Project Title: Prediction of Band Gap in Perovskite Materials using ML

- **Subtitle:** A Data-Driven Study Using the `castelli_perovskites` Dataset
- **Author:** Eisuke Otsuka, Ren Sudo, Mamoru Sakaibara
- **Date:** May 29, 2025

Excuetive Summary

- Dataset
  - Used the `castelli_perovskites` dataset from the `matminer` library.
  - Contains 18,928 hypothetical perovskite compositions.
  - Each sample includes:
    - `gap gllbsc` (target variable: band gap from GLLB-SC functional),
    - `e_form` (formation energy), `mu_b` (magnetic moment),
    - `formula`, `structure`, and other DFT-calculated properties.
- Data Cleaning
  - Dropped `structure` column due to object type incompatibility.
  - Removed `cbm` and `vbm` features, which are mathematically linked to `gap gllbsc` and would lead to data leakage.
  - In the second sets of features, removed 13 constant and 18 low-variability features from the Magpie descriptor set.
- Exploratory Data Analysis (EDA)
  - Verified no missing values.
  - Pairplots and correlation heatmaps identified key features linked to band gap.
    - The first set of features: Atomic fractions (`O`, `N`, etc.), `mu_b`, `e_form`
    - The second sets of features: Physicochemical descriptors like `MagpieData avg_dev NdValence`, `MagpieData mean NdValence`, `MagpieData maximum SpaceGroupNumber`
- Modeling
  - Trained and evaluated both Linear Regression and Random Forest using 5-fold cross-validation.
  - The first set of features (Atomic Fractions):
    - Linear Regression: $R^2 \approx 0.08$
    - Random Forest: $R^2 = 0.539 \pm 0.059$, MAE = 0.068 ± 0.006
  - The second sets of features (Magpie Features):
    - Linear Regression: $R^2 \approx 0.086$
    - Random Forest: $R^2 = 0.636 \pm 0.063$, MAE = 0.065 ± 0.003
  - Permutation importance analysis revealed meaningful features in both updates:
    - `e_form`, `mu_b`, and elemental properties in the first set of features
    - Valence statistics and electronegativity in the second sets of features

    - We tried graph-based learning using GCN and made efforts to incorporate structural data, but accuracy was not sufficient

- Key Insights

    - Predicting band gaps using composition-only features is moderately successful.
        - Random Forest captures non-linear effects more effectively than Linear Regression.
    - Magpie-derived features offer a significant performance boost over basic atomic fractions.
    - Physically meaningful descriptors enhance both interpretability and accuracy.
    - Despite improvements, structure-based features are expected to further improve performance.

- Next Steps

    - We tried graph-based learning using GCN and made efforts to incorporate structural data, but accuracy was not sufficient, so improvements will be made.
    - Explore feature interactions and global/local explanations using SHAP.
    - Evaluate model uncertainty and generalizability for materials discovery pipelines.
    - Consider hybrid feature sets combining composition, structure, and graph-based representations.

Abstract

- This project explores the prediction of electronic band gaps in hypothetical perovskite materials using the `castelli_perovskites` dataset provided by the `matminer` library. In the first phase, we extracted atomic fraction features from chemical compositions and applied traditional machine learning models. While linear regression performed poorly ($R^2 \approx 0.08$), a random forest model achieved moderate accuracy ($R^2 \approx 0.54$, MAE ≈ 0.068), suggesting the non-linear nature of band gap formation. In the second phase, we expanded our feature set using Matminer's Magpie descriptors, resulting in over 100 physicochemical features per sample. Random forest performance improved further ($R^2 \approx 0.64$, MAE ≈ 0.065), and feature importance analysis highlighted physically interpretable features such as formation energy, magnetic moment, and valence electron statistics. These results demonstrate that composition-based features alone can yield meaningful band gap predictions. Future work will focus on incorporating structural descriptors via graph neural networks (We tried GCN but accuracy was not sufficient) and adding model interpretability using SHAP.

Introduction

- In materials science, predicting key physical properties from chemical composition and crystal structure is a critical task in accelerating the discovery of novel materials. One particularly important property is the electronic band gap, which governs a material's electrical behavior—determining whether it acts as a metal, semiconductor, or insulator. This makes band gap prediction a central problem in materials informatics.

- Perovskite materials (especially ABX_3-type compounds) are of growing interest due to their versatile physical and chemical properties, which make them promising candidates for applications such as photovoltaics, optoelectronics, and photocatalysis. In this project, we aim to predict the electronic band gaps of hypothetical perovskites using data-driven methods.

- We use the `castelli_perovskites` dataset from the `matminer` library, which includes 18,928 computationally generated perovskite structures along with properties such as:
    - `gap gllbsc` (band gap computed with the GLLB-SC functional),
    - `e_form` (formation energy),
    - `mu_b` (magnetic moment),
    - `structure` information (as pymatgen Structure objects).
- In the first phase of this project, we focused on predicting the band gap using simple compositional features—specifically, atomic fractions derived from the chemical formula using `pymatgen`. These features allowed us to establish baseline models using linear regression and random forest. While linear regression failed to capture non-linear trends, random forest achieved moderate performance ($R^2 \approx 0.54$), indicating some predictive power from composition alone.
- Building on this, the second phase introduced more expressive composition-based features using `matminer`'s Magpie featurizer. These features encapsulate a wide range of physicochemical descriptors, including valence statistics, electronegativity, and atomic size, leading to a total of over 100 numerical features. Incorporating these into the same modeling pipeline significantly improved performance ($R^2 \approx 0.64$ using random forest), highlighting the value of domain-aware feature engineering even in the absence of structural information.
- While the dataset contains structural data in the form of pymatgen Structure objects, we defer their use to future work due to current implementation constraints. Our ultimate aim is to incorporate structural features through graph-based deep learning models such as MEGNet to further improve accuracy and enable more physics-aware predictions.
- Our goals are:
    - To explore how traditional machine learning models (e.g., linear regression, random forest) perform in predicting band gaps from compositional descriptors,
    - To evaluate the effectiveness of Magpie-derived physicochemical features for band gap regression,
    - And to build a foundation for incorporating structural and deep learning-based models in future work.
- Data Source: Matminer `castelli_perovskites` dataset.

Data Science Methods

- In this project, we used classical supervised learning techniques to predict the electronic band gap (`gap gllbsc`) of inorganic perovskite materials.
- Tools and Libraries:
    - `matminer`: dataset loading
    - `pymatgen`: chemical formula handling
    - `pandas`, `numpy`, `seaborn`, `matplotlib`: data handling and visualization
    - `scikit-learn`: modeling and evaluation

- Feature Engineering:
    - In The first set of features, we converted the chemical formula to atomic fraction vectors using `pymatgen.Composition`.
    - In The second sets of features, we extended this by generating over 100 physicochemical features using `matminer.featurizers.composition.ElementProperty` with the `"magpie"` preset.
    - The `structure` column was excluded from both phases due to data complexity and limitations in our current pipeline.
- Modeling:
    - Linear Regression and Random Forest Regressor were trained.
- Evaluation
    - 5-fold cross-validation (metrics: MAE, R^2).
    - Permutation importance was used to interpret model results.
- Graph Construction and Modeling
    - Although good accuracy not yet obtained, we planned to incorporate structure-based graph representations using the GCN. In this approach:
    - Graph Construction:
        - Each perovskite material are converted into a graph, where atoms are represented as nodes and edges correspond to interatomic interactions within a specified cutoff radius.
        - We utilized graph converter, which accepts `pymatgen.Structure` objects directly and transforms them into model-ready graph inputs.
    - Modeling:
        - The GCN model were used as a graph neural network (GNN) for regression, predicting the `gap gllbsc` value (band gap) from structural information.
        - This method allows learning directly from the atomic environment and spatial relationships, which are often critical for electronic property prediction.
    - This enhancement aims to address current limitations of composition-only models by incorporating richer physical structure into the learning process.

Exploratory Data Analysis

Explanation of your data set

- We analyzed the `castelli_perovskites` dataset from `matminer`library, which contains 18,928 hypothetical perovskite samples and a rich variety of computed physical properties. Each entry represents a unique chemical composition, with properties calculated using density functional theory (DFT) based on the GLLB-SC functional.
- Original Features (Shared Across Both Phases)
    - `gap gllbsc`: the target variable, representing the electronic band gap (in eV)
    - `e_form` (formation energy)

    - `mu_b` (magnetic moment)
    - `fermi level`, `fermi width`: electronic structure indicators
    - `cbm`, `vbm`: conduction and valence band extrema
    - `structure`: pymatgen `Structure` object (not used in this analysis)

- Feature Engineering Approaches
    - To explore the effectiveness of different compositional descriptors, we created two distinct feature sets across two project phases.
- The first set of features – Atomic Fraction Features
    - In the first phase, we extracted elemental atomic fractions from each chemical formula using `pymatgen.Composition`. This process decomposes a formula like `BaTiO3` into fractional representations such as Ba:0.25, Ti:0.25, O:0.5. These were used alongside numeric DFT-calculated properties (`e_form`, `mu_b`, etc.) for machine learning.

```
# Convert formula to comoposition
from pymatgen.core.composition import Composition
from tqdm import tqdm
tqdm.pandas()

def get_atomic_fraction_dict(formula):
    try:
        comp = Composition(formula)
        return comp.fractional_composition.as_dict()
    except Exception as e:
        print(f"Error at: {formula}, {e}")
        return {}

# Apply
df_atomic_frac = df["formula"].progress_apply(get_atomic_fraction_dict)

df_atomic_frac = pd.DataFrame(df_atomic_frac.tolist()).fillna(0)

# Merged into the original DataFrame
df = pd.concat([df, df_atomic_frac], axis=1)

# Basic information on the data set
print("Number of rows:", df.shape[0])
print("Number of variables (columns):", df.shape[1])
print("\nData types:\n", df.dtypes.value_counts())
print("\nMissing values per column:\n", df.isnull().sum())
print("\nUnique values in categorical columns:\n", df.drop(columns=
["structure"]).select_dtypes(include=["object", "bool"]).nunique())
```

Item	Value
Number of rows	18,928
Number of columns	66

Item	Value
Data types	float64 (63), object (2), bool (1)
Missing values	None
Categorical variables	`gap is direct`: 2 values, `formula`: 18,928 unique

- Table 1: Summary of dataset after atomic fraction expansion (The first set of features)
- The second sets of features – Magpie Physicochemical Features
    - In the second phase, we enhanced the dataset using the Magpie feature set from `matminer.featurizers.composition.ElementProperty`. This featurizer calculates over 100 statistical descriptors (mean, range, std, etc.) based on physical and chemical properties of the constituent elements—such as atomic number, covalent radius, electronegativity, valence orbital counts, melting point, and more.

```
from matminer.featurizers.composition import ElementProperty
featurizer = ElementProperty.from_preset("magpie", impute_nan=True)
df = featurizer.featurize_dataframe(df, col_id="composition", ignore_errors=True)
# Basic information on the data set
print("Number of rows:", df.shape[0])
print("Number of variables (columns):", df.shape[1])
print("\nData types:\n", df.dtypes.value_counts())
print("\nMissing values per column:\n", df.isnull().sum())
print("\nUnique values in categorical columns:\n", df.drop(columns=
["structure"]).select_dtypes(include=["object", "bool"]).nunique())
```

Item	Value
Number of rows	18,928
Number of columns	112
Data types	float64 (109), object (3)
Missing values	None
Categorical variables	`composition`: 9646 values, `formula`: 18,928 unique

- Table 2: Summary of dataset after applying Magpie featurization (The second sets of features)
- This two-tiered approach allowed us to:
    - Benchmark simpler atomic-fraction features (lightweight, interpretable)
    - Evaluate the value added by richer, statistically aggregated physical descriptors (Magpie)
- Both versions retained consistent row indices and target values, enabling a fair model comparison.

Data Cleaning

- To prepare the dataset for machine learning, several preprocessing steps were applied consistently across both phases of the project.
- Common Cleaning Procedures (The first set of features & 2)
    - Dropped the `structure` column due to incompatibility with standard ML pipelines (non-hashable object).
    - Removed `cbm` and `vbm` from the modeling phase, as they are mathematically related to `gap gllbsc`.
        - Verification:

```
diff = df["cbm"] - df["vbm"]
is_equal = np.isclose(diff, df["gap gllbsc"])
print(is_equal.sum() / len(df))  # → About 96% are exact matches.
```

- Additional Cleaning for Magpie Features (The second sets of features)
    - When the dataset was augmented with Magpie features using `matminer`, additional cleaning was required:
        - Constant columns (with only one unique value across all samples) were removed.
        - Example: `MagpieData minimum NsValence`, `MagpieData minimum NfUnfilled`
        - Low-uniqueness features (columns with ≤3 unique values) were also discarded, as they are unlikely to contribute useful signal and may introduce noise.
        - Example: `MagpieData maximum NfValence`, `gap is direct`
    - In total, 13 constant and 18 low-uniqueness features were removed in The second sets of features.

```
# Remove constant columns
constant_cols = [col for col in df.columns if df[col].nunique(dropna=False) <= 1]
df = df.drop(columns=constant_cols)

# Remove low-uniqueness features
low_unique_cols = [col for col in df.columns if df[col].nunique() <= 3]
df = df.drop(columns=low_unique_cols)
```

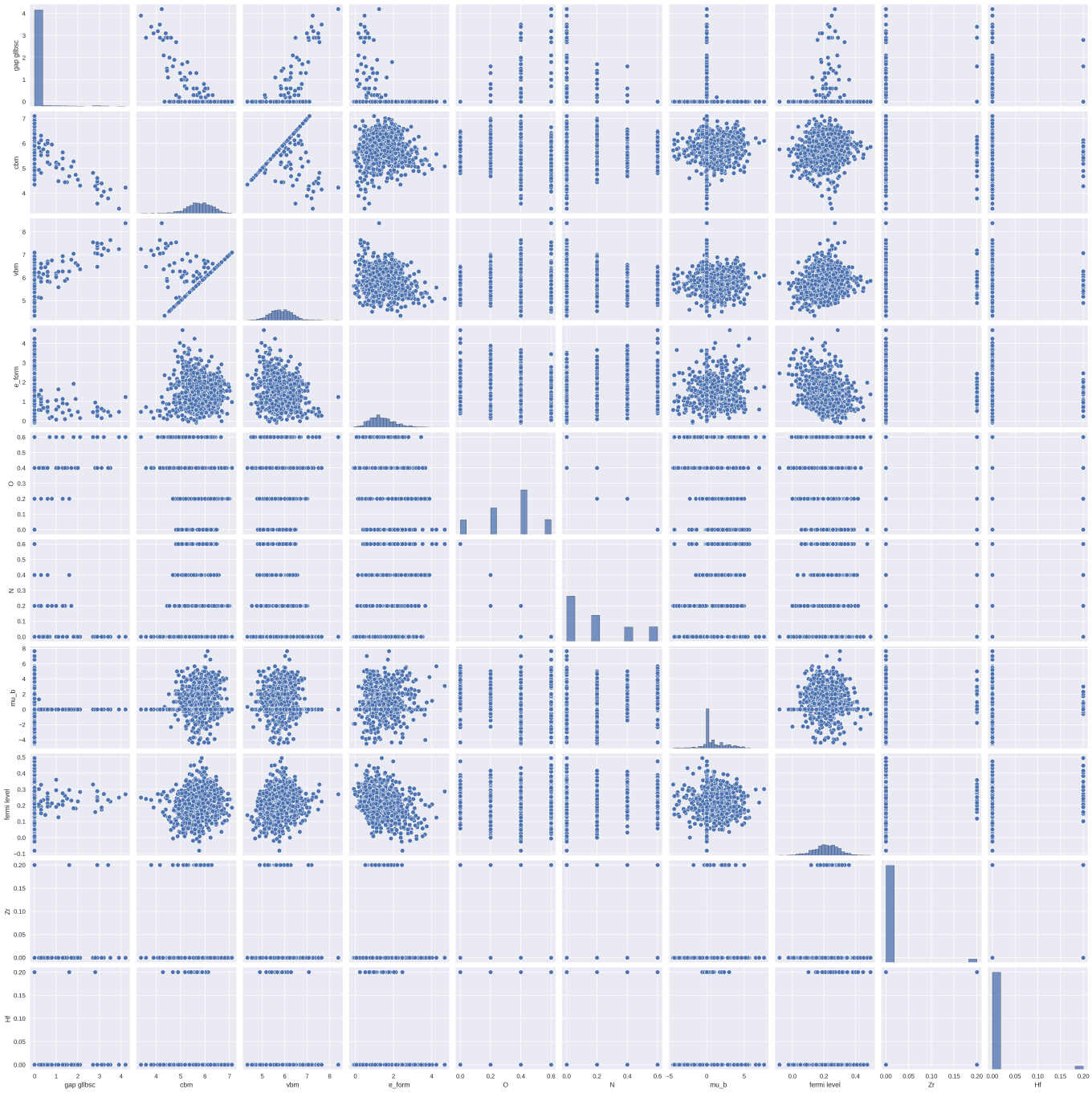
- Final Result
    - After cleaning:
        - All remaining features were numeric and suitable for modeling.
        - No missing values were present.
        - The dataset was fully standardized and ready for cross-validation and interpretation.

Data Vizualizations

- To explore the underlying structure of the data and guide feature selection, we generated visualizations focusing on the pairwise relationships between `gap gllbsc` (band gap) and the most influential input features. Separate analyses were conducted for the atomic-fraction-based feature set (The first set of features) and the Magpie-based feature set (The second sets of features).

- The first set of features – Atomic Fraction Features
    - In the first phase, we visualized the top 10 features correlated with the band gap, including `e_form, mu_b`, and selected elemental atomic fractions (e.g., `O`, `N`, `Zr`). Pairplots revealed the following trends:
        - Pairwise trends between `gap gllbsc`, `e_form`, and atomic fractions were weakly non-linear.
        - Many element fraction features (e.g., `O`, `N`, `Zr`) showed sparse or binary distributions.

```
# Select numeric columns only
num_cols = df_top_features.select_dtypes(include="number").columns

# Limit to 1000 samples as too many samples are heavy
sns.pairplot(df_top_features[num_cols].sample(1000, random_state=0))
plt.show()
```



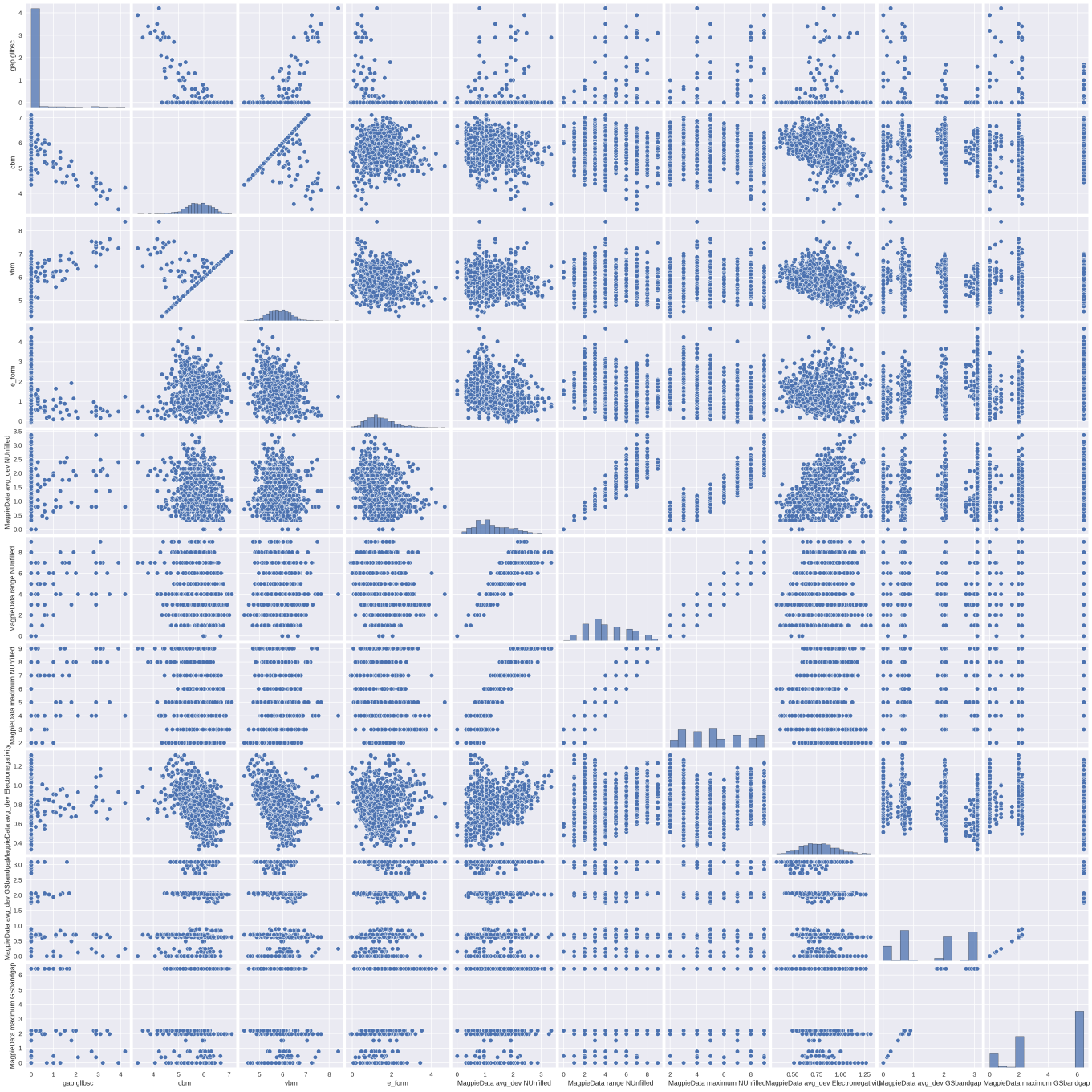
- Fig.1: Pairplot of top 10 features related to band gap.

- The second sets of features – Magpie Features

    - In the second phase, we repeated the pairplot analysis using the top 10 Magpie-derived features most correlated with `gap gllbsc`. These included statistical descriptors such as:
        - `MagpieData avg_dev/range/maximum NUnfilled`
        - `MagpieData ave_dev Electronegativity`
        - `MagpieData ave_dev/maximum GSbandgap` etc.
    - These features tended to show broader distributions with smoother gradients across samples, compared to the binary-like patterns in atomic fractions.

```
# Select numeric columns from Magpie features
num_cols = df_magpie_top_features.select_dtypes(include="number").columns
```

```
# Sample for clarity
sns.pairplot(df_magpie_top_features[num_cols].sample(1000, random_state=0))
plt.show()
```

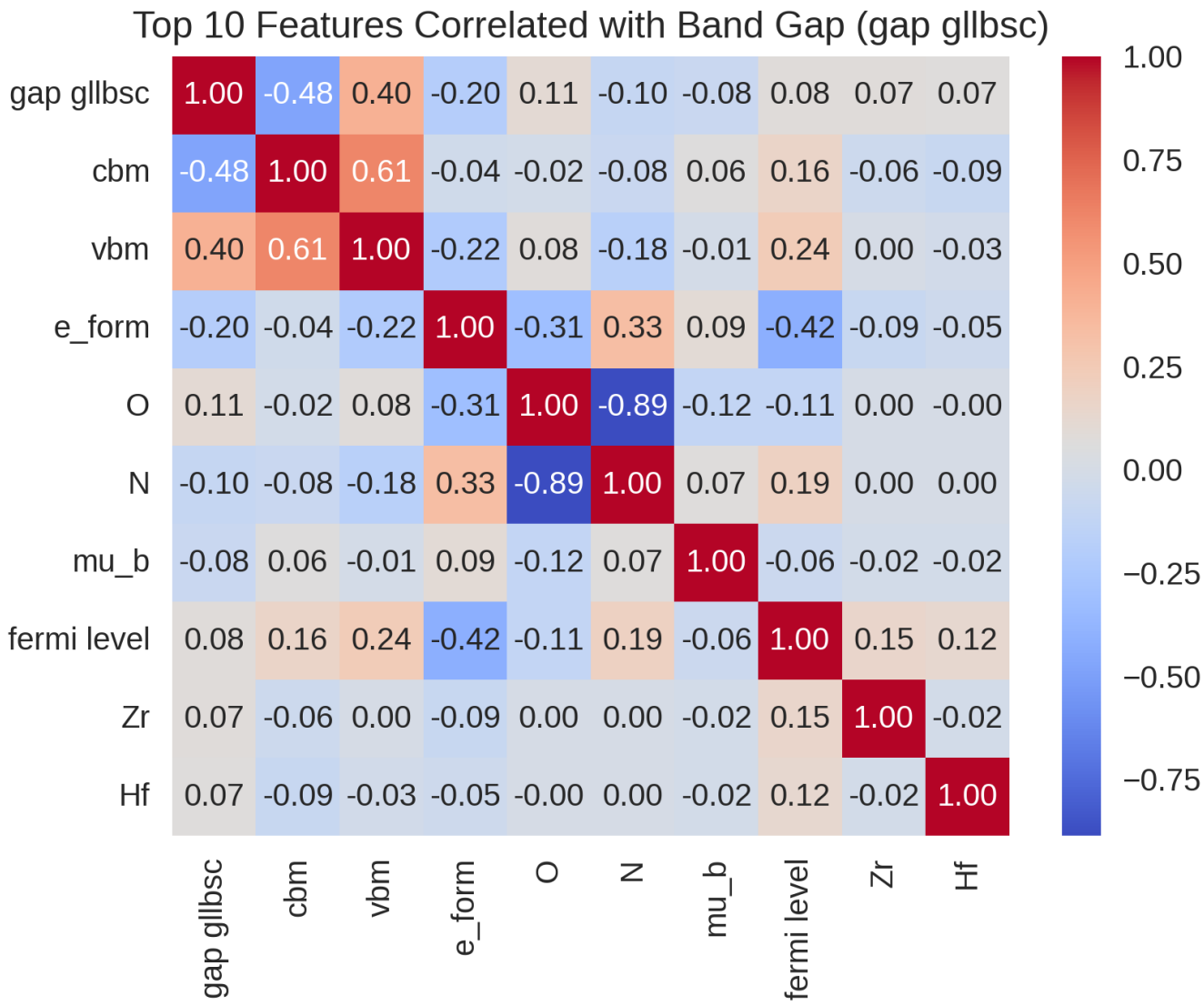


- Fig. 2: Pairplot of top 10 Magpie features related to band gap.
- Insights
    - Atomic fraction features provide a coarse but interpretable view of compositional effects on band gap.
    - Magpie features capture richer, more continuous trends due to their aggregation of elemental physical properties.
    - Both feature sets exhibit non-linear relationships with the target, motivating the use of non-linear models.

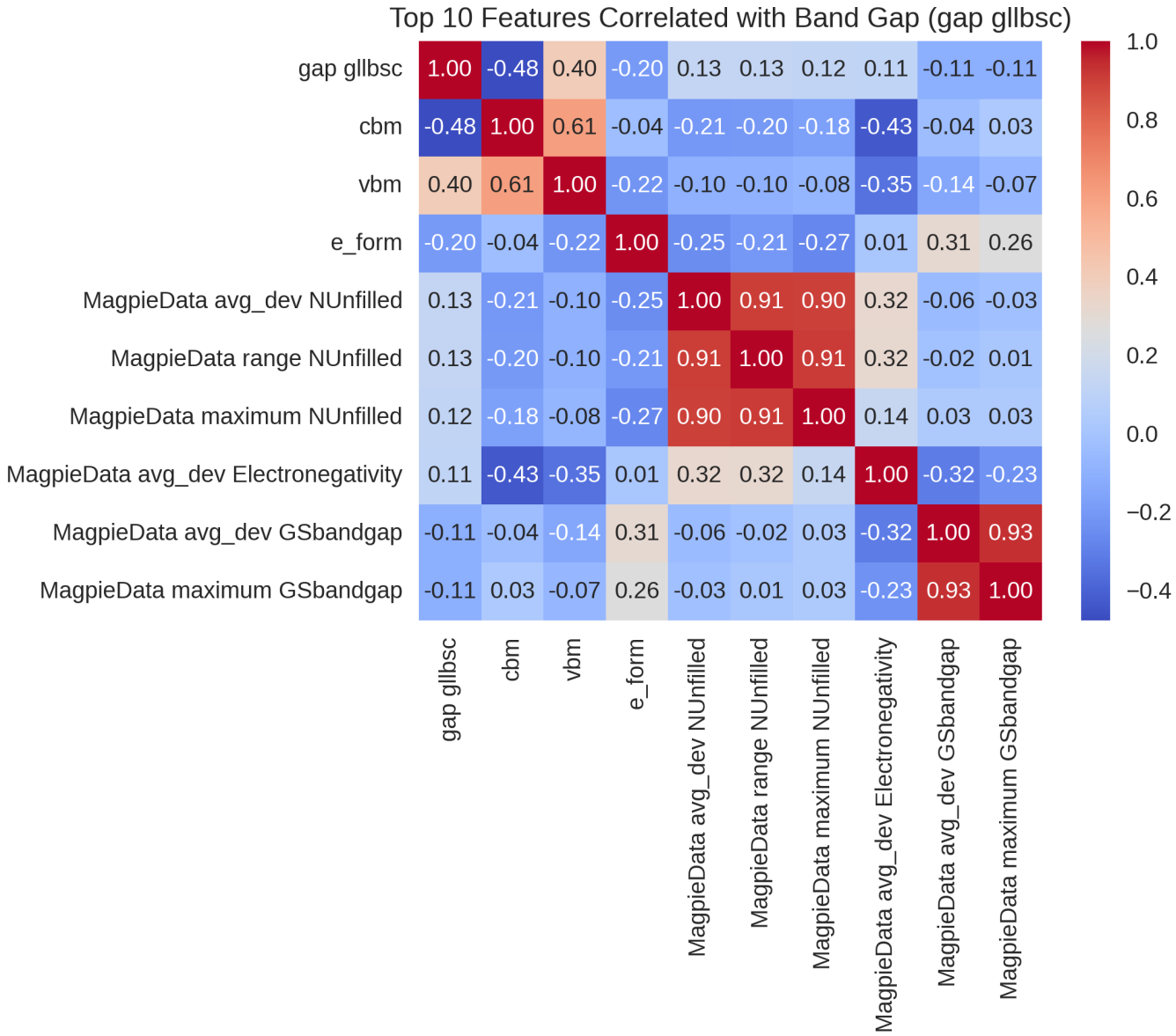
Variable Correlations

- To better understand which features most influence the target variable `gap gllbsc`, we computed pairwise Pearson correlation coefficients and visualized the top 10 most correlated features in each feature set.
- The first set of features – Atomic Fraction + Physical Properties
    - In the first phase, the correlation matrix revealed the following:
        - `cbm` and `vbm` were highly positively correlated with `gap gllbsc`, as expected given their direct mathematical relationship.
        - Other correlated features included:
            - Elemental atomic fractions (e.g., `O`, `N`)
            - Computed properties like `mu_b` (magnetic moment), `e_form` (formation energy), and `fermi level`
        - This analysis confirmed that both compositional and physical properties carry some predictive signal, although many relationships were weak and non-linear.

```
# Correlation heat map
import numpy as np

# Target variable.
target_col = "gap gllbsc"

# Obtain the absolute value of the correlation coefficient and select the top n
correlations with the objective variable
top_n = 10
corr_with_target = df_features[numeric_cols].corr()
[target_col].abs().sort_values(ascending=False)
top_features = corr_with_target.head(top_n).index.tolist()

df_top_features = df_features[top_features]

# Visualisation of correlation matrices (limited to upper-level features)
sns.heatmap(df_top_features.corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title(f"Top {top_n} Features Correlated with Band Gap (gap gllbsc)")
plt.show()
```



- Fig.3: Top 10 features correlated with `gap gllbsc`.
- The second sets of features – Magpie Features
    - In the second phase, using the expanded Magpie-derived feature set, the correlation analysis yielded a more nuanced picture:
        - Physically meaningful variables such as:
            - `MagpieData avg_dev/range/maximum NUnfilled`
            - `MagpieData ave_dev Electronegativity`
            - `MagpieData ave_dev/maximum GSbandgap`
        - were among the top features correlated with the band gap.
    - These features captured subtler physical and chemical trends not evident in elemental fractions alone.



- Fig. 4: Top 10 correlated features from Magpie descriptors.
- Insights
    - The first set of features: Correlations were sparse and often binary or step-like due to the nature of atomic fractions.
    - The second sets of features: Correlation patterns were smoother and revealed richer inter-feature relationships, enabling more nuanced modeling.
- This analysis validated that the Magpie feature set offers improved physical interpretability and modeling potential over simple compositional features.

Statistical Learning: Modeling & Prediction

- To explore the relationship between material features and the band gap (gap gllbsc), we applied two types of supervised learning models: a simple linear regression and a non-linear ensemble model (Random Forest Regressor).

Model selection

- Linear Regression
  - A baseline model was constructed using a standard linear regression on all numerical features, including physical.
- Random Forest Regression
  - To capture potential non-linear relationships, we trained a Random Forest model with default parameters, using 5-fold cross-validation to ensure generalization.
- GCN
  - Encoding the crystal structure of materials and capturing spatial information that is not possible with elemental properties alone

Cross-validation, Predictive R2

- To evaluate the predictive performance of each model and feature set, we used 5-fold cross-validation with two metrics: Mean Absolute Error (MAE) and coefficient of determination (R^2). For each fold, the model was trained on 80% of the data and tested on the remaining 20%, with results averaged over all folds.

- The first set of features – Atomic Fraction Features

  - We first assessed the predictive performance of both linear regression and random forest models trained on atomic fraction features.

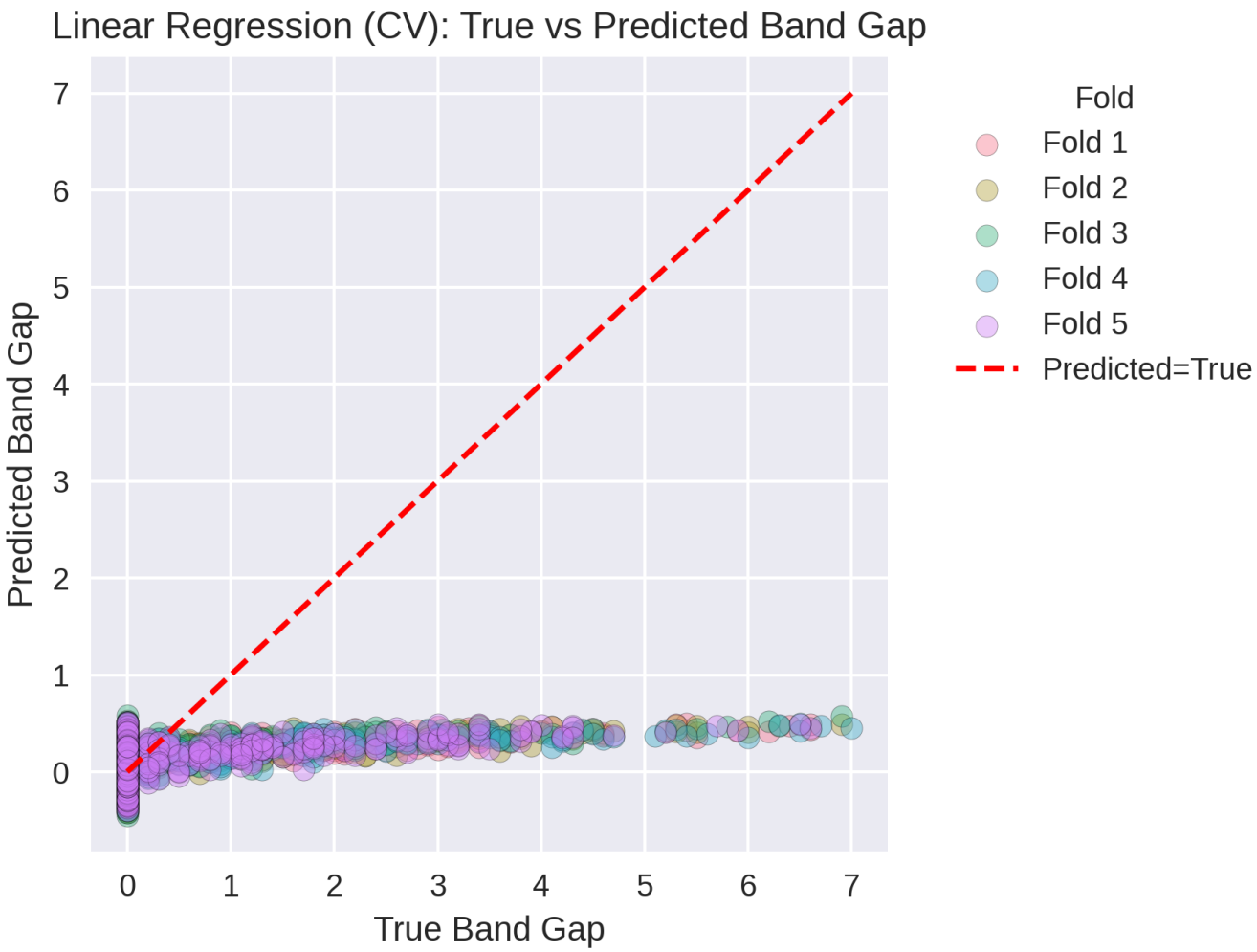
- Random Forest significantly outperformed the linear model(Fig.5,Fig.6), highlighting the importance of non-linear modeling in materials property prediction. These results show that the linear model failed to capture the non-linear structure in the data, while Random Forest significantly improved performance, validating its suitability for this task.

```
# Definition of features and target variables
X = df_features.drop(columns=["gap gllbsc", "formula", "gap is direct", "cbm",
"vbm", "structure"])
y = df_features["gap gllbsc"]
feature_names = X.columns

# Scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Preparing models and cross-validation
model = RandomForestRegressor(n_estimators=100, random_state=42, n_jobs=-1)
cv = KFold(n_splits=5, shuffle=True, random_state=42)

# For saving results
y_true_all, y_pred_all, fold_ids = [], [], []
importances_list = []
mae_list, r2_list = [], []

# Color palette for folds
colors = sns.color_palette("husl", n_colors=cv.get_n_splits())

# CV loop
```

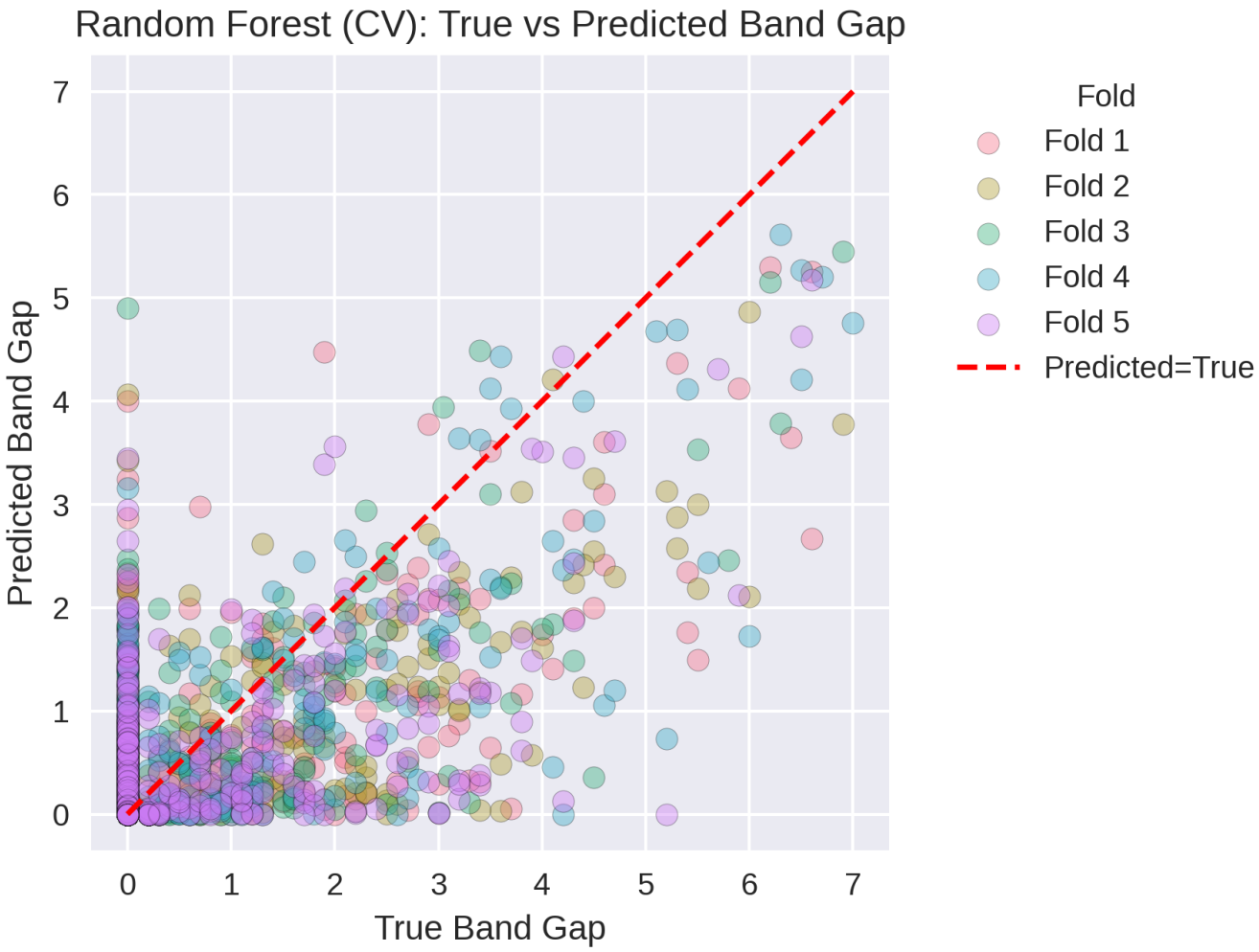
```
for fold, (train_idx, val_idx) in enumerate(cv.split(X_scaled)):
    X_train, X_val = X_scaled[train_idx], X_scaled[val_idx]
    y_train, y_val = y.iloc[train_idx], y.iloc[val_idx]

    model.fit(X_train, y_train)
    y_pred = model.predict(X_val)

    y_true_all.extend(y_val)
    y_pred_all.extend(y_pred)
    fold_ids.extend([fold] * len(y_val))  # Store fold info for each point

    # Score output
    mae = mean_absolute_error(y_val, y_pred)
    r2 = r2_score(y_val, y_pred)
    mae_list.append(mae)
    r2_list.append(r2)
    print(f"Fold {fold+1} - MAE: {mae:.3f}, R²: {r2:.3f}")

    # Permutation importance (by val data)
    result = permutation_importance(model, X_val, y_val, n_repeats=5,
random_state=42, n_jobs=-1)
    importances_list.append(result.importances_mean)

# Average score output
print(f"\nAverage MAE: {np.mean(mae_list):.3f} ± {np.std(mae_list):.3f}")
print(f"Average R²: {np.mean(r2_list):.3f} ± {np.std(r2_list):.3f}")

# Scatterplot with fold-wise color & legend
for fold in range(cv.get_n_splits()):
    indices = [i for i, f in enumerate(fold_ids) if f == fold]
    plt.scatter(
        np.array(y_true_all)[indices],
        np.array(y_pred_all)[indices],
        color=colors[fold],
        alpha=0.4,
        label=f"Fold {fold+1}",
        edgecolor='k',
        linewidth=0.2
    )
plt.plot([min(y_true_all), max(y_true_all)], [min(y_true_all), max(y_true_all)],
'r--',label="Predicted=True")
plt.xlabel("True Band Gap")
plt.ylabel("Predicted Band Gap")
plt.title("Random Forest (CV): True vs Predicted Band Gap")
plt.legend(title="Fold", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

Model	MAE (± std)	R^2 (± std)
Linear Regression	0.173 ± 0.005	0.084 ± 0.012
Random Forest	0.068 ± 0.006	0.539 ± 0.059

- Table.3:Scores for each model in 5-fold CV



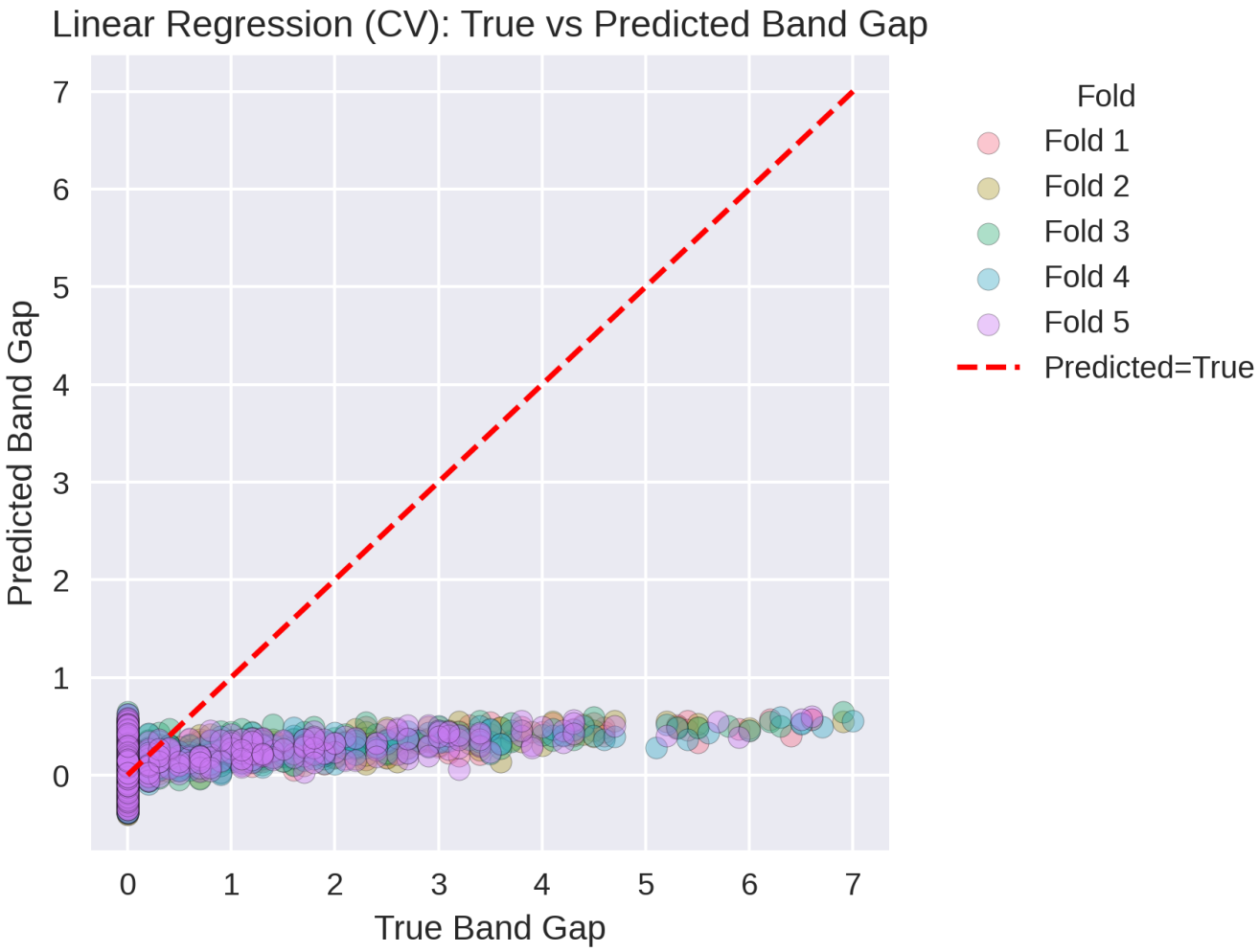
- Fig.5:Linear Regression (CV): True vs Predicted Band Gap



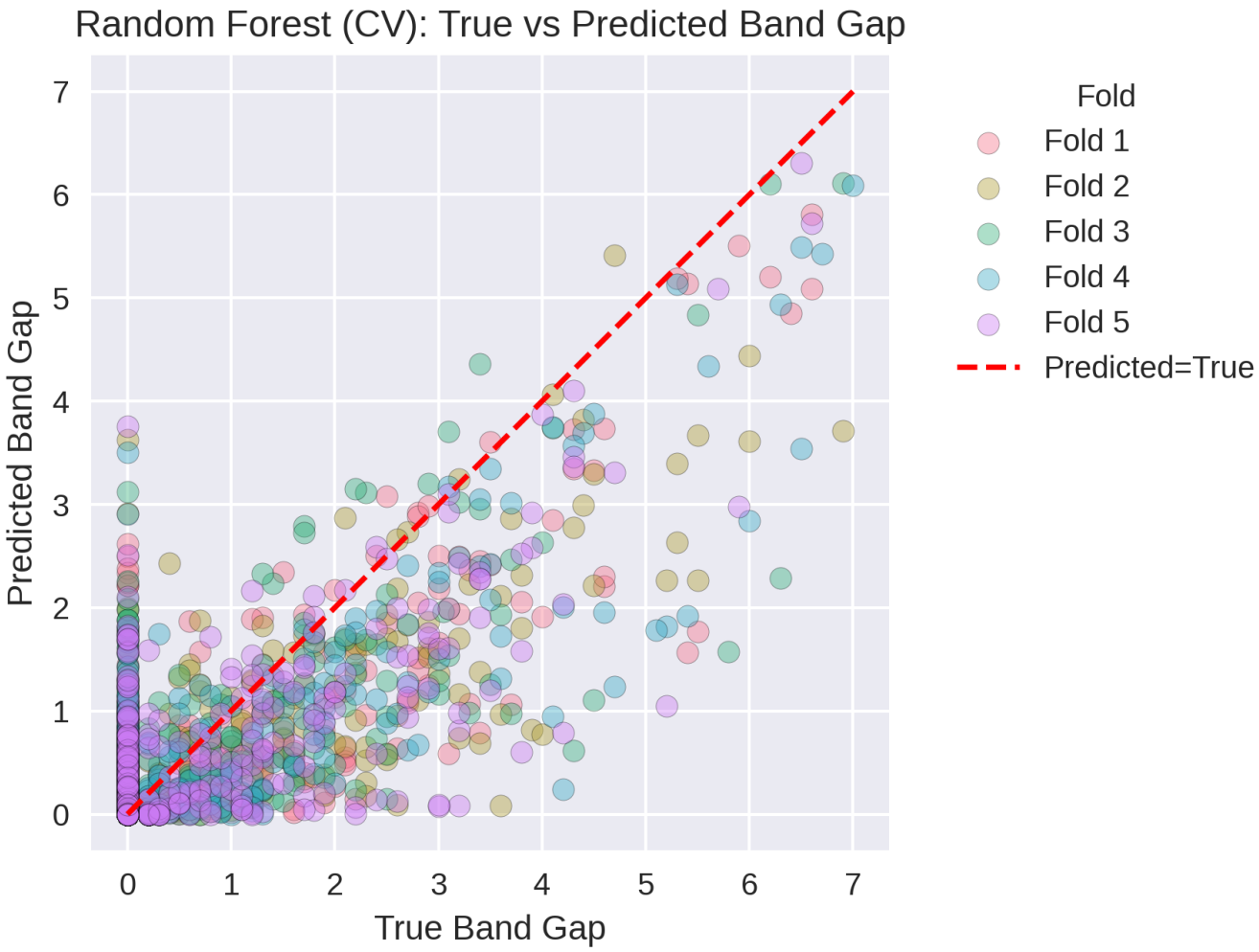
- Fig.6:Random Forest (CV): True vs Predicted Band Gap
- The second sets of features – Magpie Physicochemical Features
    - We then repeated the cross-validation using the Magpie feature set, which provides richer, statistically aggregated descriptors derived from elemental properties.
- These results demonstrate a clear gain in accuracy when using Magpie features, especially with Random Forest. This suggests that the additional physicochemical information captured by the Magpie featurizer allows the model to learn more meaningful patterns.

Model	**MAE (± std)**	**R^2 (± std)**
Linear Regression	0.172 ± 0.005	0.086 ± 0.019
Random Forest	0.065 ± 0.003	0.636 ± 0.063

- Table 4: 5-fold CV scores using Magpie features (The second sets of features)



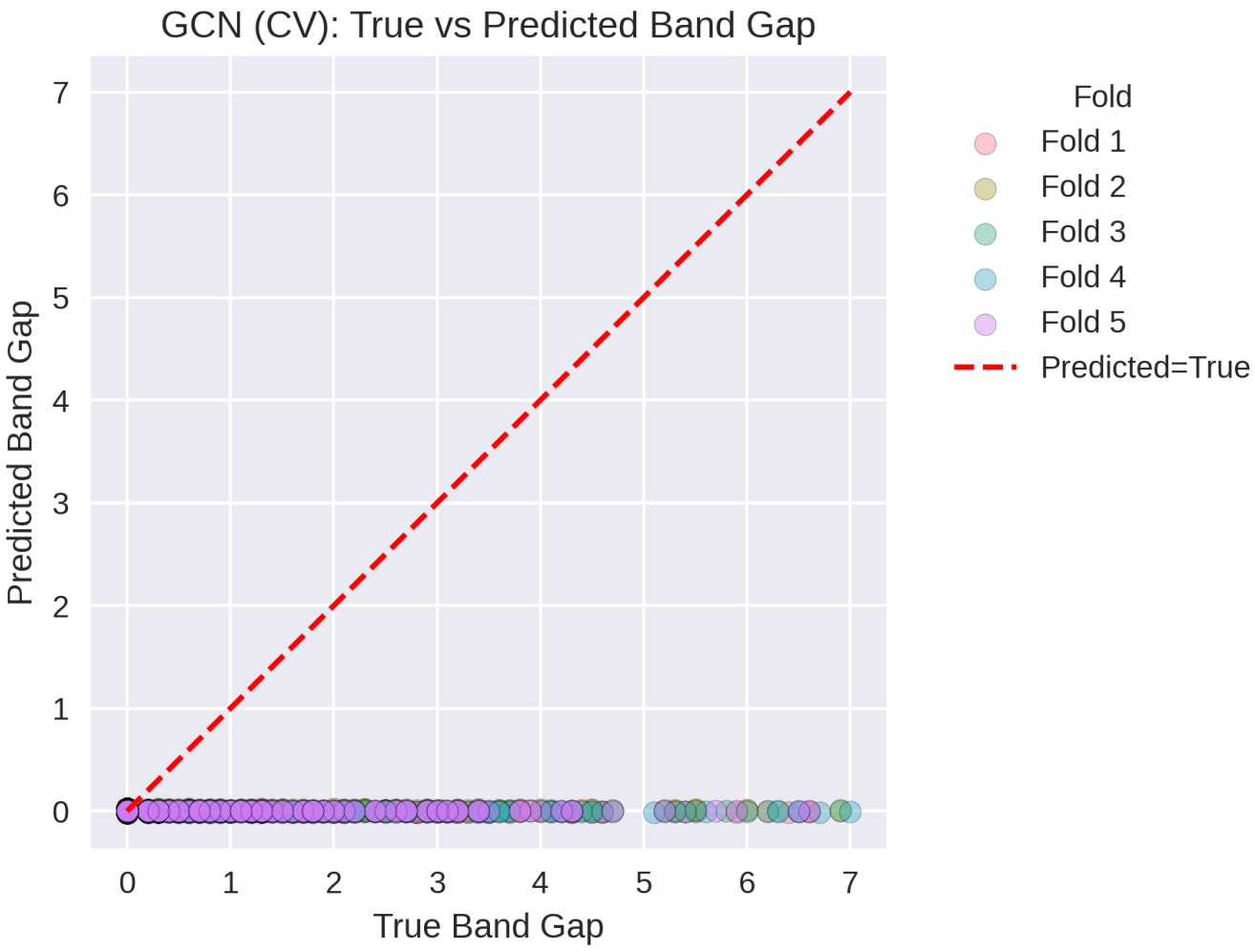
- Fig.7: Linear Regression (Magpie) – True vs Predicted Band Gap



- Fig.8: Random Forest (Magpie) – True vs Predicted Band Gap
- GCN modelling
    - We tried Graph-based learning using GCN and made efforts to incorporate structural data, but accuracy was not sufficient.
- Almost all predictions were constants (=0), worse than linear regression, probably because no distance information was given for GCN.

```
def structure_to_graph(structure: Structure, cutoff=5.0):
    positions = torch.tensor([site.coords for site in structure],
dtype=torch.float)
    atomic_numbers = torch.tensor([site.specie.number for site in structure],
dtype=torch.long)

    edge_index = []
    for i, site_i in enumerate(structure):
        for j, site_j in enumerate(structure):
            if i != j and site_i.distance(site_j) < cutoff:
                edge_index.append([i, j])
    edge_index = torch.tensor(edge_index, dtype=torch.long).t().contiguous()
    return Data(x=atomic_numbers.unsqueeze(1), edge_index=edge_index,
pos=positions)
```

```
class GCN(torch.nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = GCNConv(1, 64)
        self.conv2 = GCNConv(64, 64)
        self.lin = nn.Linear(64, 1)

    def forward(self, x, edge_index, batch):
        x = F.relu(self.conv1(x, edge_index))
        x = F.relu(self.conv2(x, edge_index))
        x = global_mean_pool(x, batch)
        return self.lin(x)
```

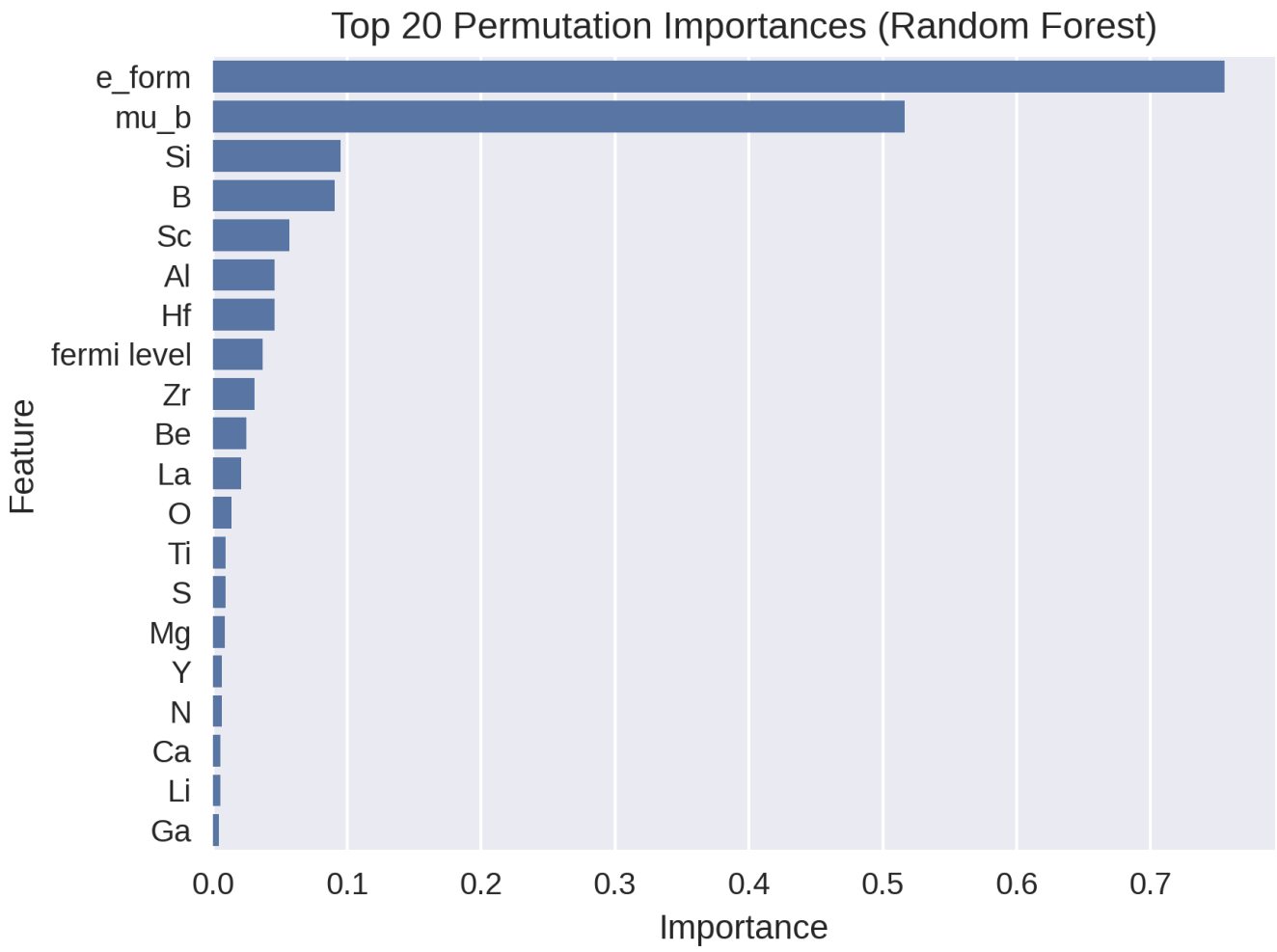


- Fig.9: GCN – True vs Predicted Band Gap
- Summary of Findings
    - Random Forest consistently outperformed Linear Regression across both feature sets.
    - The use of Magpie features resulted in improved generalization, with R^2 increasing from ~0.54 to ~0.64 and MAE decreasing from ~0.068 to ~0.065.
    - These results support the hypothesis that richer, physically grounded features lead to better predictive models, even without structural information.

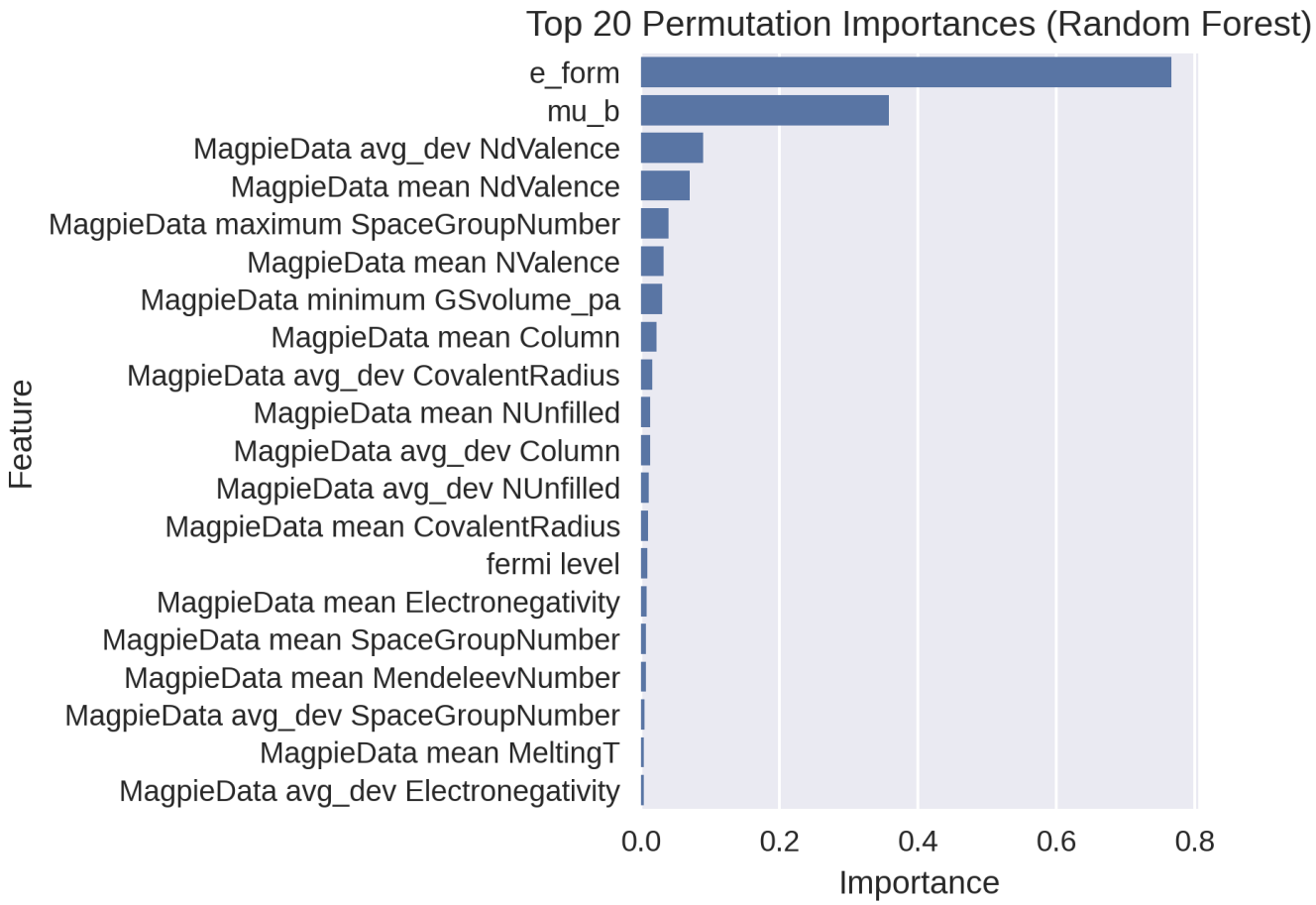
Interpret results

- To interpret model behavior and understand which features drive predictions, we applied Permutation Importance analysis to the trained Random Forest models in both The first set of features and The second sets of features.
- The first set of features – Atomic Fraction + Physical Properties
    - Using the baseline atomic fraction and DFT-derived features, the most influential variables were:
        - `e_form`: formation energy
        - `mu_b`: magnetic moment
        - Atomic fractions of elements such as Si, B, Sc, Al
    - These results are consistent with physical intuition: formation energy reflects chemical stability, magnetic moment captures spin effects, and light elements influence band structure through orbital hybridization.

```
# Permutation importance
avg_importance = np.mean(importances_list, axis=0)
imp_df = pd.DataFrame({"Feature": feature_names, "Importance": avg_importance})
imp_df = imp_df.sort_values("Importance", ascending=False).head(20)

sns.barplot(data=imp_df, x="Importance", y="Feature")
plt.title("Top 20 Permutation Importances (Random Forest)")
plt.tight_layout()
plt.show()
```



- Fig.10:Top 20 Permutation Importances (Random Forest, Update1)
- The second sets of features – Magpie Features
    - When trained on the richer Magpie feature set, Random Forest identified additional high-importance features:
        - `MagpieData avg_dev NdValence`
        - `MagpieData mean NdValence`
        - `MagpieData maximum SpaceGroupNumber`, among others
    - These features reflect physicochemical trends such as bonding character and electronic configuration, which are tightly coupled to band structure. Feature rankings became more physically interpretable and diverse with Magpie features.



- Fig.11:Top 20 Permutation Importances (Random Forest, Update2)

Discussion

- The central question of this study was:
    - Can we predict the electronic band gap of hypothetical perovskite materials using only compositional features and classical machine learning?
- Key Findings
    - Non-linearity is critical:
        - Linear regression performed poorly ($R^2 \approx 0.08$ in both Updates), indicating that band gap formation is inherently non-linear.
    - Random Forest captures complexity:
        - Random Forest models achieved:
        - $R^2 \approx 0.54$ with atomic fraction features
        - $R^2 \approx 0.64$ with Magpie features
        - This highlights the importance of using both non-linear models and informative features.
    - Feature importance is physically meaningful:
        - `e_form` and `mu_b` ranked highly in both updates
        - Magpie features brought in broader physical context, such as and valence variation
    - Composition-only modeling is surprisingly effective:

        - Even without structural features, chemical formula encodes useful information for screening purposes.

- Limitations
    - No structural descriptors:
        - Excluding structure objects omits key variables like symmetry, coordination, and orbital overlap.
    - Redundant or collinear features:
        - Especially in the atomic fraction set, redundancy among elements may hinder generalization.
    - No uncertainty estimation:
        - Current models are deterministic; in materials discovery, confidence intervals or prediction intervals are often necessary.

Conclusions

- This project demonstrated that machine learning models trained solely on compositional features can moderately predict the band gaps of perovskite materials. Our main conclusions are:
    - Non-linear modeling is essential:
        - Random Forest consistently outperformed linear regression across all feature sets.
    - Physically meaningful features dominate:
        - Formation energy, magnetic moment, and other domain-informed features were highly predictive.
    - Magpie features outperform atomic fractions:
        - The added physical context improves accuracy, interpretability, and generalization.
    - Composition-only models are useful for screening, particularly when structural data are unavailable.
- Future directions to improve model performance and utility include:
    - Incorporating structural features (e.g., space group, coordination environment, graph-based representation).
    - We tried graph-based learning using GCN and made efforts to incorporate structural data, but accuracy was not sufficient, so improvements will be made.
    - Introducing uncertainty quantification, which is crucial for experimental prioritization.
    - Using SHAP or other model-agnostic explanation tools to better understand interactions among features.
- This study lays a foundation for more advanced modeling and paves the way for integrating structure-aware deep learning techniques in future updates.

Acknowledgments

- We would like to thank Professor Pawan Tripathi for organizing the course Materials Informatics and for providing valuable guidance throughout the project.

- We also acknowledge the contributions of Ivano E. Castelli et al. for making the `castelli_perovskites` dataset publicly available via the `matminer` library.
- This project was conducted as part of the coursework at materials data science course offered in spring of 2025 at Tohoku University, Japan, and we appreciate the collaborative discussions within the class and group.

References

1. Castelli, I. E., Landis, D. D., Thygesen, K. S., Dahl, S., Chorkendorff, I., Jaramillo, T. F., & Jacobsen, K. W. (2012).
   New cubic perovskites for one- and two-photon water splitting using the computational materials repository.
   Energy & Environmental Science, **5**(10), 9034–9043. https://doi.org/10.1039/C2EE22341D
2. Matminer documentation.
   http://hackingmaterials.lbl.gov/matminer/
3. Sklearn documentation.
   https://scikit-learn.org/stable/
4. Github Repo. *https://github.com/re-re-code/25s-MatDataSci-Tohoku/tree/main/5-proj*