Summer Fellowship Report

On

Physics-Informed Cascaded Neural Surrogate for Multi-Objective Optimisation of a Fluid Catalytic Cracking Unit

Submitted by

Ishaan Ghosh

Chemical Engineering, 3rd Year

MNIT Jaipur

Under the guidance of

Prof. Prabhu Ramachandran

Aerospace Engineering Department

IIT Bombay

Mentor

Priyam Nayak

Chemical Engineering Department

IIT Bombay

August 8, 2026

Acknowledgment

I would like to express my deepest gratitude to **Prof. Prabhu Ramachandran**, for providing me with the remarkable opportunity to be a part of the FOSSEE Summer Fellowship at IIT Bombay. His vision and passion for this organisation pursuing Open source software was clearly visible in his induction speech which inspired me to work on this internship with hardwork and passion.
I am thankful to my mentor, **Priyam Nayak**, for his constant guidance, patience, and invaluable technical feedback. Checking every idea, every result with him back and forth, checking benchmarks set by his model against mine. Every bit of that interaction helped me with progression of this model and ultimately the paper.
I would also like to thank the entire FOSSEE team and the Ministry of Education (NMEICT) for fostering such a collaborative open-source environment. Finally, I extend my thanks to my institution, MNIT Jaipur, for providing me with the foundational knowledge that enabled me to undertake this fellowship.

Contents

Chapter 1

Introduction

1.1 Background of Fluid Catalytic Cracking (FCC)

Fluid Catalytic Cracking (FCC) is one of the central conversion processes in a petroleum refinery. It converts the high boiling point, high molecular weight hydrocarbon fractions of crude oil into fractions with lower molecular weights and higher commercial value, such as gasoline (naphtha), olefinic gases such as propylene and butenes, and other light products.

The process is centralized around two vessels thermally coupled together through which the catalyst circulates. The *riser*, a thin pipe like structure where the endothermic cracking reactions take place, and the *regenerator*, where the used up catalyst that is burned to remove the carbon and hence regenerate the catalyst. The heat provided to the regenerator also supplies heat to the cracking reaction.

Hence the model was based on the thermodynamics of these two units working together.

1.2 Problem Statement

Rigorous, first principles simulations of an FCC unit are highly accurate but computationally expensive and time consuming for industries.

The problem this fellowship set out to address is the gap between *speed* and *accuracy*: to build a plant model fast enough to save the losses, yet precise enough to the underlying physics that the "optimal" operating points it recommends are real solutions.

1.3 Objectives

The fellowship had two evolving goals, which are reflected in the structure of this report:

1. To build a fast neural surrogate of the FCC unit from a rigorous steady state dataset, capable of predicting product yields, characteristic temperatures, and catalyst/emission variables from a small set of operator controlled inputs.

2. To map the Pareto optimal trade offs between profit, propylene production and CO_2 emissions using multi objective optimisation (MOO).

Chapter 2

Literature Review and Evolution of the Model

2.1 Background Reading

The project began, at my mentor's suggestion, with a set of papers on FCC units and the different ways they are modelled. I looked at the papers provided by my mentor which mentioned MOO for optimising different variables altogether, perfect for this model. I spent the first week searching for papers and analysing interesting ones after confirming their credibility from Priyam sir. Ultimately, I found a paper on physics informed neural network since it was perfect solution to the extrapolation problem ML models usually face. The paper described introducing physics constraint to the model in the form of loss function.

Then I started to formulate bunch of equations that would eventually form my current loss functions. I had referred to Levenspiel and Fogler chemical engineering books from the IIT Bombay library to write down a loss function that included mass balance, heat balance and the kinetics of an FCC unit based on lumping the different sections of FCC feed into 4 or 7 groups.

2.2 How the Thinking Changed

After a week of theoretical study, I was handed the data set with 42 different columns of steady state operating points, spanning the plant's controllable inputs, its product yields, and its internal thermal, catalyst and emission variables. Helpfully, the spreadsheet itself was colour coded, with most of the important columns already marked as inputs or outputs, and a few others flagged as otherwise useful. That marking gave me a strong starting point for deciding which columns were genuine *operator handles*, which were the *targets* worth predicting, and which were needed only to close the physics.

The nine I settled on as model inputs were the true operating handles: *Feed Temperature*, *Feed Mass Flow*, *Regen Air Mass Flow*, *Dispersion Steam Mass Flow*, *Stripping Steam Mass Flow*, *Regen Air Blower Discharge Temperature*, *Cat/Oil Ratio (Riser)*, *Fresh Catalyst Make Up Rate*, and *Regen Enrich O2 Mass Flow* (Regen Enrich O2 was selected in the later stages of development when carbon balance was involved.). The prediction targets were the eight product-yield percentages (*Bottoms*, *LCO*, *Naphtha*, *Dry Gas*, *Butenes*, *Propylene*, *C3-C4 Paraffins* and *Coke*), the three characteristic temperatures (*Regen Dense Bed Temperature*, *Regen Flue Gas Temperature* and *Riser Outlet Temperature*), and the catalyst and flue-gas variables (*Coke on Spent/Regen Catalyst*, *Riser Catalyst Circulation Rate*, and the *Regen Flue Gas CO* and *CO2* streams). The Riser Outlet Temperature (ROT) marked as an input proved tricky since it could leak data to many predicted outputs like temperatures but was a crucial factor in predicting the yields. This lead me to selecting the cascaded structure, separating the inputs into 3 separate groups of temperatures, yields and other clubbed columns like CO_2 with regen catalyst circulation rate and other factors explaining the catalyst. Thus, ROT was the output for temperatures and the clubbed group, while the predicted ROT was used as an input for yields. Separately, when I came to implement the carbon balance in the final phases, I found that some of the columns I now needed had not been marked as important

at all: the *wt% Hydrogen in Coke*, *Regen Enrich O2* and the *Regen Flue Gas CO* stream alongside the *CO2*, were never prediction targets in the usual sense but are indispensable to the physics penalty, since the carbon balance cannot be written without them.

Although I wanted to include the energy balance and the kinetic lumped model, acquiring the rate of reaction for each reaction of the lumps proved troublesome, and energy balance required more input columns which will be worked on for the completion of the paper which is not a part of this report.

Chapter 3

Methodology

3.1 Data Generation and Preprocessing

The dataset provided consisted of approximately 50000 rows of data at steady state. To mimic data produced mby actual industries in a single day, this data set was pruned to around 1500 and added artificial noise to ensure model does not overfit from learning steady state values. These errors were based on error percentage and behaviour from actual industrial sensors and gauges.

Pruning. To obtain a compact, well spread training set I standardised the feature space, applied K-means clustering, and kept only the points nearest to the cluster centroids — a "nearest-to-centroid" pruning that covers the operating envelope without over representing any one region. This step stayed in the pipeline through every later revision.

Noise injection. The first version was plain uniform Gaussian noise, enough to prove the idea. It was later replaced by a per channel model (a `SENSOR_SPECS` table) that assigns every measured channel its own noise characteristic: a percentage-of-reading term paired with an absolute floor, applied multiplicatively (log-normal) on flow measurements and additively on temperatures, with the percentage and floor drawn from published accuracy figures for the corresponding real instrument.The inputs refer to this method of adding noise, whereas the outputs are separated into two groups, clean outputs and noisy outputs. Training the model with clean outputs and later validating with noisy ones benefit the model. The complete channel settings are given in Table 3.1.

Table 3.1: Per-sensor noise settings (`SENSOR_SPECS`), grouped by sensor type. The percentage-of-reading and floor values are drawn from published accuracy figures for the corresponding industrial instrument classes rather than chosen arbitrarily. *flow* channels are perturbed multiplicatively (log-normal); *temp* channels additively; *analyzer* and *yield* channels are treated as bounded percentages. Each noisy value is clipped to its physical range (e.g. $[0, 100]$ for percentages).

Channel	Type	% of reading	Floor
Flow sensors			
Feed Mass Flow	flow	0.20%	0.05
Regen Air Mass Flow	flow	0.20%	0.05
Dispersion Steam Mass Flow	flow	0.20%	0.02
Stripping Steam Mass Flow	flow	0.20%	0.02
Fresh Catalyst Make Up Rate	flow	0.50%	0.0005
Regen Enrich O2 Mass Flow	flow	0.50%	0.03
Riser Catalyst Circulation Rate	flow	3.00%	0.5
Cat/Oil Ratio (Riser)	flow	2.00%	0.05
Regen Flue Gas CO2 Emission	flow	0.10%	0.10
Temperature sensors (all channels)			
Feed Temp, Regen Air Blower, Riser Outlet, Regen Dense Bed, Regen Flue Gas, Riser Mix, Reactor Plenum	temp	0.50%	0.3
Analyzer sensors (bounded)			
Regen Flue Gas CO, Dry	analyzer	2.00%	0.005
Regen Flue Gas SOx, Dry	analyzer	3.00%	0.002
Coke on Spent Cat	analyzer	2.00%	0.005
Coke on Regen Catalyst	analyzer	2.00%	0.005
Yield / composition (bounded, %)			
wt% Hydrogen in Coke	yield	1.50%	0.05
Eight product yields (Bottoms, LCO, Naphtha, Dry Gas, Butenes, Propylene, C3–C4 Paraffins, Coke)	yield	1.00%	0.05

3.2 Cascaded Network Architecture

As mentioned earlier the cascaded structure emerged from a data leakage problem. The Riser Outlet Temperature (ROT) is a genuinely useful predictor of the product yields as they are governed by the reactor's thermal state, so it was tempting to feed ROT in as an input to the yield model. But ROT is itself an *output* of the plant with respect to the base operator inputs. If I fed the true ROT in as an input to predict yields, then at optimisation time (when ROT is not known in advance) I would either have no value to supply, or I would be leaking a quantity the model was supposed to predict. Feeding it to the yield model but treating it as a free output everywhere else was inconsistent.

The resolution was a **decoupled cascade** of three sub-networks:

- **Model B1 (Temperatures)** takes the base process inputs and predicts the three characteristic temperatures: Regen Dense Bed Temperature, Regen Flue Gas Temperature, and Riser Outlet Temperature.
- **Model B2 (Catalyst & Emissions)** takes the same base inputs and predicts coke on spent and regenerated catalyst, riser catalyst circulation rate, and the flue-gas CO and CO_2 variables.
- **Model A (Yields)** predicts the eight product yields. Crucially, it does *not* receive the true ROT; it receives the ROT *predicted by Model B1*, stitched into its input vector alongside the base inputs. At both training and optimisation time the yield model therefore consumes a self-consistent, internally-generated ROT, and no privileged output information leaks in.

This structure has the added benefit that an error in one output family cannot pollute another: temperatures, emissions and yields are produced by separate networks, joined only through the one deliberate, physically-motivated ROT link. The full data flow, including the final layer sizes of each sub-network and the physics-informed training objective, is shown in Figure 3.1.

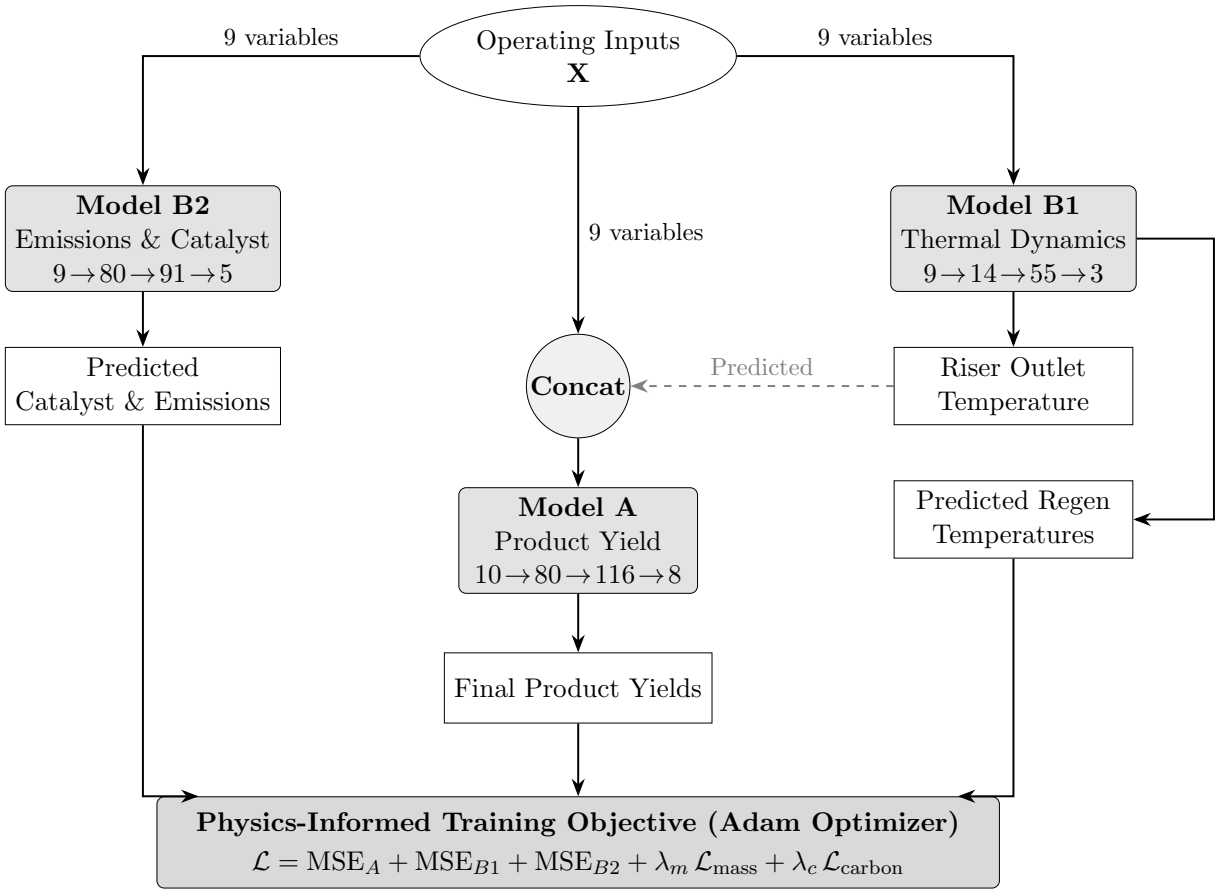

Figure 3.1: The cascaded architecture. The nine operating inputs feed Models B1 and B2; the Riser Outlet Temperature predicted by B1 is concatenated with the inputs to form the ten-dimensional input to Model A. All three sub-networks are trained jointly against a physics-informed objective combining data-fit MSE with mass- and carbon-balance penalties. Layer sizes shown are the final tuned values.

Every time the model changed shape,the sub network sizes and learning rates were re-tuned with an automated **architecture / hyper-parameter search** rather than chosen by hand, using cross-validated search over the hidden layer widths and learning rates with a cap on the number of trainable parameters to keep each sub network light.

3.3 Physics-Informed Loss Formulation

The training objective combines a data term with physics penalties. The data term is the sum of the mean squared errors of the three sub-networks in standardised space:

$$\mathcal{L}_{\mathrm{data}} = \mathrm{MSE}(\hat{y}_A, y_A) + \mathrm{MSE}(\hat{y}_{B1}, y_{B1}) + \mathrm{MSE}(\hat{y}_{B2}, y_{B2}).$$

Mass balance. The eight predicted yields, once inverse-transformed to physical percentages, must sum to 100%. A tolerance band τ is allowed and only violations beyond it are penalised, using a ReLU gate so in-tolerance predictions incur no penalty (a tolerance of $\tau = 0.05$ was used):

$$\mathcal{L}_{\mathrm{mass}} = \frac{1}{N} \sum_{i=1}^{N} \Big[\mathrm{ReLU}\big(\left| \textstyle\sum_j \hat{y}^A_{i,j} - 100 \right| - \tau \big) \Big]^2.$$

Carbon balance. This penalty was added in the second major revision of the model. Carbon entering the regenerator as coke must leave in the flue gas as CO and CO_2. The coke mass flow is

obtained from the predicted coke yield and the feed rate, and its carbon content from the hydrogen in coke fraction. The carbon leaving is assembled from the predicted CO and CO_2 via their molar mass ratios. A subtlety that cost some debugging is that the two emission variables are reported in different units CO_2 as a mass flow and CO as a percentage of the dry flue gas so each required its own conversion. The dry flue gas flow itself is obtained by taking the total (wet) flue gas and removing the water formed when the hydrogen in the coke combusts. Writing the wet flue gas as the sum of the incoming air, the enrichment oxygen and the burned coke, and the water as the coke's hydrogen scaled by the H_2O/H_2 molar mass ratio (18.015/2.016):

$$\dot{m}_{\text{wet flue}} = \dot{m}_{\text{air}} + \dot{m}_{\text{O}_2} + \dot{m}_{\text{coke}}, \qquad \dot{m}_{\text{H}_2\text{O}} = \dot{m}_{\text{coke}}\, w_{\text{H}_2}\, \tfrac{18.015}{2.016},$$

$$\dot{m}_{\text{dry flue}} = \dot{m}_{\text{wet flue}} - \dot{m}_{\text{H}_2\text{O}}.$$

The carbon balance then closes between the carbon entering as coke and the carbon leaving as CO and CO_2:

$$C_{\text{in}} = \dot{m}_{\text{coke}}\,(1 - w_{\text{H}_2}), \qquad C_{\text{out}} = \dot{m}_{\text{CO}_2}\tfrac{12}{44} + \Big(\tfrac{\text{CO\%}}{100}\dot{m}_{\text{dry flue}}\Big)\tfrac{12}{28},$$

$$\mathcal{L}_{\text{carbon}} = \frac{1}{N}\sum_{i=1}^{N}\big(C_{\text{in},i} - C_{\text{out},i}\big)^2.$$

The total loss is $\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda_{\text{mass}}\mathcal{L}_{\text{mass}} + \lambda_{\text{carbon}}\mathcal{L}_{\text{carbon}}$, with both physics weights active and set to $\lambda_{\text{mass}} = \lambda_{\text{carbon}} = 1$. The model was trained with the Adam optimiser (per-module learning rates), mini-batches, and early stopping on a held-out validation set evaluated on a fixed noisy realisation, so that the stopping criterion reflects deployment-realistic conditions.

3.4 Multi-Objective Optimization (NSGA-II)

With the PINN frozen in evaluation mode it became the physics engine inside an NSGA-II search implemented in `pymoo`. The search was posed over Feed Temperature, Regen Air Mass Flow, Dispersion Steam, Stripping Steam, Regen Air Blower Discharge Temperature, and (as noted above) the Cat/Oil ratio and fresh catalyst make-up rate, with feed throughput held fixed at **46.7 kg/hr**. ROT is deliberately *not* a decision variable: it is predicted by Model B1 and fed forward through the cascade, exactly as in training.

Objectives. The final formulation is three-objective: maximise hourly profit, maximise propylene production, and minimise CO_2 emission. In `pymoo`'s minimisation convention the two maximised objectives are negated:

$$f_1 = -\,\text{Profit}, \qquad f_2 = -\,\dot{m}_{\text{propylene}}, \qquad f_3 = \dot{m}_{\text{CO}_2}.$$

Profit. Profit is computed inside every evaluation as revenue minus operating cost. Each product stream is valued at its market price, and each yield Y_k (%) is converted to a mass flow through the fixed feed rate, $\dot{m}_k = (Y_k/100)\,\dot{m}_{\text{feed}}$:

$$\text{Revenue} = \sum_{k\in\{\text{prop, but, C3–C4, gas, LCO}\}} p_k\,\dot{m}_k,$$

with prices $p_{\text{prop}} = 1.10$, $p_{\text{but}} = 0.92$, $p_{\text{gas}} = 0.78$, $p_{\text{LCO}} = 0.75$ and $p_{\text{C3–C4}} = 0.55$ (all \$/kg). The operating cost sums the feed, catalyst make-up and steam costs with the energy cost of pre-heating the feed and the regenerator air:

$$\text{Cost} = c_{\text{feed}}\dot{m}_{\text{feed}} + c_{\text{cat}}\dot{m}_{\text{cat}} + c_{\text{steam}}\big(\dot{m}_{\text{disp}} + \dot{m}_{\text{strip}}\big) + \tfrac{c_{\text{fur}}}{1000}\,\dot{m}_{\text{feed}}C_{p,\text{feed}}\,\Delta T_{\text{feed}} + \tfrac{c_{\text{elec}}}{3600}\,\dot{m}_{\text{air}}C_{p,\text{air}}\,\Delta T_{\text{air}},$$

$$\Delta T_{\text{feed}} = \max(T_{\text{feed}} - T_{\text{ref}},\,0), \qquad \Delta T_{\text{air}} = \max(T_{\text{blower}} - T_{\text{amb}},\,0),$$

$$\text{Profit} = \text{Revenue} - \text{Cost}.$$

Here $c_{\text{feed}} = 0.500$, $c_{\text{cat}} = 6.000$ and $c_{\text{steam}} = 0.022$ \$/kg; $C_{p,\text{feed}} = 3.38$ and $C_{p,\text{air}} = 1.005$ kJ/(kg K); $T_{\text{ref}} = 250\,^{\circ}\text{C}$, $T_{\text{amb}} = 25\,^{\circ}\text{C}$; $c_{\text{fur}} = 0.005$ \$/MJ and $c_{\text{elec}} = 0.08$ \$/kWh.

Objectives. The final formulation is three-objective: maximise hourly profit, maximise propylene production, and minimise CO_2 emission. In `pymoo`'s minimisation convention the two maximised objectives are negated:

$$f_1 = -\,\text{Profit}, \qquad f_2 = -\,\dot{m}_{\text{propylene}}, \qquad f_3 = \dot{m}_{\text{CO}_2}.$$

Constraints. The search is fenced by physical limits: the Regenerator Dense Bed Temperature must stay below its metallurgical ceiling, the predicted ROT must remain inside its safe window, and profit must be non-negative. Written as the $g \le 0$ inequalities `pymoo` expects:

$$g_1 = T_{\text{dense bed}} - 720 \le 0, \qquad g_2 = 475 - T_{\text{ROT}} \le 0, \qquad g_3 = T_{\text{ROT}} - 575 \le 0, \qquad g_4 = -\,\text{Profit} \le 0,$$

with all temperatures in °C. As described in the evolution of the model (Chapter 2), an explicit constraint on flue gas CO was also introduced at one stage to stop the optimiser from "cheating" via incomplete combustion. This idea was later removed as industries do not follow hard CO constraints as discussed with Priyam sir.

Chapter 4

Results and Discussion

4.1 Surrogate Model Performance

On a held out test set the cascaded PINN reproduces the simulator to sub percent error on most yields and captures the characteristic temperatures to within a few degrees Celsius. Table 4.1 lists the mean absolute errors. Two evaluation modes are worth distinguishing, and I report both: *deployment* error (noisy inputs → clean targets), which reflects how the model behaves on realistic instrument data, and *function recovery* error (clean inputs → clean targets), which isolates how well the network learned the underlying map. The gap between them is a direct measure of the model's noise sensitivity.

Table 4.1: Test-set Mean Absolute Error (MAE) of the cascaded PINN, under deployment conditions (noisy inputs) and in function-recovery (clean inputs).

Output	**MAE (noisy in)**	**MAE (clean in)**
Product yields (% absolute)		
Bottoms	0.321	0.230
LCO	0.193	0.107
Naphtha	0.932	0.309
Dry Gas	0.331	0.085
Butenes	0.139	0.049
Propylene	0.175	0.051
C3–C4 Paraffins	0.328	0.090
Coke	0.025	0.017
Temperatures (°C)		
Regen Dense Bed Temperature	2.471	1.538
Regen Flue Gas Temperature	2.528	1.605
Riser Outlet Temperature	2.531	1.495
Catalyst & flue gas		
Coke on Spent Cat (wt%)	0.013	0.009
Coke on Regen Catalyst (wt%)	0.013	0.010
Riser Catalyst Circulation Rate (kg/s)	7.101	3.367
Regen Flue Gas CO, Dry (%)	0.021	0.013
Regen Flue Gas CO_2 Emission (kg/s)	0.046	0.035

Two observations follow from the table. First, the gap between the two columns is almost entirely attributable to *input noise*, not to a deficiency in the learned map: under clean inputs the surrogate recovers the simulator extremely well (every yield under 0.31%, all temperatures within $\sim 1.5\,^{\circ}\mathrm{C}$, and the emission channels essentially exact), and the deployment column is simply that map viewed through realistic sensor noise. This is the intended behaviour — the model was trained on freshly re-corrupted inputs precisely so that it would degrade gracefully rather than catastrophically on noisy telemetry.

Second, the yields carry the largest absolute errors of the three families, and this is structural rather than a error. Naphtha is the clearest case, its deployment MAE (0.932%) is the largest in the table, yet its clean input MAE (0.309%) is small, so most of that error is noise sensitivity on the single largest magnitude yield rather than a poorly learned response. The MAE when converted to MAPE more or less turns out to be similar for all yields (%) .More fundamentally, Model A is the only sub network that consumes a *predicted* input i.e., ROT from Model B1, so its error floor is B1's ROT error (about 2.5 °C under noise) propagated forward, plus its own. That propagation is the deliberate price paid for eliminating the ROT data leakage, and even so the light olefins that matter most for the optimisation (propylene 0.175%, butenes 0.139%) remain tightly predicted.

4.2 Multi-Objective Optimisation Results

The three objective NSGA-II search (maximise profit, maximise propylene, minimise CO_2) with a large population and long generation budget returned a dense, well-spread Pareto set of 1000 solutions. Because the front lives in three objective dimensions, clearest way of reading the plot a 2-D plot showing profit vs CO_2 with colour coded propylene yield.

The three-way front (operator decision map). Figure 4.1 shows across the front, profit ranges from break even up to about \$26,700/hr, CO_2 from roughly 12,200 to 26,300 kg/hr, and propylene from about 3,170 to 11,390 kg/hr.

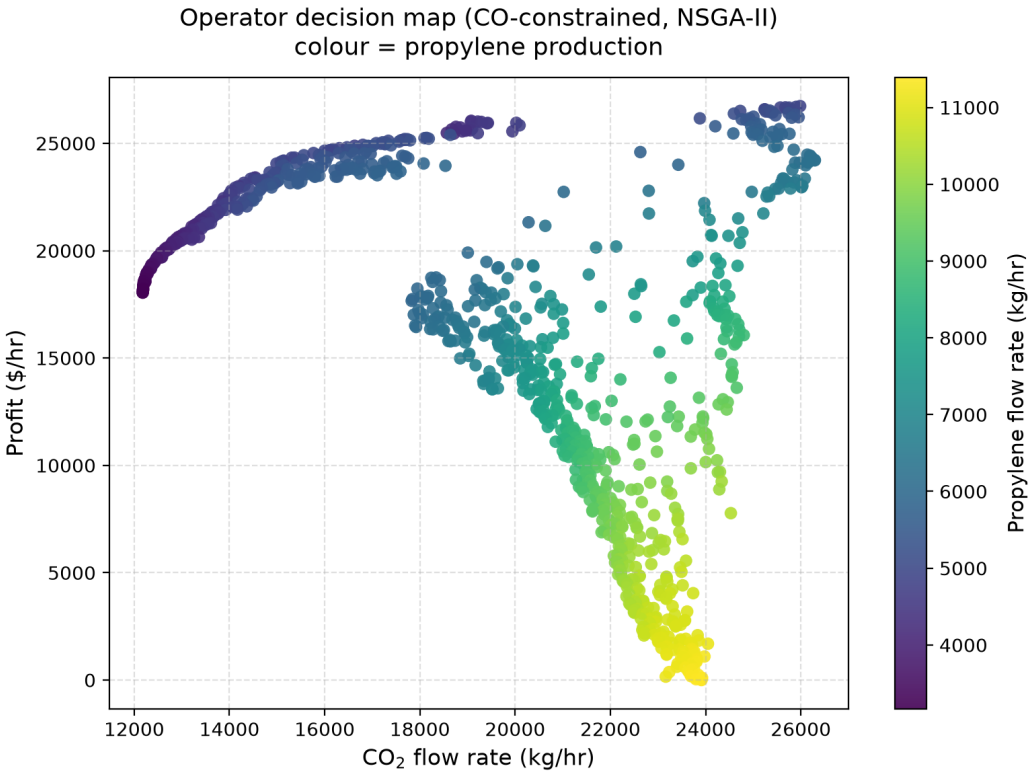

Figure 4.1: Three way operator decision map: the Pareto set projected onto profit vs CO_2, coloured by propylene production. The upper left arc is the high profit / low emission / low propylene regime; the lower right plume is the maximum propylene regime, reached only at high CO_2 and near zero profit.

The map shows tight upper left arc is the profitable regime, high profit at comparatively low CO_2, but low propylene (dark points). Moving toward the lower right, propylene production rises steadily (the colour warms to yellow) but only by accepting progressively higher CO_2 and sharply lower profit. The extreme propylene maximising solutions sit at essentially zero profit and the highest emissions producing highest propylene flow seen.

A thought on the maximum propylene corner. Inspecting the decision variables of the lowest profit / highest propylene solutions shows that every parameter is pushed to the upper or lower floors to obtain this result (feed temperature, air, blower temperature, Cat/Oil and make

up all at their maxima, the steams at their minima). This is the optimiser's way to guarantee maximum propylene production by pushing the process to it's limits. The profitable upper left arc and the knee of the trade off are trustworthy, the extreme tip of the propylene plume is better regarded as the edge of the model's validity than as a concrete operating recommendation.

The two way front (profit vs CO_2) and benchmark. Figure 4.2 shows the same front reduced to the two economic objectives, profit against CO_2, which makes the shape of the core trade off explicit. The most profitable configurations cluster near the top at roughly \$26,700/hr, and the curve has a pronounced *knee*, the solution farthest from the line joining its two extremes, i.e. the point of diminishing returns where buying more profit begins to cost disproportionately more CO_2. That knee sits at approximately **\$25,941/hr and 20,682 kg/hr CO_2**, and is the most stable single operating recommendation, it captures most of the achievable profit while staying well inside the emission envelope.

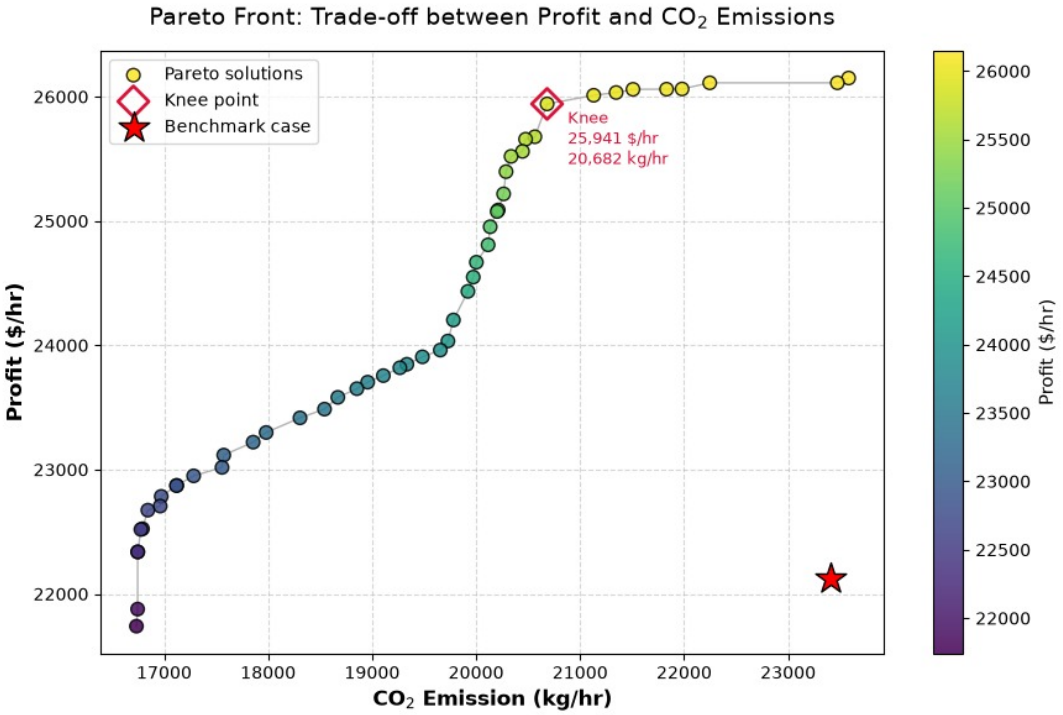

Figure 4.2: Two way profit vs CO_2 projection of the final front, with the knee point (diamond) and a benchmark reference case (star). The benchmark is dominated on all fronts.

The same plot places the optimiser's front against a benchmark reference case (the best operating point available from my mentor), shown as the red star at roughly **22,128\$/hr** and **23,408 kg/hr** CO_2. The key result is that the benchmark is *dominated* by the front, the knee improves on it in *both* objectives at once, delivering about **3,800 \$/hr** more profit (roughly 17%) while emitting about **2,700 kg/hr** less CO_2 (roughly 12%). A refiner would not have to trade one against the other to beat the reference operation; the optimiser finds settings that are simply better on both counts, which is the strongest form of result a trade off study can produce.

4.3 Real World Applications

Where this model shines is the fact that it can produce an entire frontier of solutions for the industry to select from in a matter of minutes using a rapid but accurate ANN model and using Multi Objective Optimisation (MOO).

Even if the operators were to make some changes in the constraints or focus on more objectives, the model script can be adjusted quickly to produce the desired results in a few minutes.

Chapter 5

Internship Experience at FOSSEE

5.1 Work Culture and Environment

Working within the FOSSEE ecosystem at IIT Bombay was a dream come true. Being about to roam around the campus, explore the chemical engineering department, working in the IIT Bombay library with borrowed books, every bit of it helped me work harder and efficiently with a happy mental state. I wanted to stick to being professional about expressing gratitude about this opportunity but it felt better expressing about working in every JEE aspirant's dream college. Prof. Ramachandran's words stuck with me and will always push me to work with genuine interest and passion, wherever I go.

5.2 Challenges Faced

Getting things wrong ultimately led to challenges and overcoming those taught me a lot:

- **The ROT data leakage problem** was the conceptual turning point that produced the cascade. Realising that feeding a predicted output back as an input is only legitimate if the model generates that input itself, consistently, at both train and optimisation time, took a while to see clearly.
- **The incomplete combustion exploit** taught me that an optimiser will find every loophole a surrogate leaves open. Fencing it with a CO constraint was a partial fix, the real fix was making the model itself carbon aware.
- **The low-CO_2 catch** from my mentor was a lesson in not trusting a plot just because it looks clean. It drove the carbon balance penalty, and with it the discovery of the **CO-vs-CO_2 unit mismatch** (percentage vs mass flow) that needed two separate conversions and a wet-to-dry flue gas derivation to resolve.

5.3 Learning Outcomes and Skill Development

Over the fellowship I moved from writing simple `nn.Sequential` models to designing custom, object oriented PyTorch, a cascaded multi network model with a bespoke physics-informed loss carrying its own scaler buffers and unit conversions. I learned how genetic algorithms such as NSGA-II actually explore a many objective space, how to pose a constrained multi objective problem in `pymoo`, and how to run a principled architecture search instead of guessing network sizes. Most of all, the project lived at the interface of Chemical Engineering and Computer Science, it forced me to translate conservation laws I knew from process calculations into differentiable penalties, and to keep asking whether a numerically good model was actually physically true.

5.3.1 From Paper to Code: My Working Method

I should be honest about how I worked, because it reflects both a limitation and a principle I tried to hold to. I am not yet a fluent programmer, I cannot always sit down and write a model from scratch without looking things up, and a good deal of the syntax in this project came from documentation and online references rather than from memory. What I could do was think the problem through *on paper*. My comfort was with the engineering and the mathematics, so my workflow was to derive an idea by hand, get the physics and the algebra completely right, understand every step of it, and only then work out how to express it in Python.

Translating these hand derived ideas into working code is also how my programming slowly improved over the fellowship. I would not claim to have become a fluent coder in a single summer, I still reach for references often, but I grew more comfortable writing custom PyTorch classes and `pymoo` problem definitions than I was at the start, and I became better at recognising when a piece of code did not faithfully express the idea behind it. Throughout the period of this internship, I am glad I was pushed by my mentor Priyam sir and a friend of mine who just wanted me to have fun and code.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This fellowship produced a fast, physics-informed cascaded neural surrogate for an FCC unit and used it as the evaluation engine of an NSGA-II multi objective optimiser. Developed over several revisions, each triggered by a concrete failure of the previous one The final model reproduces the results of the simulator to sub percent error on most yields and a few degrees on temperatures, while embedding mass and carbon balance constraints that keep its predictions physically consistent. Coupled with an economic model and metallurgical/combustion constraints, it maps a genuine three way trade-off between profit, propylene and CO_2, and with appropriate caution about the edges of its validity identifies the operating regimes a refiner would actually care about.

6.2 Future Scope

Several threads would strengthen the framework:

- **A rigorous heat balance.** The model currently enforces mass and carbon conservation and captures the riser-regenerator thermal coupling only implicitly through the ROT cascade. An explicit energy-balance penalty would further constrain the optimiser away from thermally inconsistent points; it was the one balance the present dataset could not close, for want of the enthalpy data it requires. I have already derived the formulation by hand and intend to pursue it, together with the kinetics based loss, as a dedicated follow-up paper with my mentor.
- **Feedstock variability.** The present study fixes feed throughput and composition, treating feed properties as additional (possibly uncertain) inputs would yield robust operating envelopes rather than single feedstock optima.

References

1 K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,” IEEE Transactions on Evolutionary Computation, 2002.

2 Standard practices for industrial noise modeling in telemetry data.

3 O. Levenspiel, *Chemical Reaction Engineering*, 3rd ed., John Wiley & Sons.

4 H. S. Fogler, *Elements of Chemical Reaction Engineering*, Prentice Hall.