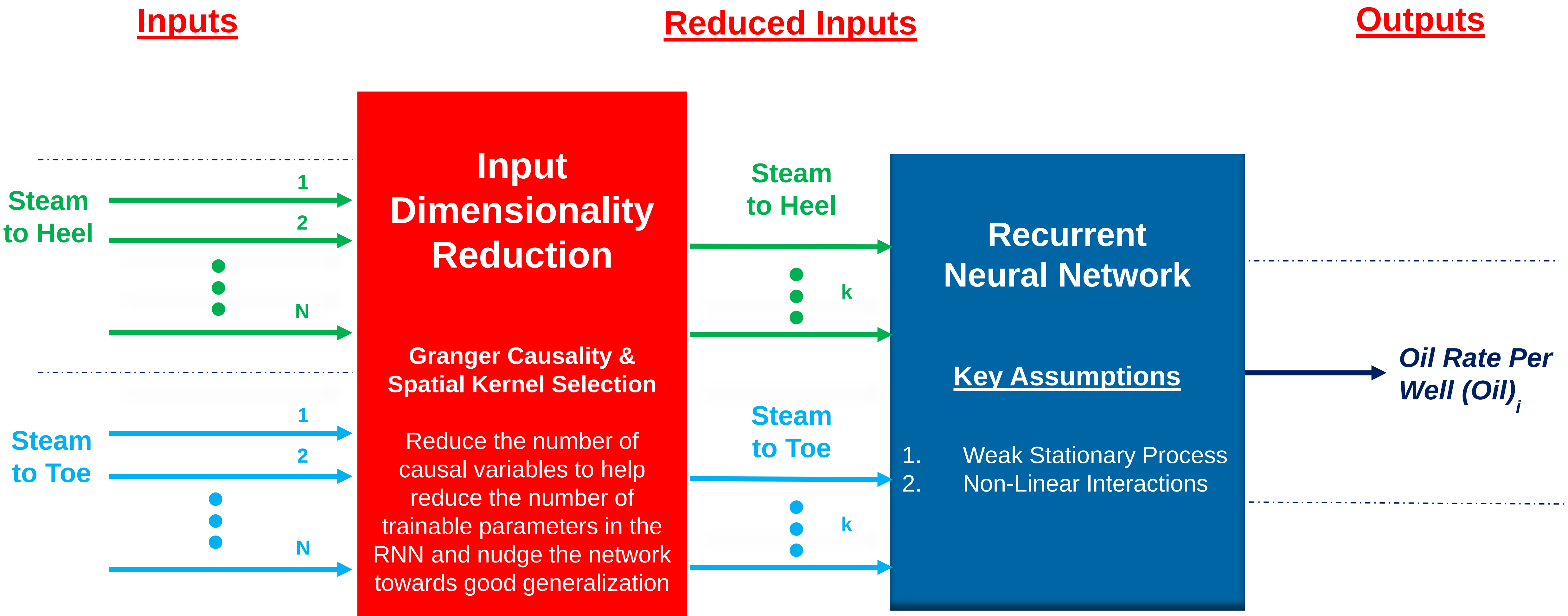
Water

$\boldsymbol{f_{k,l,m}^*}$ ...... **denote the arbitrary regression functions that we need to solve for**

There are several drawbacks of this regression approach (which we will elaborate on later):

- **Temporal Dependencies:** Reservoir behavior is transient and dynamic. Our current approach does not take into account this time-dependence and assumes the aggregated historical behavior projects into the short-term future.
- **Correlations in Input Predictors:** There is likely significant correlation between steam rates and ESP speeds of wells close to each other. To create accurate predictive models, **one needs a greater understanding of causal variables (interactions). The empirical data alone cannot provide us with understanding of true causality. We must make critical assumptions here.**

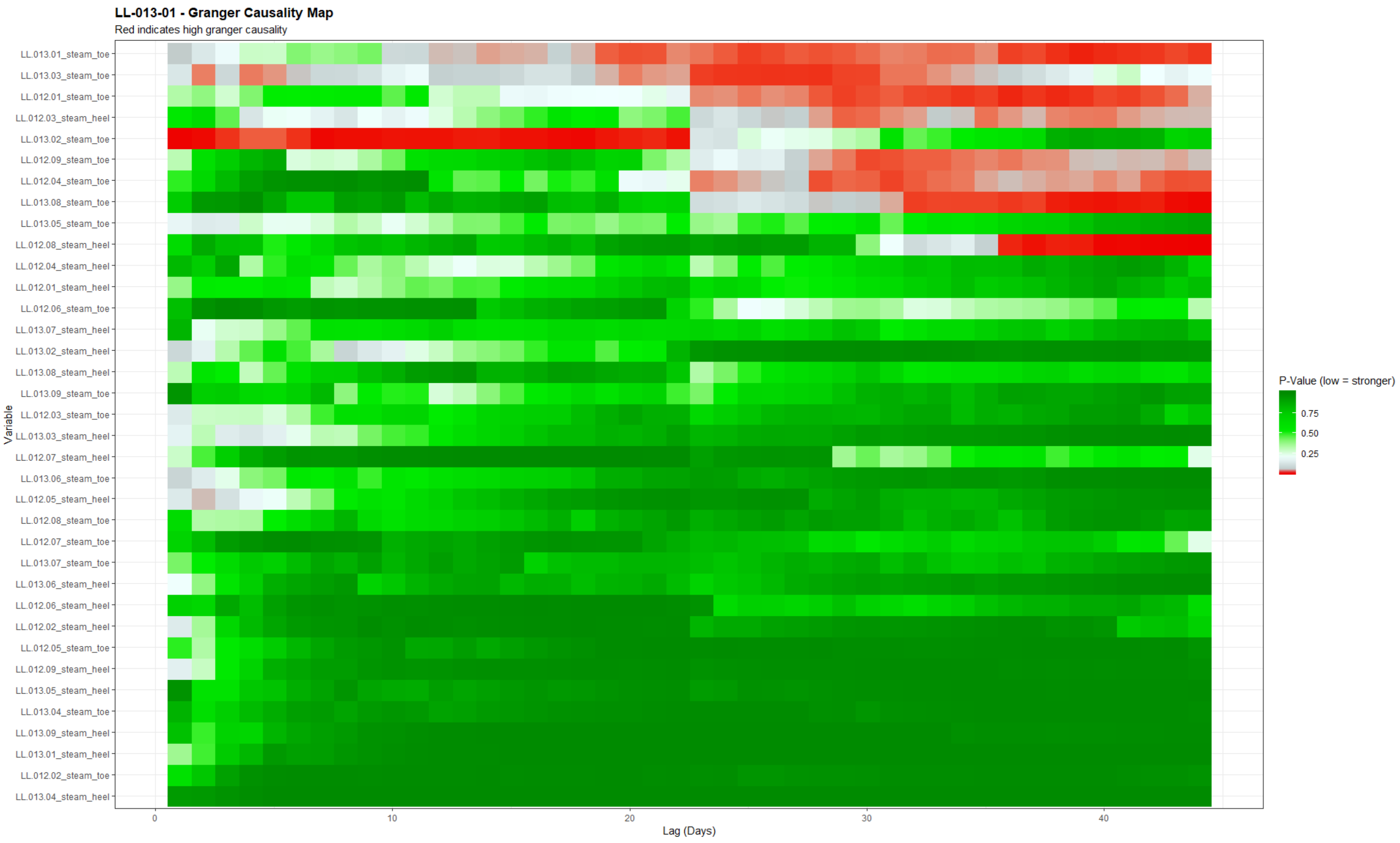
Reductionist RNN Approach (Preferred)



The model uses steam injection data from the last 30-45 days to predict current production

Removing predictive variables will create more uncertainty in the outputs. But we can be sure that these variables are "causal".

Granger Causality Example



- A more robust method to test for cross-correlation (developed in the field of economics in the 1970s)
- Other methods like transfer entropy work ok too
- ***Worth noting that we only used "training data" to avoid information leaks.***

Why RNN? (In Hindsight)

- We believe that this is a state-space problem, in that the reservoir state at one point affects the reservoir performance at the next point

- In our view of the reservoir, **sequence of Steam Injection plays a larger causal role than Steam Injection at one point**

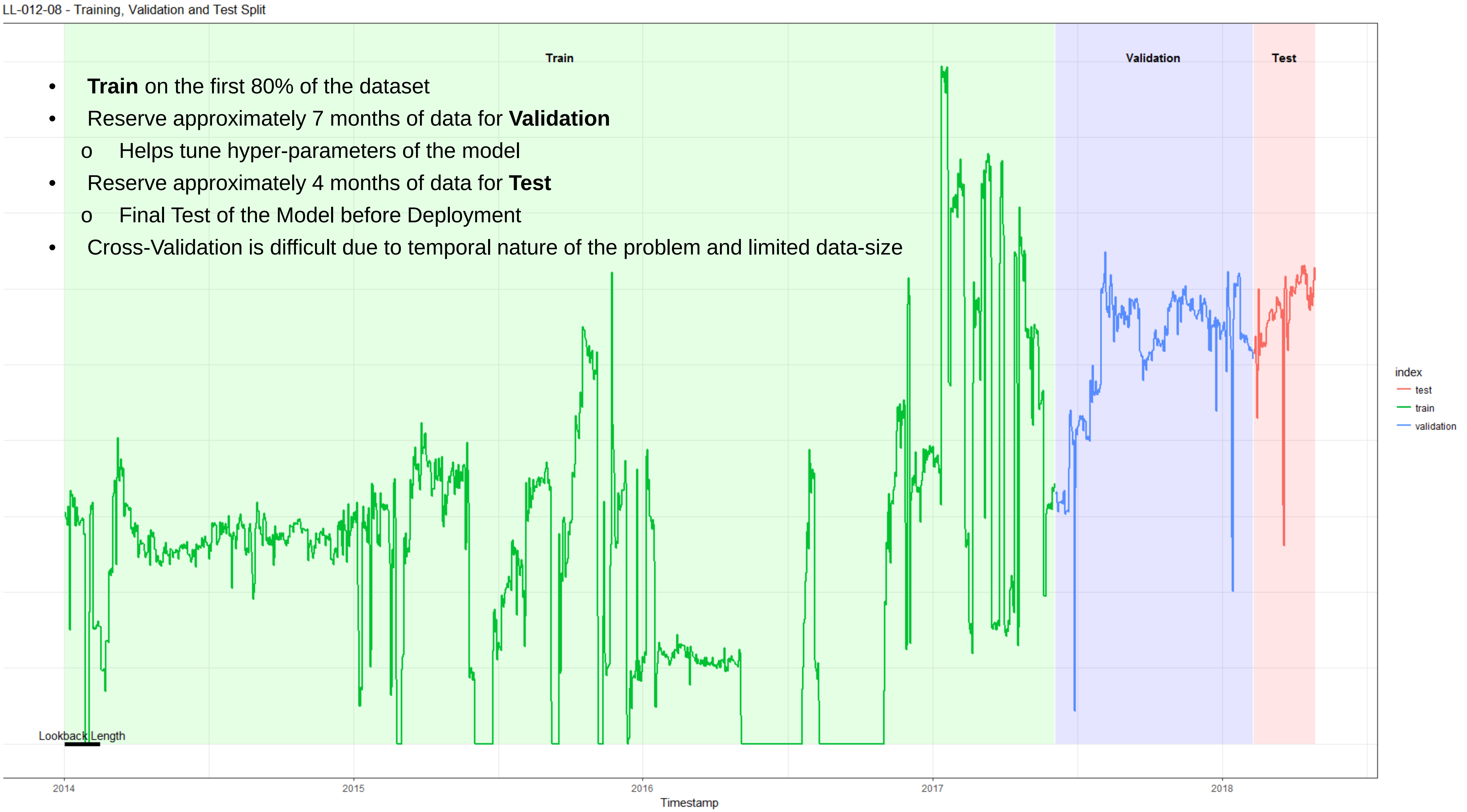
- We also tinkered with other deterministic models that could help us learn sequences:

    - *Linear Dynamical Models*
    - *Kalman Filters*
    - *Gradient Boosting (with feature generation)*
    - *Convolutional Neural Networks*

- We predominantly used Maximum Likelihood as our cost function during fitting (or minimizing MSE) and hope to convert our modelling framework to a more robust Bayesian framework

- In the end, our current view is that ***RNNs offer sufficient non-linearity, large enough hypothesis space and modestly decent predictions for us to continue using them unless another model type can offer better performance***

Overfitting: Validation and Test Philosophy



- **Train** on the first 80% of the dataset
- Reserve approximately 7 months of data for **Validation**
  - Helps tune hyper-parameters of the model
- Reserve approximately 4 months of data for **Test**
  - Final Test of the Model before Deployment
- Cross-Validation is difficult due to temporal nature of the problem and limited data-size

Some Details of RNN Architecture

Model Details or Hyperparameter	Value
Lookback	**30-35 days**
Forward Prediction Delay	**1-3 days**
Type of Recurrent Cell	**GRU (Gated Recurrent Unit)**
Activation Functions	**ReLu (Rectified Linear Units)**
Dropout Rate in Recurrent Layers	**0.5**
Dropout in Feedforward Layers	**0.1**
Total Model Parameters (Weights)	**~600 (fairly small)**

- These parameters were chosen through empirical testing – please come talk to me after if you'd like to hear my theories on why these work better than others!
- All models were developed using keras and tensorflow (R) – a package that serves as a wrapper to the keras/tensorflow environments in Python

```
Model
________________________________________________________________________________
Layer (type)                        Output Shape                    Param #
================================================================================
input_6 (InputLayer)                (None, None, 6)                 0
________________________________________________________________________________
gru_11 (GRU)                        (None, None, 5)                 180
________________________________________________________________________________
dropout_11 (Dropout)                (None, None, 5)                 0
________________________________________________________________________________
gru_12 (GRU)                        (None, 10)                      480
________________________________________________________________________________
dropout_12 (Dropout)                (None, 10)                      0
________________________________________________________________________________
dense_6 (Dense)                     (None, 1)                       11
================================================================================
Total params: 671
Trainable params: 671
Non-trainable params: 0
________________________________________________________________________________
```

Uncertainty in Predictions

- Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning (2016)
  - Yarin Gal and Zoubin Gharamani (2016)

- Gal and Gharamani show that one can obtain model uncertainty using dropout during prediction, and not just during training of the model. Dropout in Neural Networks is effectively viewed as approximate Bayesian inference in deep Gaussian Processes, and predicting with dropout turned on is effectively sampling from the posterior of the parameter space

- This helps us manage extrapolation risk **(model shows massive uncertainty on predicting over steam injection sequences and interactions that it hasn't seen before)**

Mathematical Formulation of the Problem

Optimization Model

- Assuming accurate oil prediction models as a function of steam injection have been tested, an overarching production optimization model can then be formulated:

Objective Function

$$\max \sum_{i=1}^{N_{wells}} Oil_i$$

Assumption: Pseudo-Stationary Process

Constraints

$$Oil_i = f_i^*\left(Steam_{heel_i}, Steam_{toe_i}\right) \quad \forall \quad i \in 1: N_{wells}$$

Oil Models

$$\sum_{i=1}^{N_{wells}} \left(Steam_{heel_i} + Steam_{toe_i}\right) \leq Steam\ Budget$$

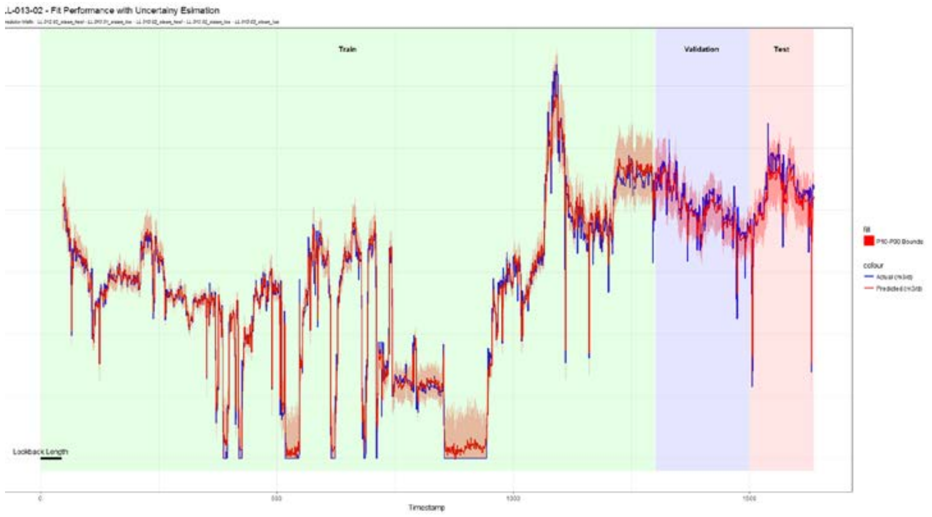
Steam Constraint

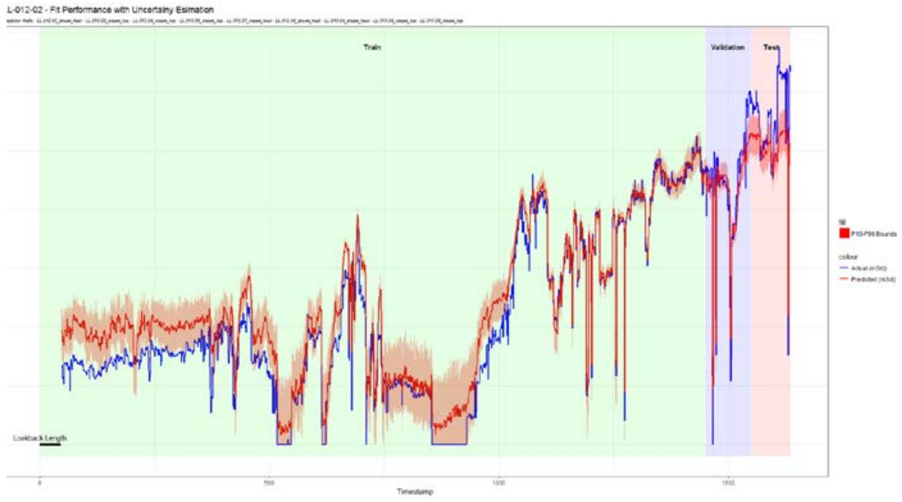
$$0 \leq Steam_{heel_i}, Steam_{toe_i} \leq Max$$

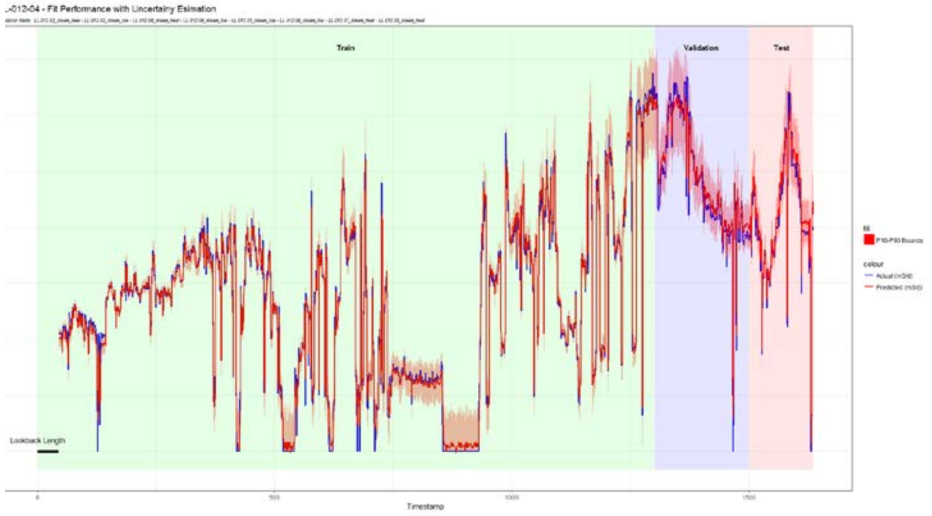
Physical Constraints

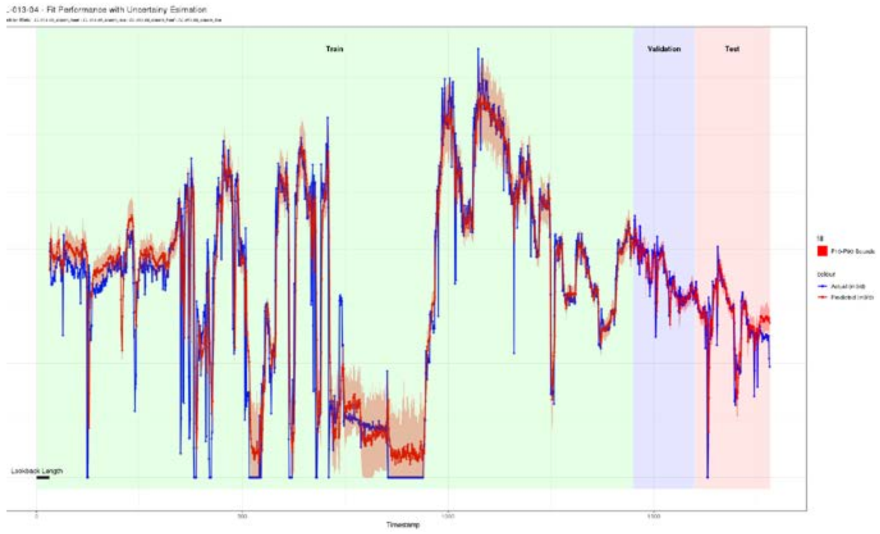
- Only the median (P50) of the predictive models is used within the optimizer – we model the uncertainty of the optimizer's recommendation through the method proposed by Gal et al.
- **If we were to use the model uncertainty during optimization, this becomes a reinforcement learning problem**

Some Sample Fits

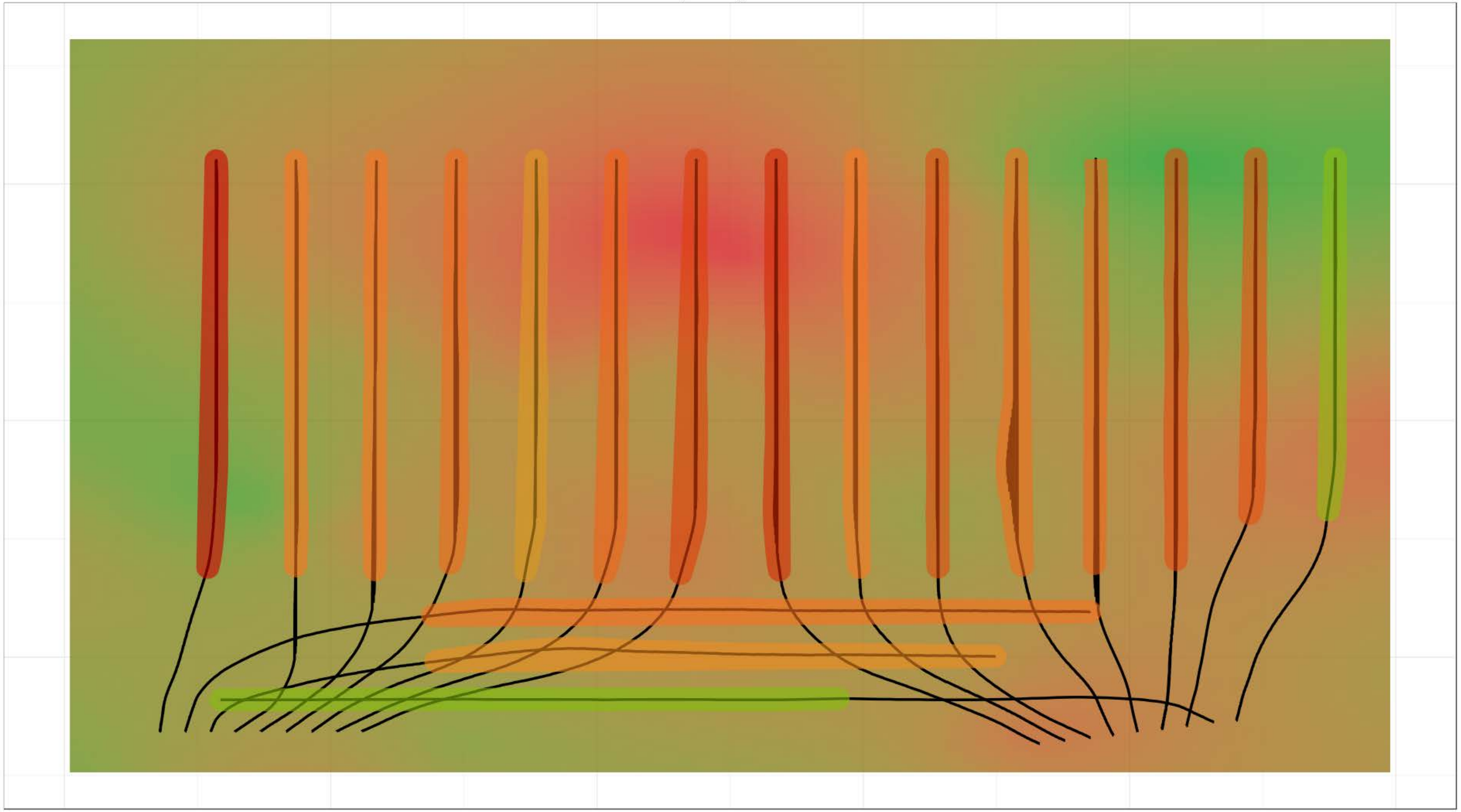








Sample Optimization Overlay on Geological Map



Future Opportunities

1. Using simpler models than RNN (such as models with attention mechanisms) and using Bayesian Optimization as our main method moving forward **(Short Term Fix)**

2. Tuning Reservoir Simulations using Bayesian Optimization to better encapsulate First Principles Models with Empirical Data **(Long Term)**

3. Developing a robust experimental design, consistent with ideas in reinforcement learning, to explore and exploit the reservoir during production period **(Long Term)**
   - *Prove the concept at a simulation level prior to adoption at the field level*

Acknowledgements

Nexen:

- *Mike Birdgeneau*
- *Marwan Mohamed*
- *Anton Biryukov*
- *Craig Herring*

Outside of Nexen, I've had many meaningful conversations on this topic with:

- *Aleksey Nozdryn-Plotnicki (Machine Learning Engineer @ MetaOptima)*
- *Shakeel Rajwani (Devon)*
- *Patrick Stanley & Team (ConocoPhillips)*