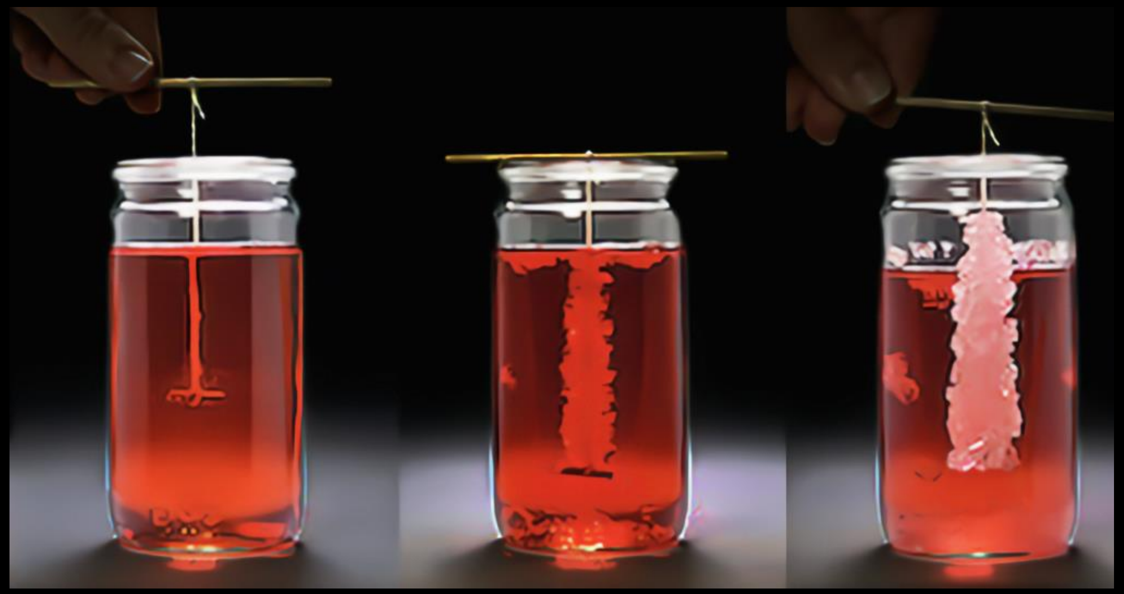
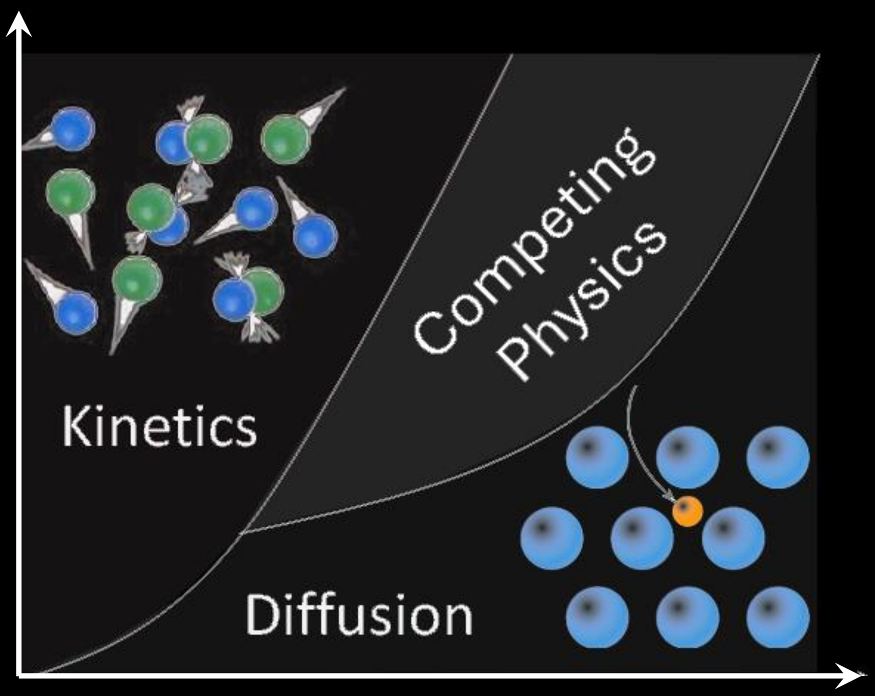
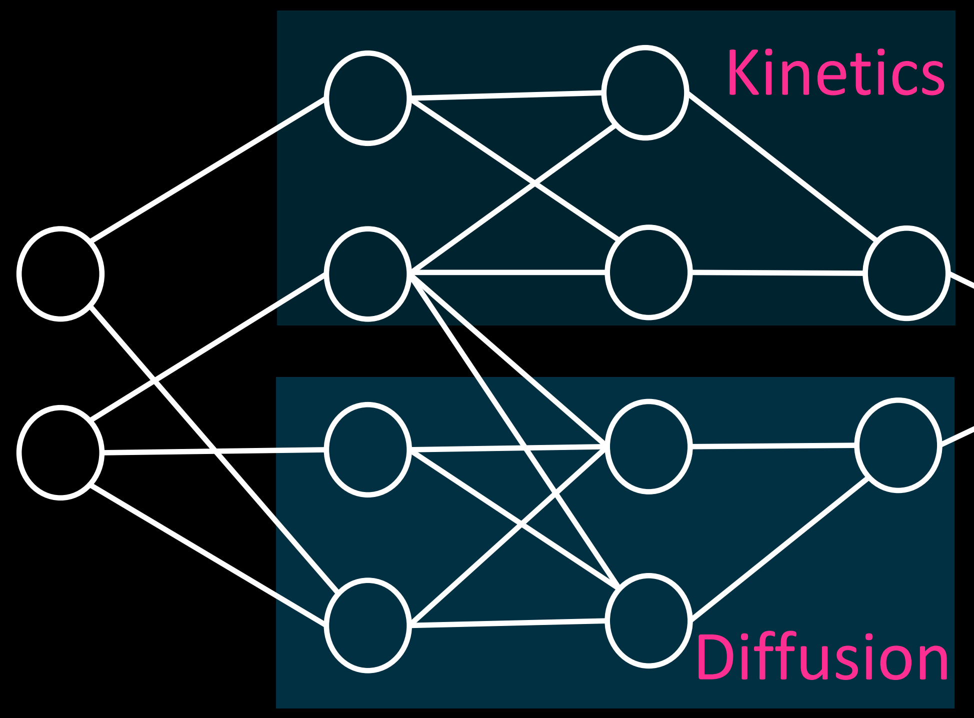
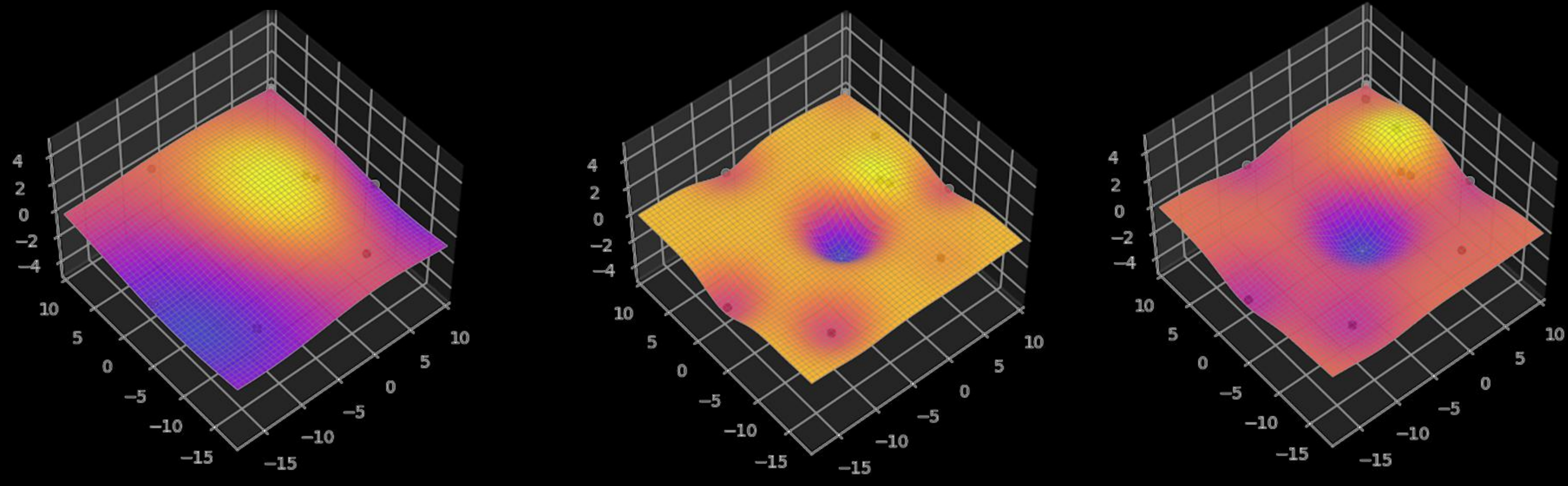
# PHYSICS INFORMED MACHINE LEARNING
# 3D PRINTING





**Navid Zobeiry, Associate Professor**
**Email: navidz@uw.edu**
**Materials Science & Engineering**
**https://composites.uw.edu/AI/**

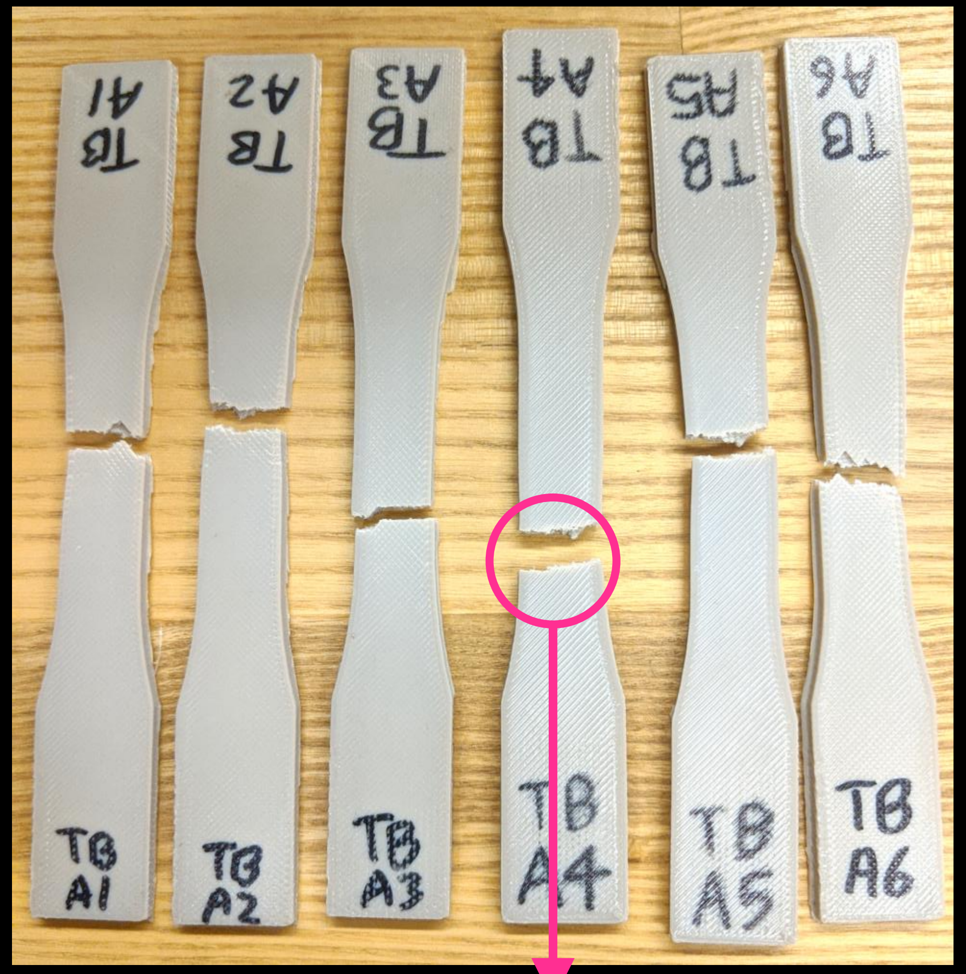
UNIVERSITY *of* WASHINGTON

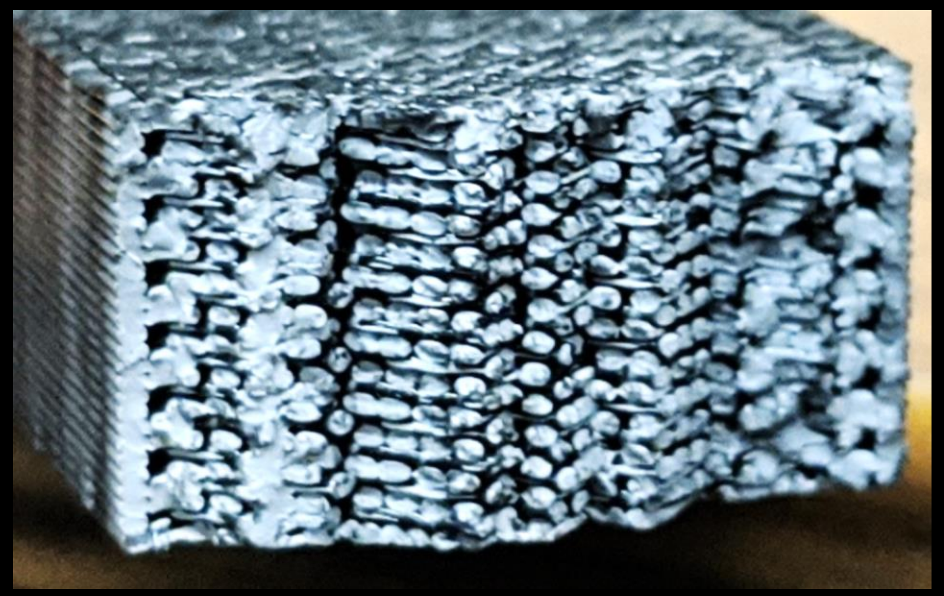
# Background - FDM (Fused Deposition Modeling)

> In FDM 3D printing, initial parameter settings significantly impact the final product's properties

Nozzle Temperature
Chamber Temperature
Platform Temperature
Nozzle Diameter
Printing Speed
Layer Thickness
Infill Percentage
Infill Orientation
…


Strength



# Problem Statement

3Dprint/3Dprint_data.csv

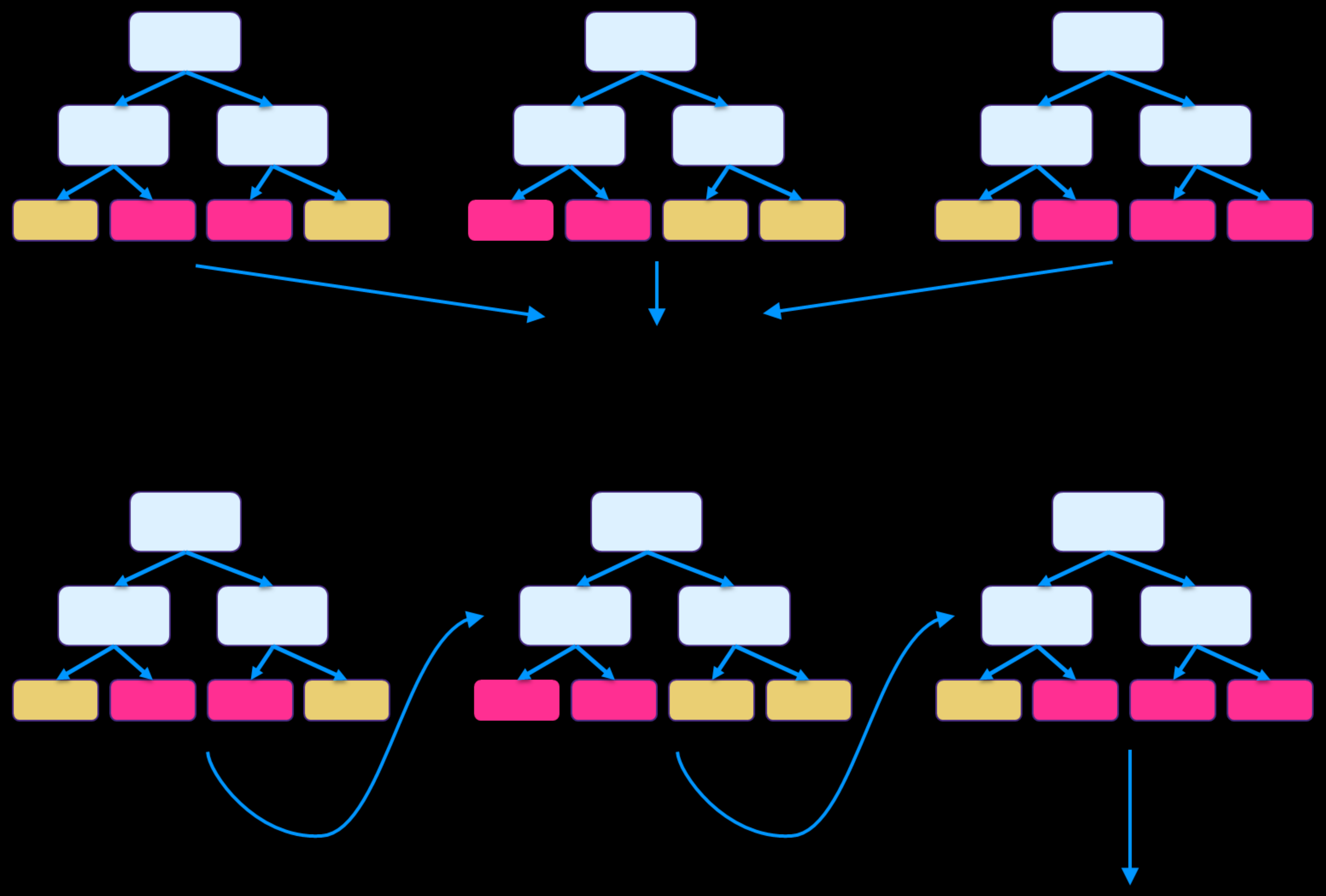
- 39 experimental data from Wang et al. (2019) on PEEK (Polyether ether ketone)
- 4 input features
  - Nozzle Temperature (380-440 °C)
  - Printing speed (17-26 mm/sec)
  - Layer thickness (0.1-0.5 mm)
  - Nozzle diameter (0.4-0.8 mm)
- 4 constants
  - Chamber temperature (~180 °C)
  - Platform temperature (280 °C)
  - Infill percentage (100%)
  - Infill Orientation (±45°)
- 1 target variable
  - Tensile strength (MPa)

https://doi.org/10.1016/j.jmatprotec.2019.03.016

| Nozzle Temp. | Printing speed | Layer thickness | Nozzle diameter | Chamber Temp. | Platform Temp. | Tensile Strength |
|---|---|---|---|---|---|---|
| 380 | 20 | 0.2 | 0.4 | 180 | 280 | 66 |
| 400 | 20 | 0.2 | 0.4 | 180 | 280 | 67.5 |
| 420 | 20 | 0.2 | 0.4 | 180 | 280 | 71 |
| 440 | 20 | 0.2 | 0.4 | 180 | 280 | 72.5 |
| 400 | 17 | 0.2 | 0.4 | 180 | 280 | 67 |
| 400 | 20 | 0.2 | 0.4 | 180 | 280 | 68 |
| 400 | 23 | 0.2 | 0.4 | 180 | 280 | 66 |
| 400 | 26 | 0.2 | 0.4 | 180 | 280 | 56.5 |
| 400 | 20 | 0.1 | 0.4 | 180 | 280 | 65.5 |
| 400 | 20 | 0.15 | 0.4 | 180 | 280 | 63 |
| 400 | 20 | 0.2 | 0.4 | 180 | 280 | 63 |
| 400 | 20 | 0.25 | 0.4 | 180 | 280 | 68 |
| 380 | 20 | 0.3 | 0.6 | 180 | 280 | 43 |
| 400 | 20 | 0.3 | 0.6 | 180 | 280 | 56 |
| 420 | 20 | 0.3 | 0.6 | 180 | 280 | 69 |
| 440 | 20 | 0.3 | 0.6 | 180 | 280 | 72.5 |
| 400 | 17 | 0.3 | 0.6 | 180 | 280 | 55 |
| 400 | 20 | 0.3 | 0.6 | 180 | 280 | 56.5 |
| 400 | 23 | 0.3 | 0.6 | 180 | 280 | 52 |
| 400 | 26 | 0.3 | 0.6 | 180 | 280 | 46 |
| 400 | 20 | 0.25 | 0.6 | 180 | 280 | 55 |
| 400 | 20 | 0.3 | 0.6 | 180 | 280 | 58 |
| 400 | 20 | 0.35 | 0.6 | 180 | 280 | 56 |
| 400 | 20 | 0.4 | 0.6 | 180 | 280 | 38 |
| 380 | 20 | 0.4 | 0.8 | 180 | 280 | 52 |
| 400 | 20 | 0.4 | 0.8 | 180 | 280 | 51 |
| 420 | 20 | 0.4 | 0.8 | 180 | 280 | 58 |
| 440 | 20 | 0.4 | 0.8 | 180 | 280 | 69 |
| 400 | 17 | 0.4 | 0.8 | 180 | 280 | 56 |
| 400 | 20 | 0.4 | 0.8 | 180 | 280 | 50 |
| 400 | 23 | 0.4 | 0.8 | 180 | 280 | 34 |
| 400 | 26 | 0.4 | 0.8 | 180 | 280 | 36 |
| 400 | 20 | 0.35 | 0.8 | 180 | 280 | 60 |
| 400 | 20 | 0.4 | 0.8 | 180 | 280 | 53 |
| 400 | 20 | 0.45 | 0.8 | 180 | 280 | 51 |
| 400 | 20 | 0.5 | 0.8 | 180 | 280 | 52.5 |

# Traditional Machine Learning

- We will compare five different machine learning methods:
  - **Linear Regression**
  - **Decision Tree**
  - *Ensemble Methods*
    - *Bagging*
      - *Random Forest*
    - *Boosting*
      - *Gradient Boosting Machine (GBM)*
      - *XGB Regressor (XGBoost)*



# Python Implementation

## > Import libraries

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.metrics import r2_score
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
from xgboost import XGBRegressor
```

## > Load and prepare data

```python
data_file = '3Dprint_data.csv'
X_columns = ['nozzle_temp', 'printing_speed', 'layer_thickness', 'nozzle_diameter']
Y_columns = ['tensile_strength']

df = pd.read_csv(data_file)
X = df[X_columns]
Y = df[Y_columns].values.ravel()

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=42)
```

3Dprint/3Dprint.py

# Python Implementation

> RandomizedSearchCV for Hyperparameter tuning, $R^2$ for evaluation:

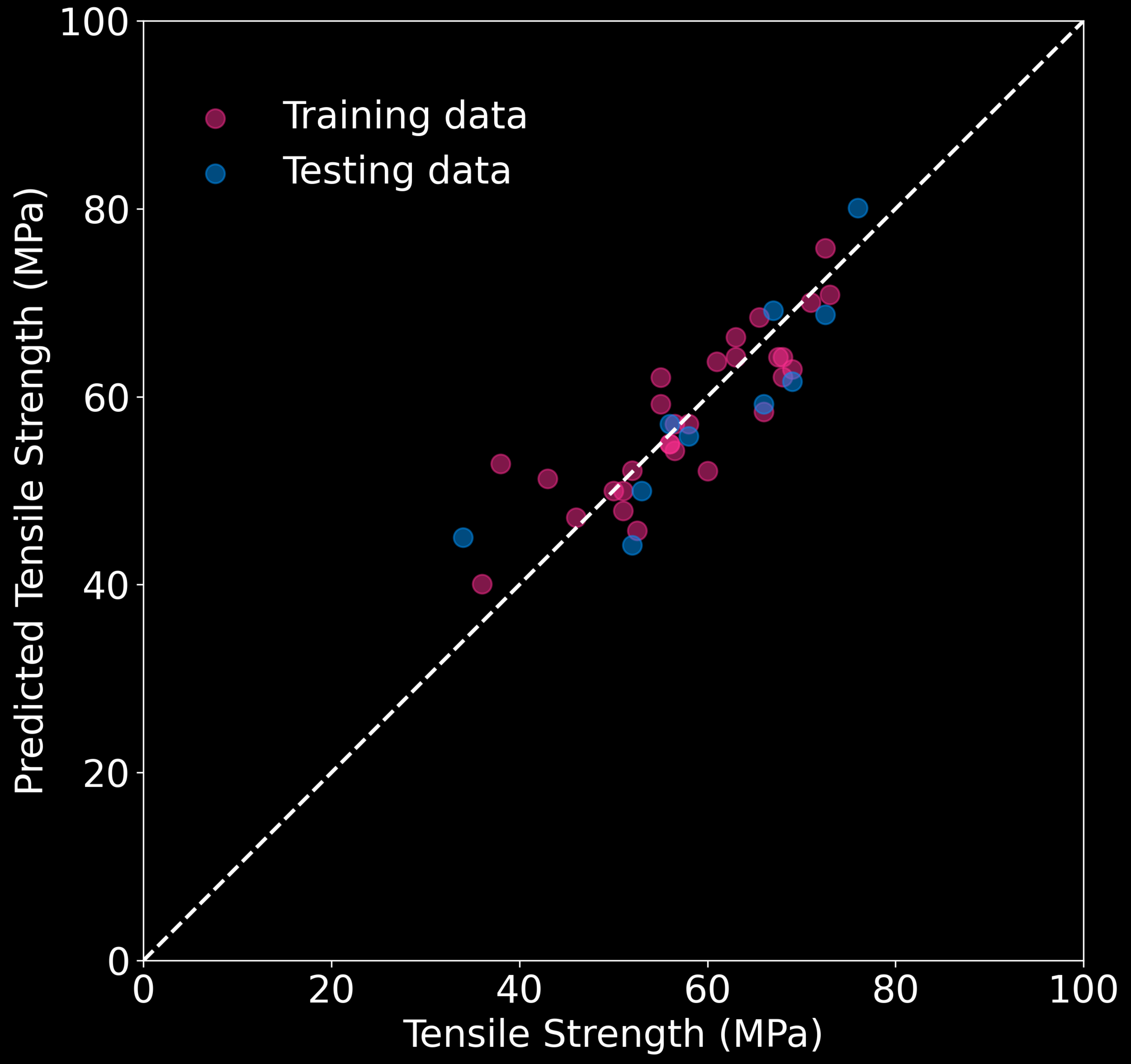
```
model_configs = [
    {
        "name": "Random Forest",
        "model": RandomForestRegressor(random_state=42),
        "params": {
            "n_estimators": [5, 10, 25, 50],
            "max_depth": [None, 5, 10],
            "min_samples_split": [2, 5],
            "min_samples_leaf": [1, 2],
        }
    },
```

```
for config in model_configs:
    rnd_search_cv = RandomizedSearchCV(config["model"], config["params"],
                                        n_iter=30, cv=5, scoring='r2', random_state=42, n_jobs=-1)
    rnd_search_cv.fit(X_train, Y_train)
    model = rnd_search_cv.best_estimator_

    # Predict on training and testing data
    Y_pred_train = model.predict(X_train)
    Y_pred_test = model.predict(X_test)

    # Calculate R2 for training and testing data
    r2_train = round(r2_score(Y_train, Y_pred_train) * 100, 1)
    r2_test = round(r2_score(Y_test, Y_pred_test) * 100, 1)
    results.append({"Model": config["name"], "R2_test": r2_test, "R2_train": r2_train})

    if r2_test > best_score:
        best_score = r2_test
        best_model = model
```

# Traditional ML: Training and Evaluation

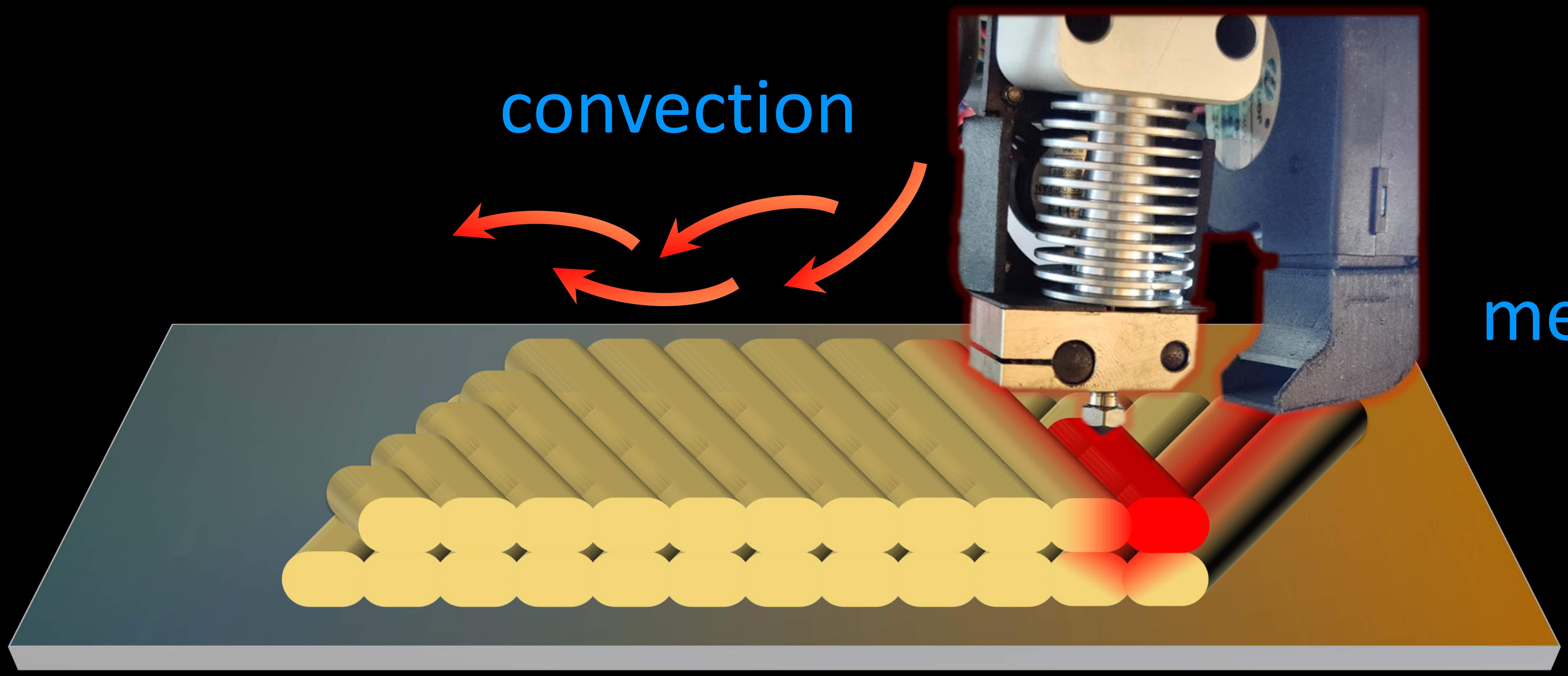
> **Best Model:**
  - **Linear Regression**
  - **Train Score: 74%**
  - **Test Score: 76%**

| Model | $R^2$ test | $R^2$ train |
|---|---|---|
| Linear Reg. | 76% | 74% |
| Decision Tree | 67% | 99% |
| XGBoost | 65% | 97% |
| GBM | 64% | 99% |
| Random For. | 61% | 94% |



## Background: Underlying Physics

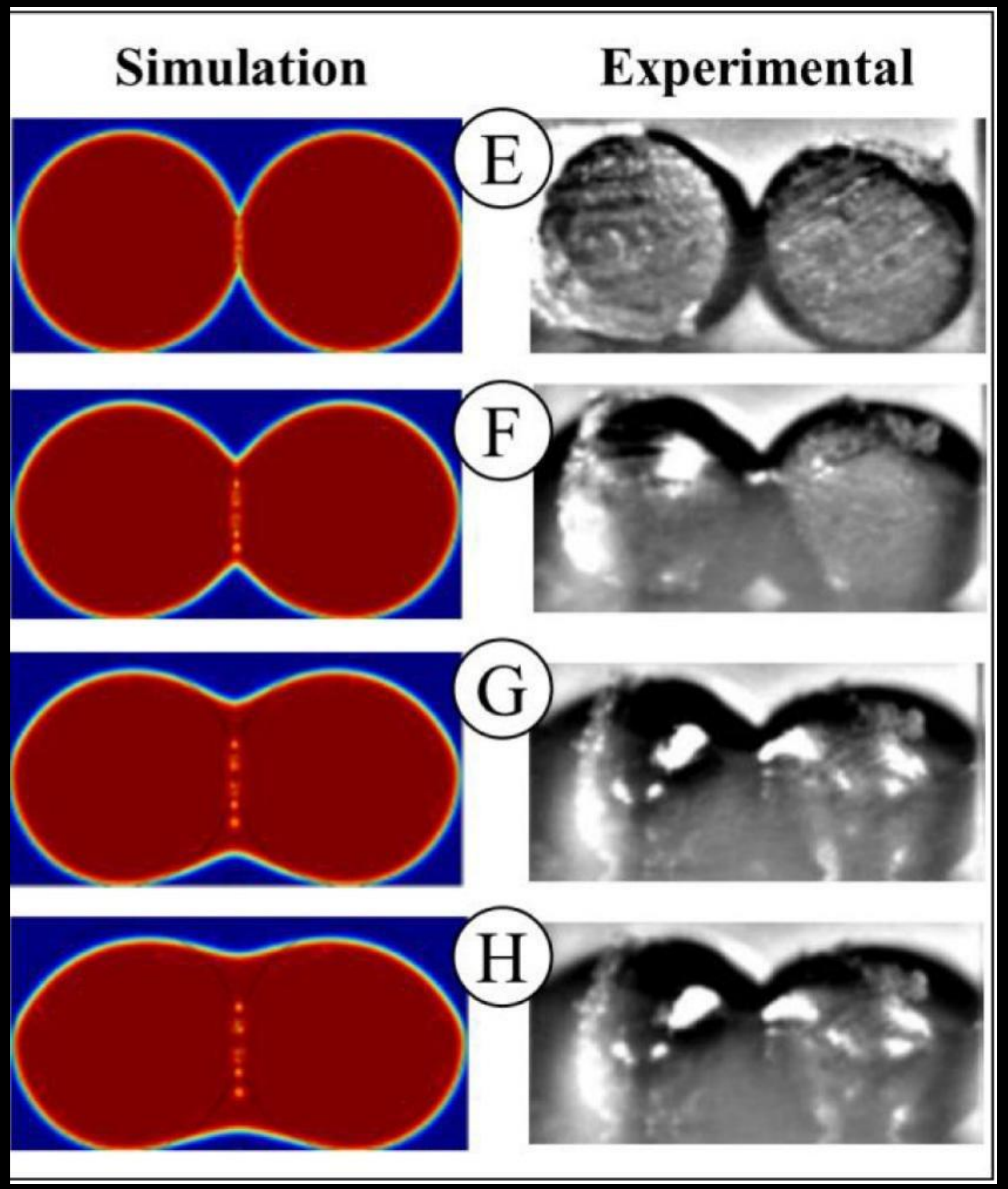
> During printing, PEEK melts around 343°C. As it cools down below this temperature, the printed filaments solidify due to heat transfer with the cooler layers and chamber air.



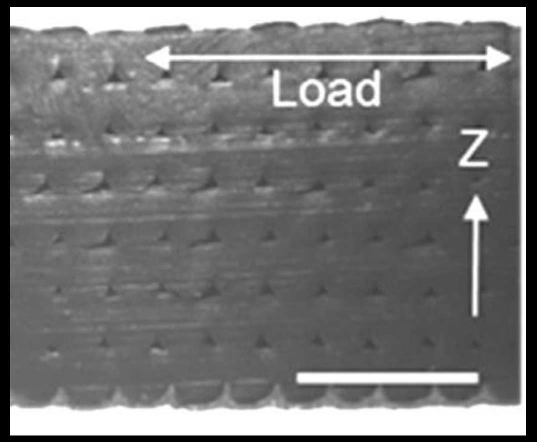
# Background: Underlying Physics

> In its melted state, the polymer flows due to surface tension, filling gaps. If it solidifies too quickly, large voids may form, reducing strength.

Surface tension-driven flow

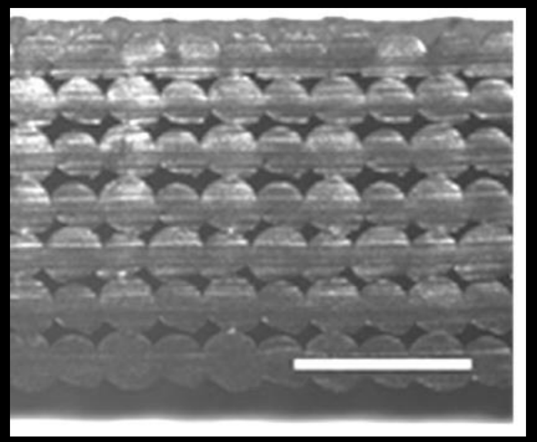



Small voids




High strength

Large voids



Low strength

## Background: Underlying Physics

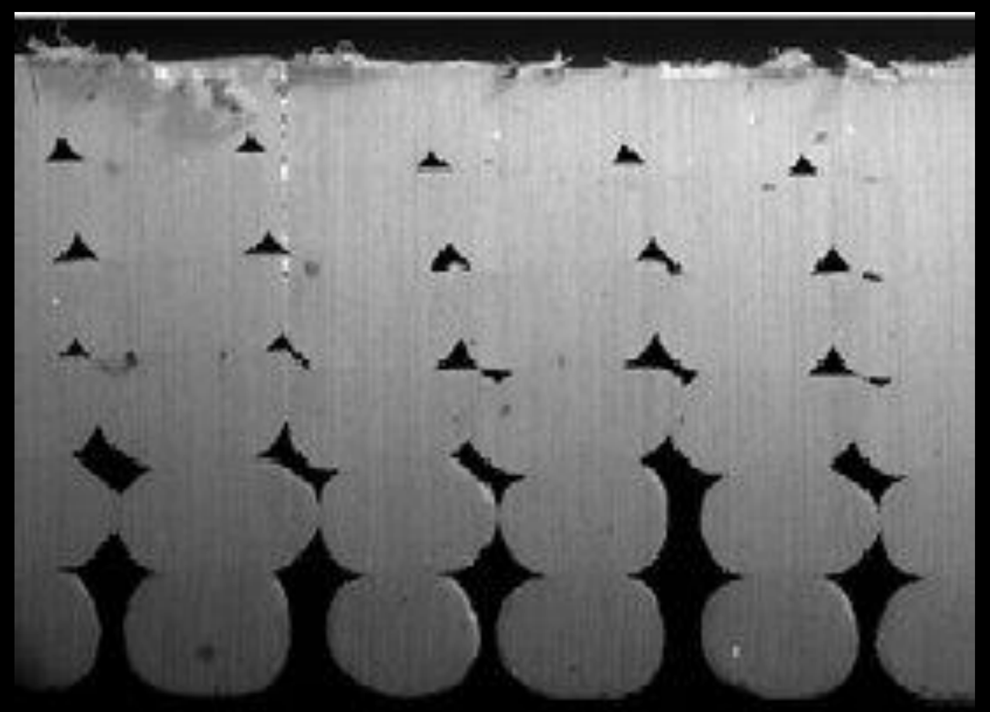
> We can approximate:

$$Strength \propto \frac{1}{Void} \propto \frac{\gamma . t}{\mu(t)}$$

$\gamma$: surface tension
$t$: time
$\mu$: viscosity

> Assuming constant surface tension:

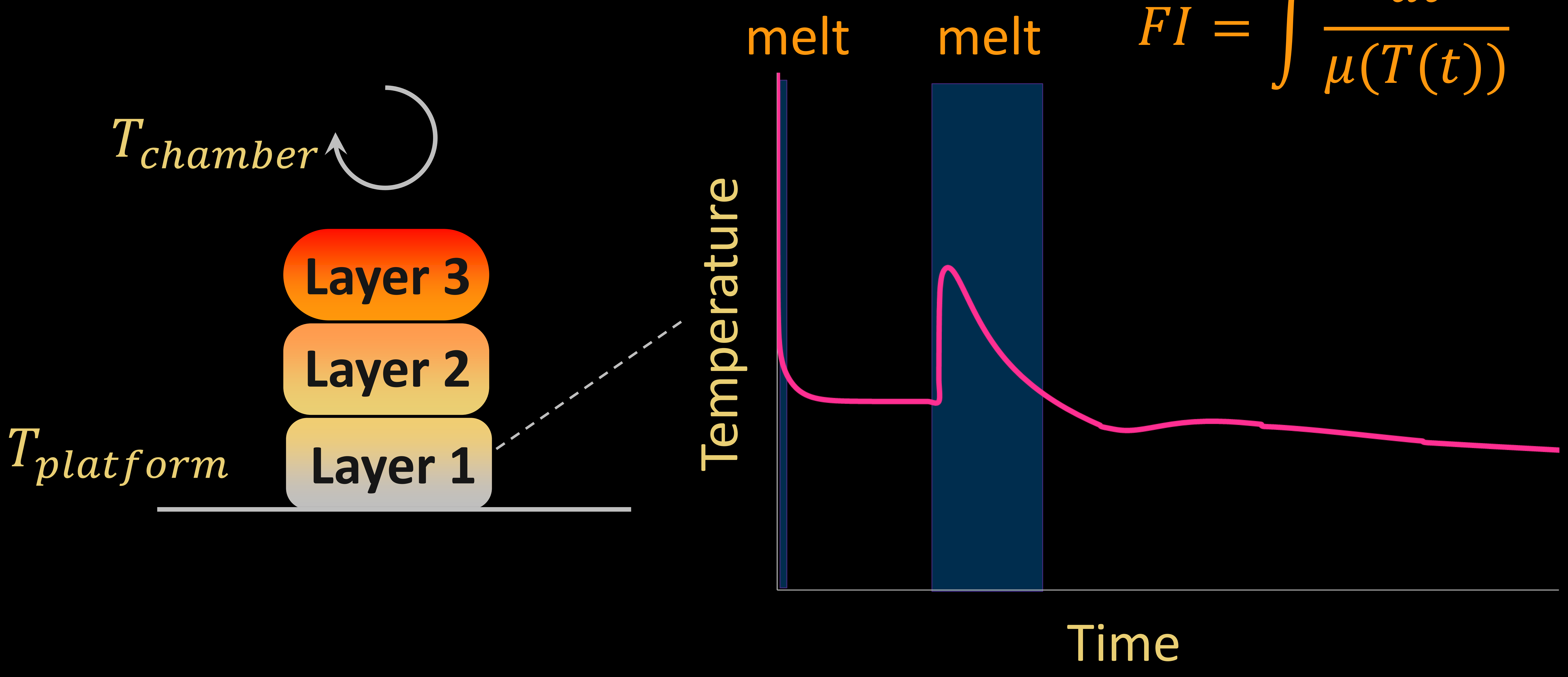
$$Strength \propto Flow\ Index = \int \frac{dt}{\mu(t)}$$



# Flow Index (FI) Calculation

> Flow Index at mid-thickness was estimated for each experimental data point using a simple Finite Element model and added to the database.

$$FI = \int \frac{dt}{\mu(T(t))}$$



# Physics-informed ML

- 39 experimental data from Wang et al. (2019) on PEEK (Polyether ether ketone)
- 5 inputs
  - Nozzle Temperature (380-440 °C)
  - Printing speed (17-26 mm/sec)
  - Layer thickness (0.1-0.5 mm)
  - Nozzle diameter (0.4-0.8 mm)
  - Average Flow Index
- 1 output
  - Tensile strength (MPa)

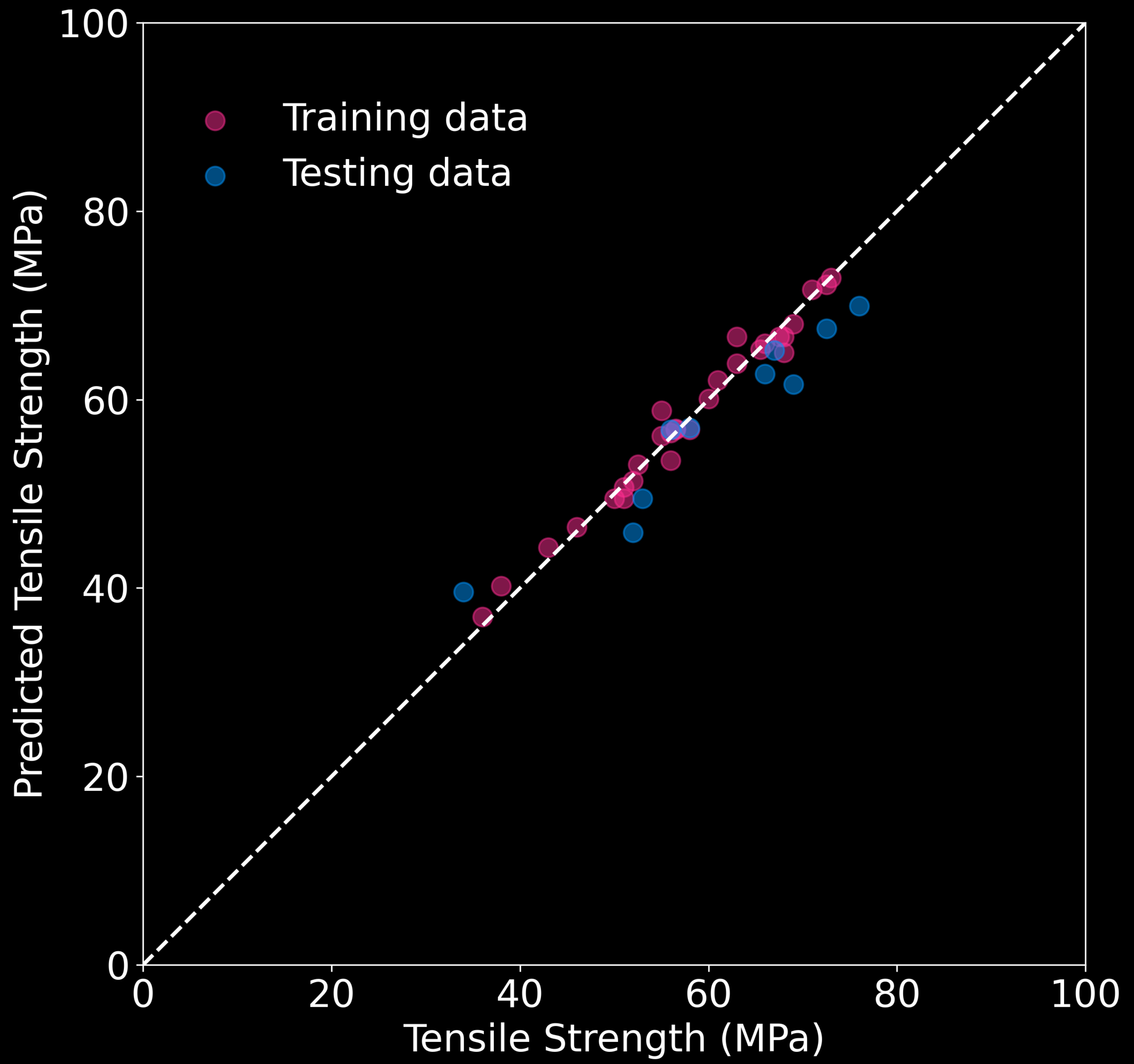
| Nozzle Temp. | Printing speed | Layer thickness | Nozzle diameter | Flow Index | Tensile strength |
|---|---|---|---|---|---|
| 380 | 20 | 0.2 | 0.4 | 0.809 | 66 |
| 400 | 20 | 0.2 | 0.4 | 0.827 | 67.5 |
| 420 | 20 | 0.2 | 0.4 | 0.965 | 71 |
| 440 | 20 | 0.2 | 0.4 | 1.343 | 72.5 |
| 400 | 17 | 0.2 | 0.4 | 0.841 | 67 |
| 400 | 20 | 0.2 | 0.4 | 0.827 | 68 |
| 400 | 23 | 0.2 | 0.4 | 0.821 | 66 |
| 400 | 26 | 0.2 | 0.4 | 0.808 | 56.5 |
| 400 | 20 | 0.1 | 0.4 | 0.877 | 65.5 |
| 400 | 20 | 0.15 | 0.4 | 0.845 | 63 |
| 400 | 20 | 0.2 | 0.4 | 0.827 | 63 |
| 400 | 20 | 0.25 | 0.4 | 0.783 | 68 |
| 380 | 20 | 0.3 | 0.6 | 0.644 | 43 |
| 400 | 20 | 0.3 | 0.6 | 0.679 | 56 |
| 420 | 20 | 0.3 | 0.6 | 0.721 | 69 |
| 440 | 20 | 0.3 | 0.6 | 0.777 | 72.5 |
| 400 | 17 | 0.3 | 0.6 | 0.707 | 55 |
| 400 | 20 | 0.3 | 0.6 | 0.679 | 56.5 |
| 400 | 23 | 0.3 | 0.6 | 0.670 | 52 |
| 400 | 26 | 0.3 | 0.6 | 0.645 | 46 |
| 400 | 20 | 0.25 | 0.6 | 0.783 | 55 |
| 400 | 20 | 0.3 | 0.6 | 0.679 | 58 |
| 400 | 20 | 0.35 | 0.6 | 0.631 | 56 |
| 400 | 20 | 0.4 | 0.6 | 0.554 | 38 |
| 380 | 20 | 0.4 | 0.8 | 0.543 | 52 |
| 400 | 20 | 0.4 | 0.8 | 0.554 | 51 |
| 420 | 20 | 0.4 | 0.8 | 0.616 | 58 |
| 440 | 20 | 0.4 | 0.8 | 0.709 | 69 |
| 400 | 17 | 0.4 | 0.8 | 0.573 | 56 |
| 400 | 20 | 0.4 | 0.8 | 0.554 | 50 |
| 400 | 23 | 0.4 | 0.8 | 0.526 | 34 |
| 400 | 26 | 0.4 | 0.8 | 0.519 | 36 |
| 400 | 20 | 0.35 | 0.8 | 0.631 | 60 |
| 400 | 20 | 0.4 | 0.8 | 0.554 | 53 |
| 400 | 20 | 0.45 | 0.8 | 0.493 | 51 |
| 400 | 20 | 0.5 | 0.8 | 0.565 | 52.5 |
| 440 | 20 | 0.1 | 0.4 | 1.194 | 76 |
| 440 | 20 | 0.25 | 0.6 | 1.300 | 73 |
| 440 | 20 | 0.35 | 0.8 | 0.688 | 61 |

# Physics-Informed ML: Training and Evaluation

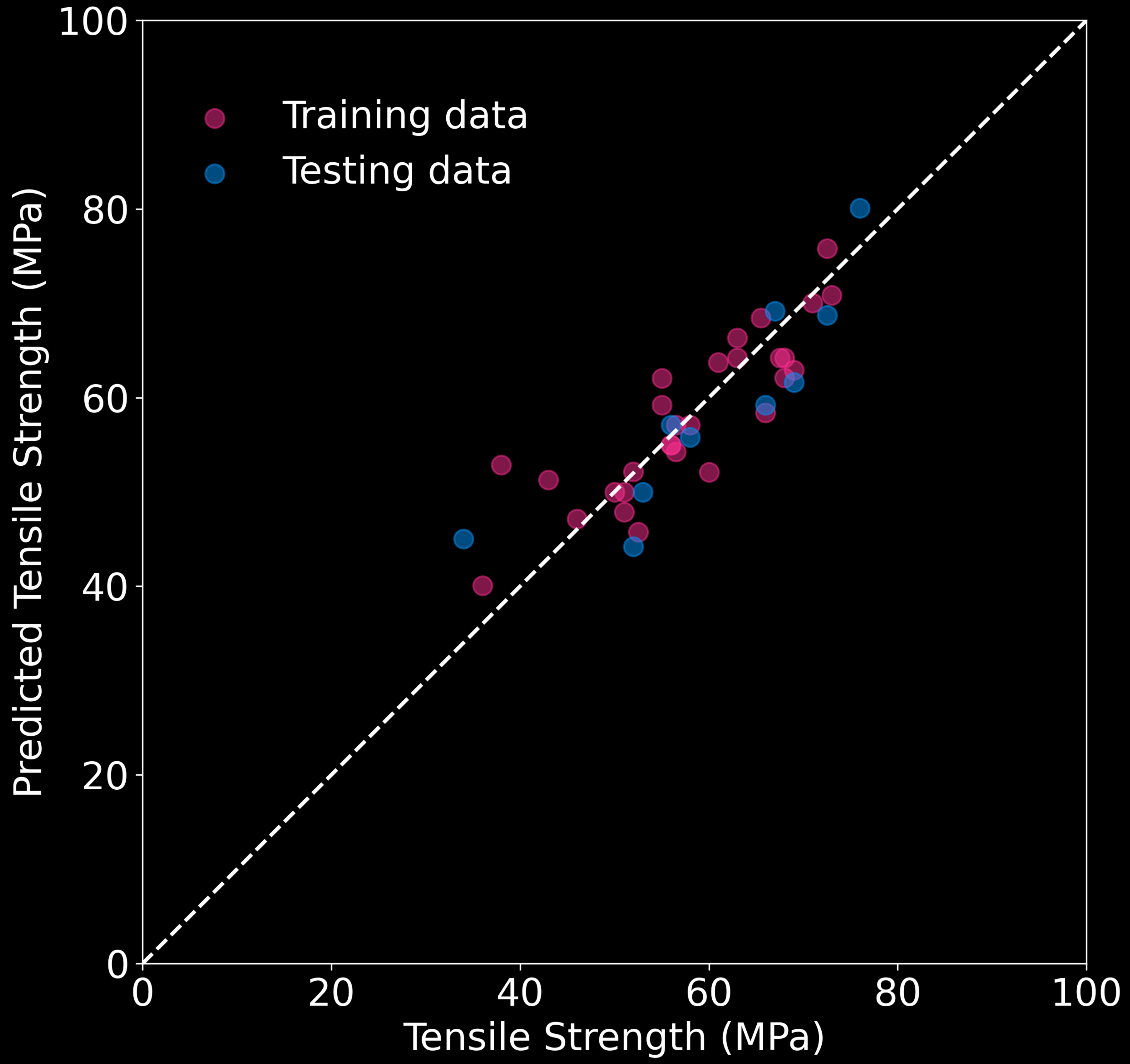
> **Best Model:**

- **XGBoost**
- **Train Score: 98%**
- **Test Score: 85%**

| Model | $R^2$ test | $R^2$ train |
|---|---|---|
| XGBoost | 85% | 98% |
| Linear Reg. | 75% | 75% |
| GBM | 73% | 99% |
| Decision Tree | 73% | 99% |
| Random For. | 71% | 84% |



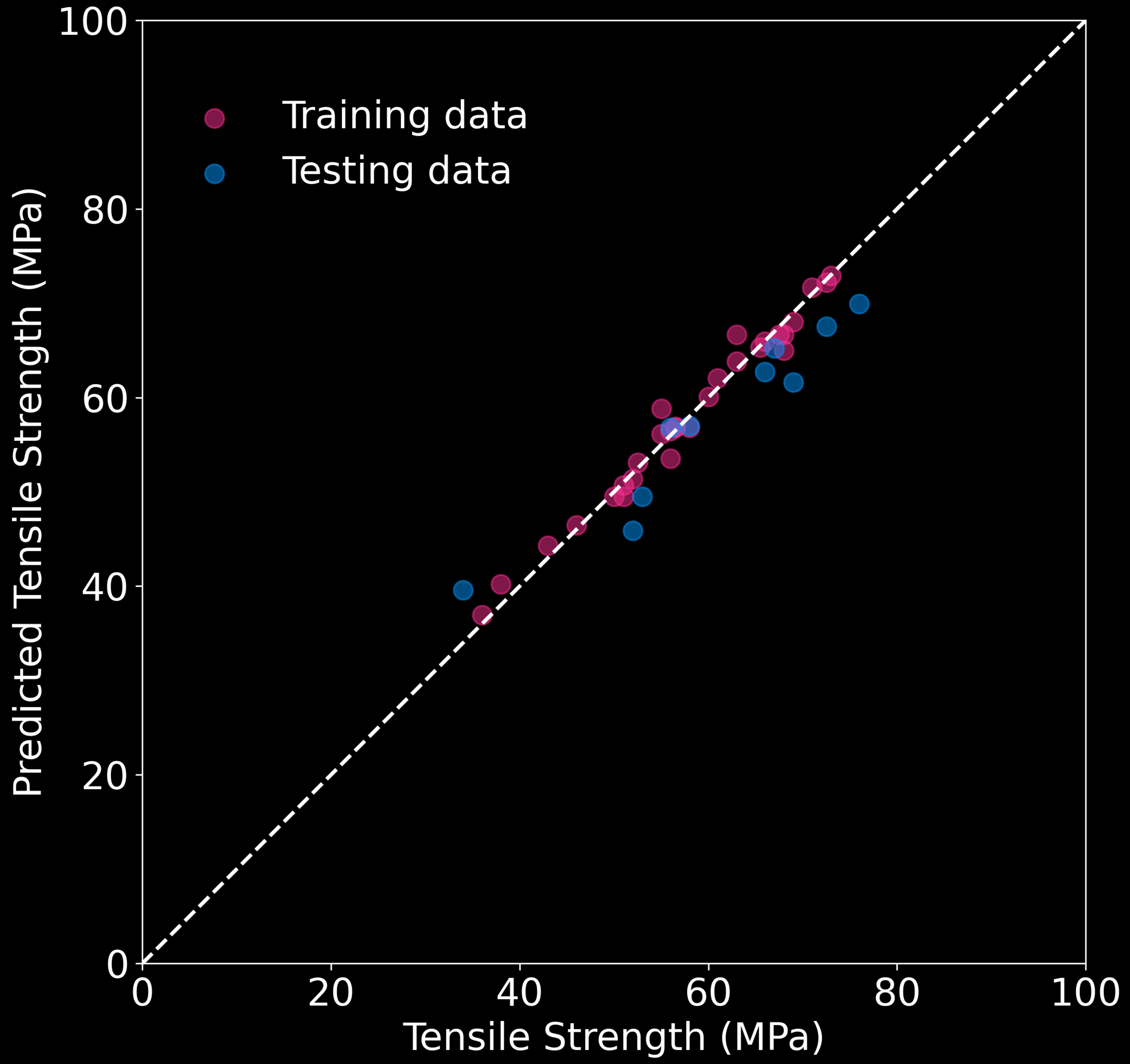
## Traditional ML




$R^2$ train = 74%

$R^2$ test = 76%

## Physics-informed ML




$R^2$ train = 98%

$R^2$ test = 85%