ACCSCIENCE PUBLISHING

International Journal of AI for Materials and Design

ORIGINAL RESEARCH ARTICLE

Machine-learned molecular modeling of ruthenium: A Kolmogorov-Arnold Network approach

Supplementary File

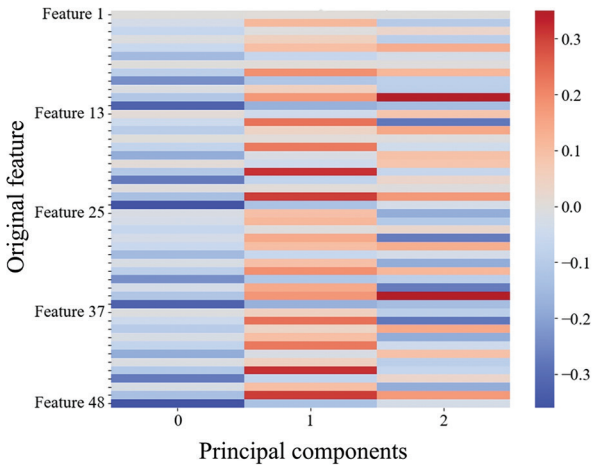

Figure S1. Structural PCA reveals dominant structural features and efficient dimensionality reduction in the crystal structure dataset, with each structural feature significantly represented in at least one PC. Heatmap of PCA loadings, revealing complex relationships between original structural features and the three main PCs. PC0 exhibits diverse strong positive and negative loadings, indicating a multifaceted combination of structural attributes. PC1 displays predominantly positive loadings, potentially representing a general size or scale factor common across features. PC2 displays more subtle loading variations, capturing finer structural nuances. Red indicates positive correlations, while blue negative correlations, with intensity reflecting correlation strength. Abbreviations: PCA: Principal component analysis; PC: Principal component.

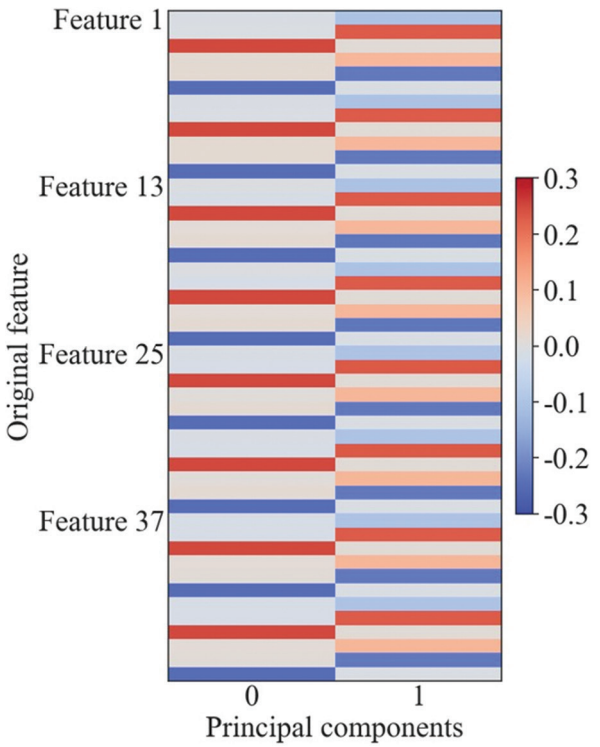

Figure S2. Force PCA reveals dominant force features and efficient dimensionality reduction in the dataset, with each force feature significantly represented in at least one PC. Heatmap of PCA loadings, revealing relationships between original force features and the two main PCs. Red indicates positive correlations, while blue indicates negative correlations, with intensity reflecting correlation strength. Abbreviations: PCA: Principal component analysis; PC: Principal component.

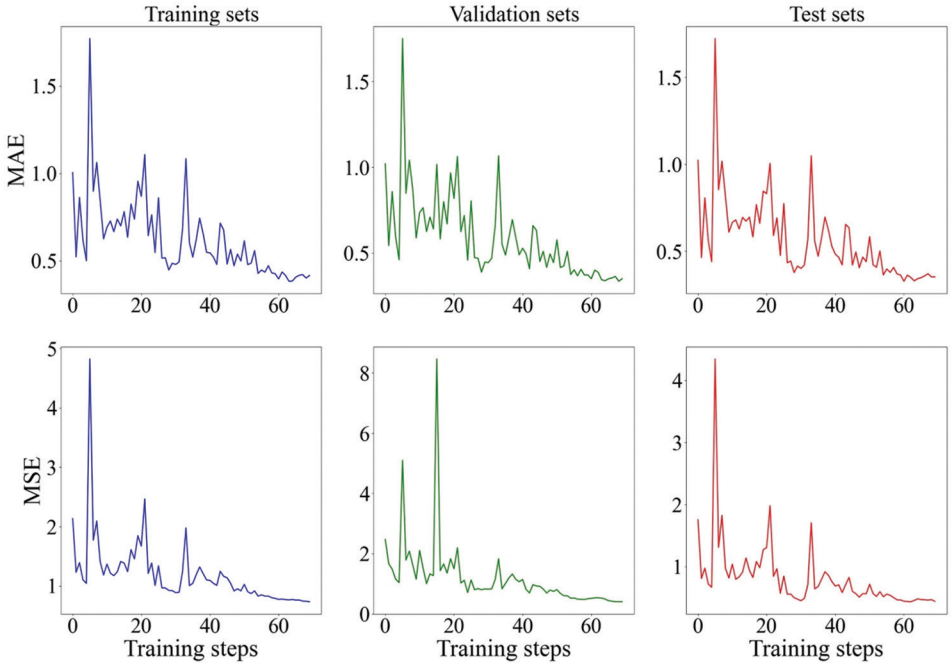

Figure S3. Training dynamics of the KAN model across different data partitions. The plots display the MAE (top row) and MSE (bottom row) versus training steps. All three datasets exhibit similar decreasing error trends, with initial fluctuations stabilizing over time, indicating robust model convergence without overfitting. Blue, green, and red lines represent the training, validation, and test sets, respectively.
Abbreviations: KAN: Kolmogorov-Arnold Network; MAE: Mean absolute error; MSE: Mean squared error.

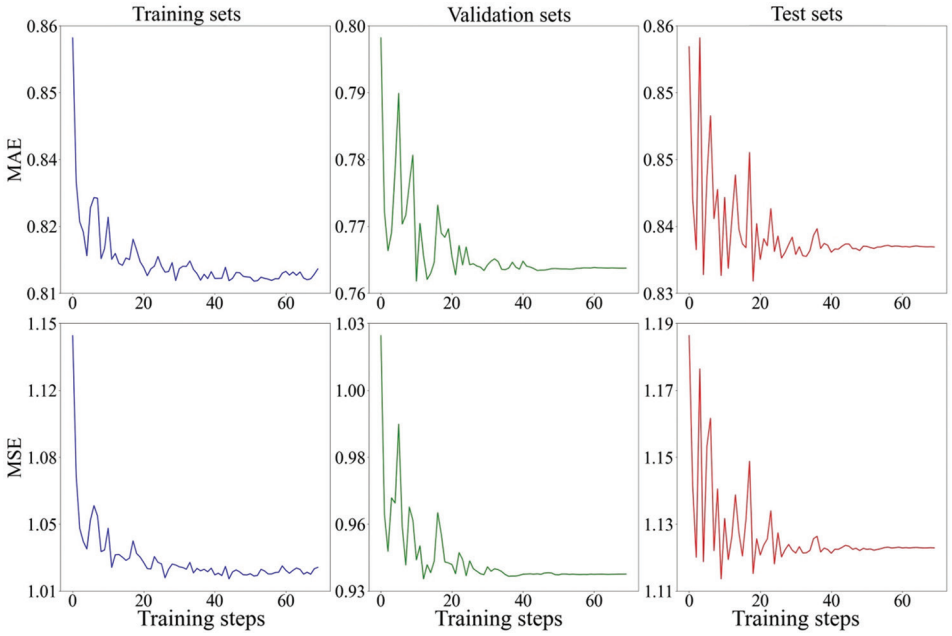

Figure S4. Training dynamics of the ANI model across different data partitions. The plots display the MAE (top row) and MSE (bottom row) versus training steps. Blue, green, and red lines represent the training, validation, and test sets, respectively.
Abbreviations: MAE: Mean absolute error; MSE: Mean squared error.

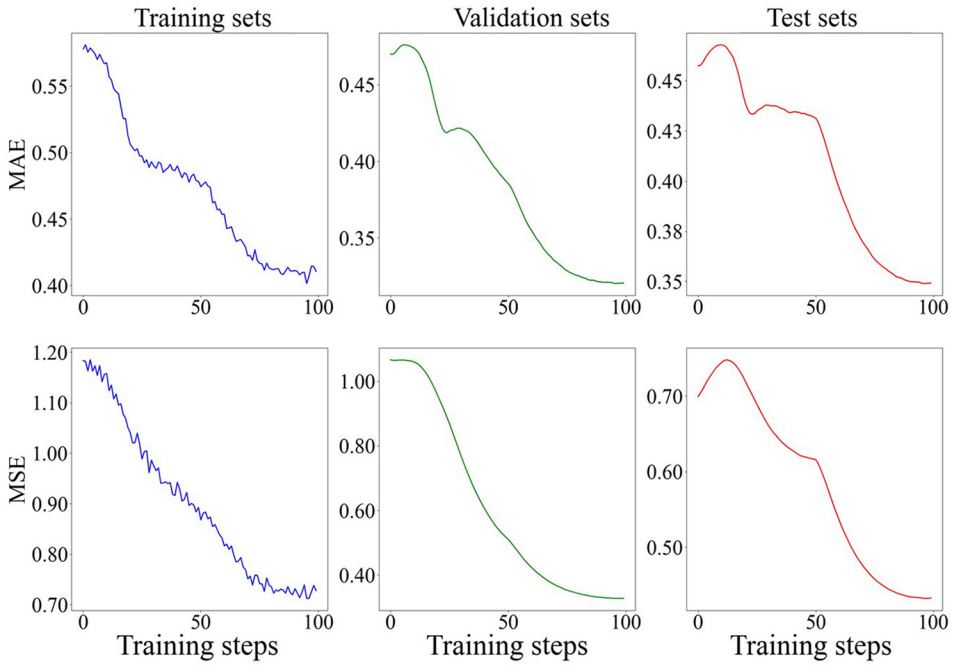

Figure S5. Training dynamics of the SchNet model across different data partitions. The plots display the MAE (top row) and MSE (bottom row) versus training steps. Blue, green, and red lines represent the training, validation, and test sets, respectively.
Abbreviations: MAE: Mean absolute error; MSE: Mean squared error.

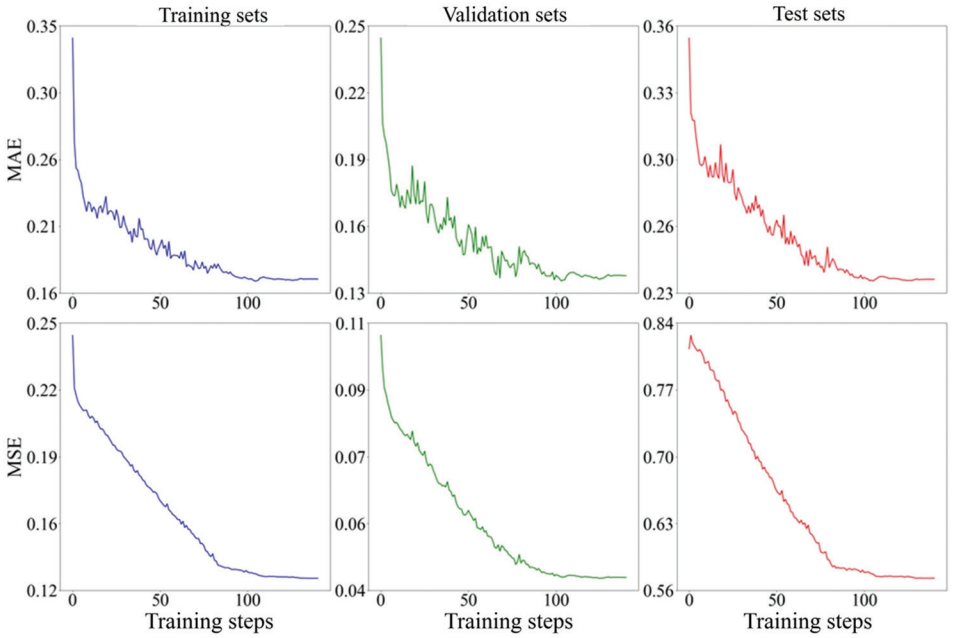

Figure S6. Training dynamics of the GNN model across different data partitions. The plots display the MAE (top row) and MSE (bottom row) versus training steps. Blue, green, and red lines represent the training, validation, and test sets, respectively.
Abbreviations: GNN: Graph neural network; MAE: Mean absolute error; MSE: Mean squared error.

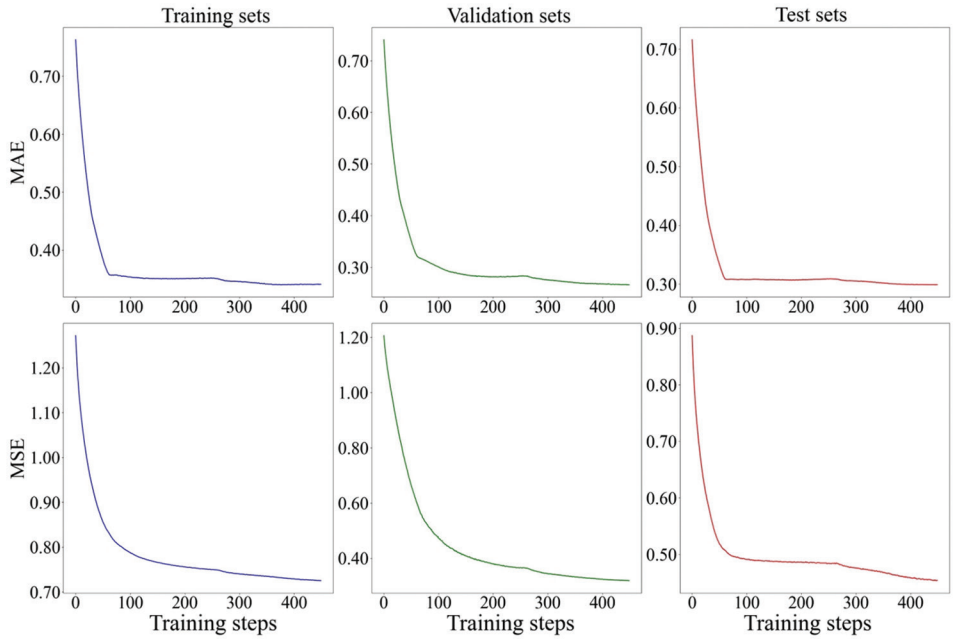

Figure S7. Training dynamics of the CalHousNet model across different data partitions. The plots display the MAE (top row) and MSE (bottom row) versus training steps. Blue, green, and red lines represent the training, validation, and test sets, respectively.
Abbreviations: MAE: Mean absolute error; MSE: Mean squared error.

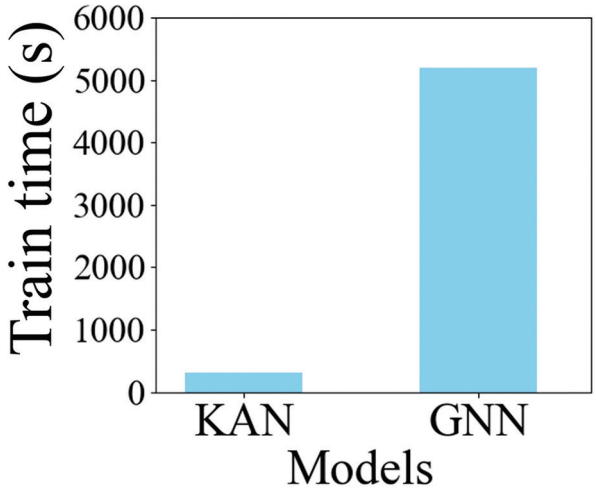

Figure S8. Comparative analysis of training time efficiency for the two machine learning models, KAN and GNN. The total training times for KAN and GNN are 362 and 5183 s, respectively.
Abbreviations: KAN: Kolmogorov-Arnold Network; GNN: Graph neural network.