Predict E_a using Regression Methods & Plot Potential Energy Surface (PES) using Neural Networks

Athish, Ciara, Nellie, Yuchen,Yuquan’s Group

Exercise 1: predict the activation energy E_a → regression

xGB

=== xGB Test Performance ===

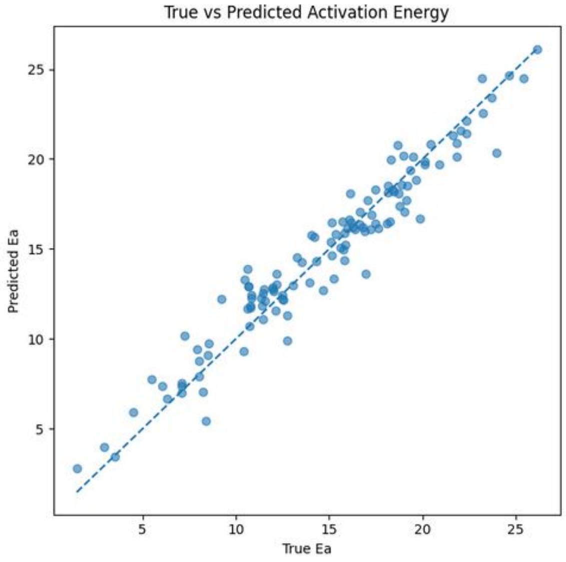
MSE: 1.7200

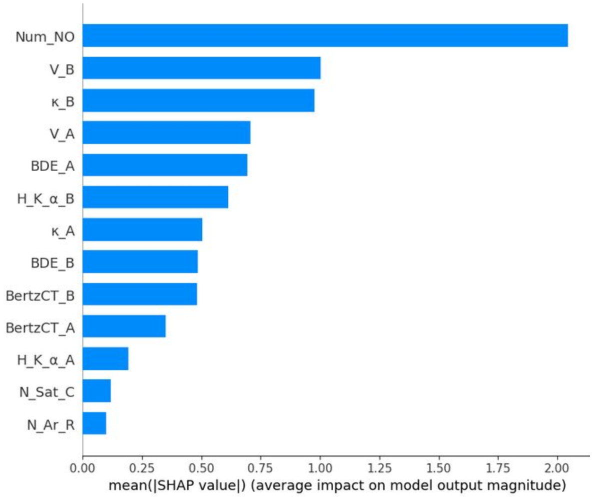
RMSE: 1.3115

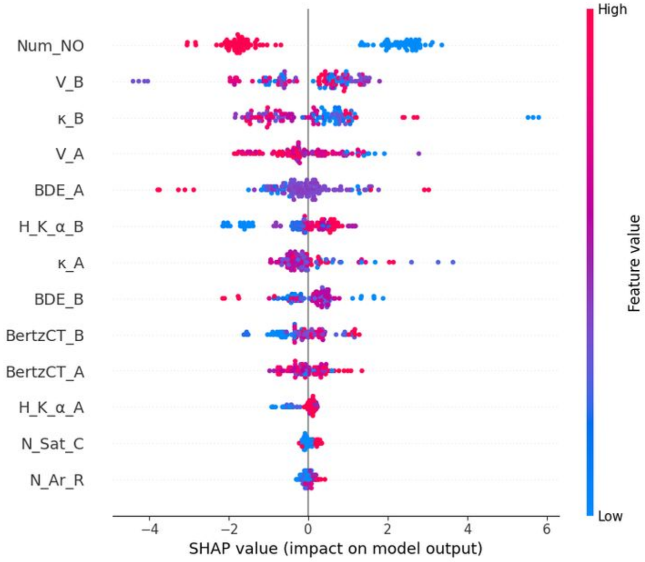
MAE: 1.0138

R^2: 0.9351

Explained Variance:0.9354



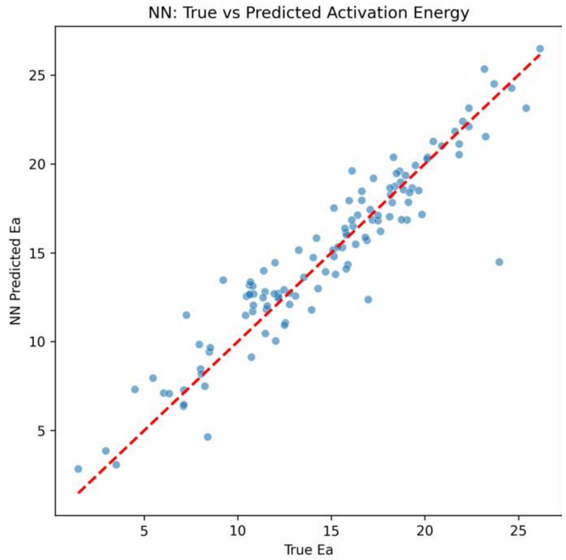


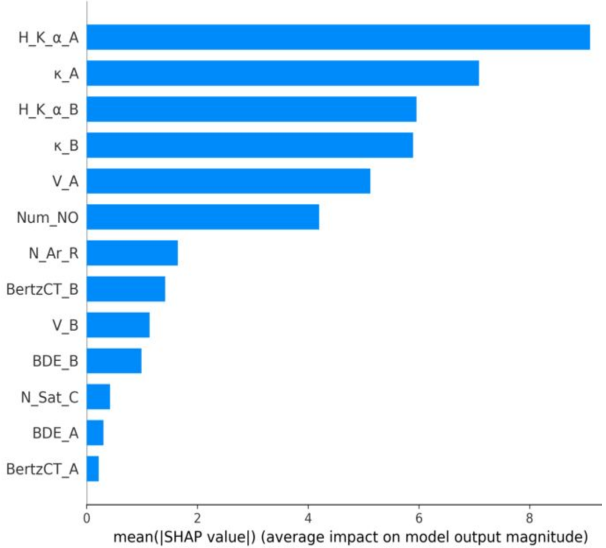


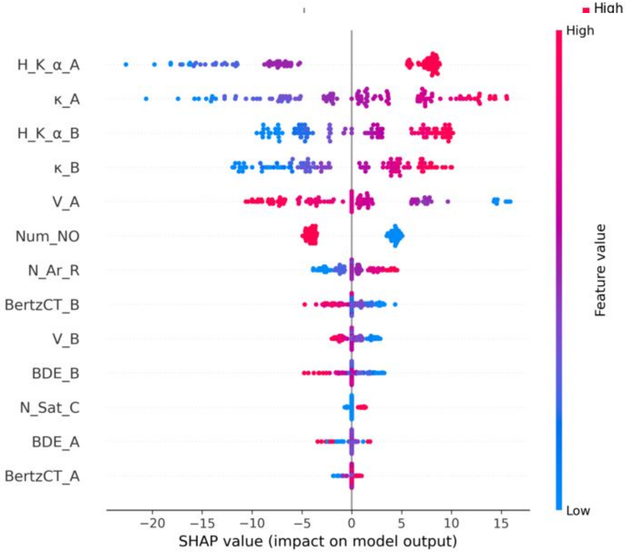
Exercise 1: predict the activation energy $E_a \rightarrow$ regression

NN

=== Neural Network Test Performance ===
MSE: 3.0426
RMSE: 1.7443
MAE: 1.2491
R2: 0.8852
Explained Variance:0.8866



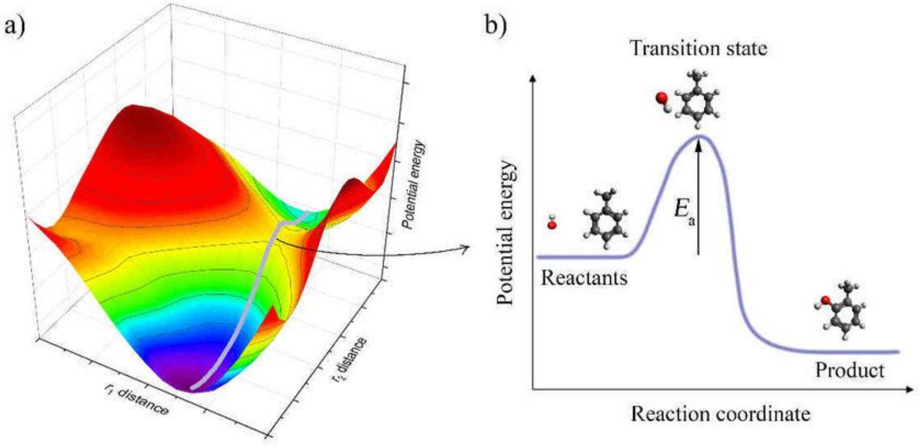




Decoded best hyperparameters: {'hidden_layer_sizes': (50,), 'activation': 'tanh', 'solver': 'adam', 'alpha': 1.960226457632454e-06, 'learning_rate': 'adaptive', 'learning_rate_init': 0.002587333173440179, 'momentum': 0.563152662892765, 'batch_size': 16, 'tol': 0.000411317705043 6119, 'max_iter': 500}

Exercise 4: potential energy surface using Neural Networks For the $OH + H_2 \rightarrow H + H_2O$ reaction

We are trying to train a neural network model to predict potential energy surfaces (PES) using atomic input features, aiming for minimal Mean Squared Error (MSE) between predicted and true energies.

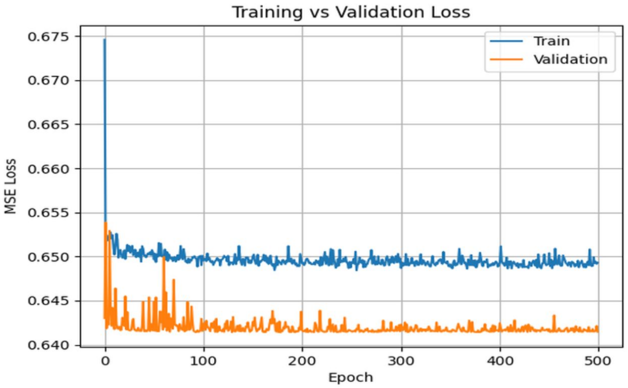


1.81760000	1.82050000	6.52810000	2.86390000	7.17980000	5.10420000	-0.70186338
2.40000000	2.41380000	14.11090000	3.79330000	14.19170000	14.14630000	1.96154491
1.81470000	1.81950000	7.26000000	2.86110000	7.80270000	6.27110000	-0.69932483
1.61380000	2.10000000	14.11010000	2.94180000	14.08980000	14.15890000	0.12738643
1.81530000	1.82230000	8.47490000	2.86390000	8.56210000	8.56250000	-0.69615850
1.61380000	1.81410000	14.10980000	2.70500000	14.08940000	14.13460000	-0.32127530
1.81490000	1.82020000	9.51850000	2.86100000	10.01790000	8.71170000	-0.69558146
1.81410000	2.11380000	14.11020000	3.10180000	14.13530000	14.11930000	-0.22066888
1.81380000	1.81410000	14.10990000	2.85980000	14.09930000	14.13490000	-0.69513468
1.81410000	2.41380000	14.11050000	3.35310000	14.13580000	14.14530000	0.68916935
2.10010000	2.11380000	14.11040000	3.32080000	14.15970000	14.11970000	0.20201330
1.81150000	1.81150000	7.81330000	2.85780000	8.57880000	9.17760000	-0.69768063
1.81150000	1.81150000	9.06780000	2.85780000	9.73510000	10.52020000	-0.69623675
1.81150000	1.81150000	10.38750000	2.85780000	10.97490000	11.90040000	-0.69561508
1.81140000	1.81140000	6.35660000	2.85740000	7.21240000	7.55960000	-0.70116419
1.81150000	1.81150000	6.66120000	2.85780000	7.54450000	7.89190000	-0.70039345
1.83050000	1.94600000	6.16480000	2.27000000	5.82630000	4.42910000	0.03845535
1.82200000	2.05890000	7.99630000	3.84720000	7.97230000	8.48890000	1.08672184
1.91850000	1.99350000	6.58560000	2.36310000	6.29710000	7.73250000	0.09642254
1.74260000	2.00330000	8.12910000	3.63720000	7.50770000	9.21380000	0.61451170
1.70710000	1.78740000	6.99370000	2.12070000	6.62310000	6.82400000	0.16749580
1.86160000	2.12640000	10.16300000	2.21640000	10.16810000	10.18140000	0.63370985
2.12220000	2.12790000	6.53970000	3.50270000	8.05130000	7.35470000	0.36459586
1.90270000	2.00020000	7.84850000	2.28420000	8.39880000	8.52260000	0.21780781
2.04810000	2.22490000	7.16200000	3.49030000	5.47140000	7.93070000	0.43574887
1.88700000	2.24250000	8.64860000	3.24630000	7.61640000	8.52460000	0.17267999
1.90860000	2.05440000	9.94950000	3.48850000	10.07030000	10.07460000	-0.02996234
2.07220000	2.09780000	6.92440000	3.27550000	5.68300000	7.44480000	0.08341985
2.11220000	2.12300000	11.27880000	3.47270000	11.37590000	11.37630000	0.31923565
1.70880000	1.86650000	9.17560000	3.32000000	10.21850000	8.37920000	-0.06977168

Parameters

- **Input layer** takes 6 features: the inverse distances 1/R for the three O–H and three H–H distances.
- **Hidden layers**: two fully-connected layers of width 64, each followed by a ReLU nonlinearity.
- **Output layer** maps those 64 hidden activations down to a single scalar
- **Optimizer:** Adam with learning rate 10−3
- **Loss:** Mean squared error
- **Batch sizes:** 32 for training, 64 for validation.
- **Epochs:** 200, 500

Training with parameter R

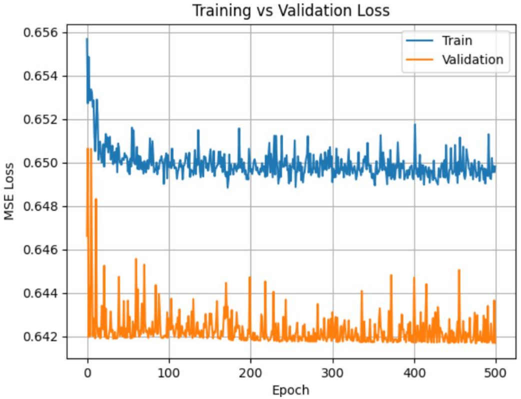


ReLU functions

$f(x)=\max(0,x)$
$f'(x)=I\ (x>0)$

tansig functions

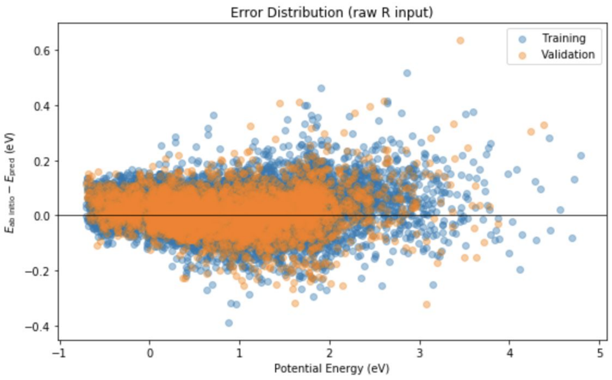
$f(x)=2/(1+\exp(-2^n))-1$
$f'(x)=1/(1+\exp(-x))$

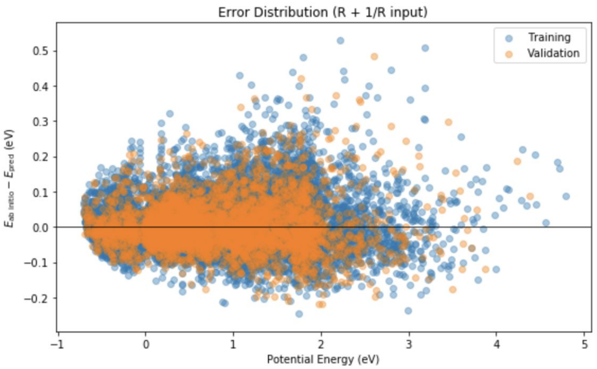


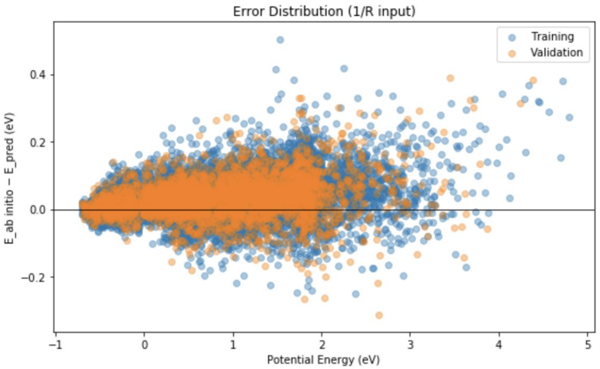
logsig functions

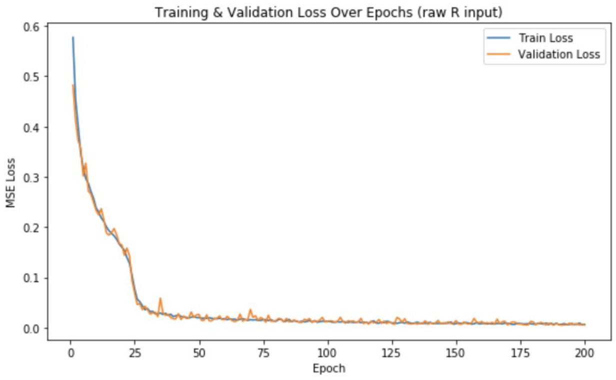
$f(x)=1/(1+\exp(-x))$
$f'(x)=\exp(x)/(1+\exp(x))^2$

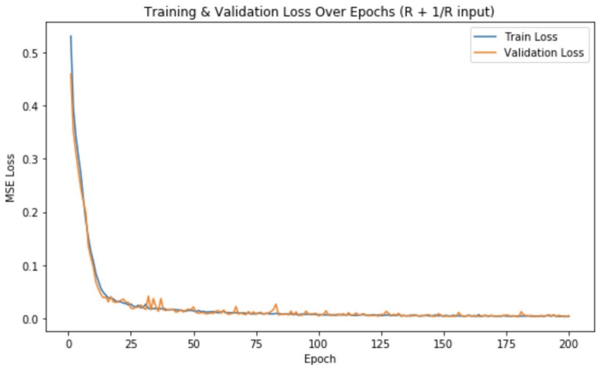
Results

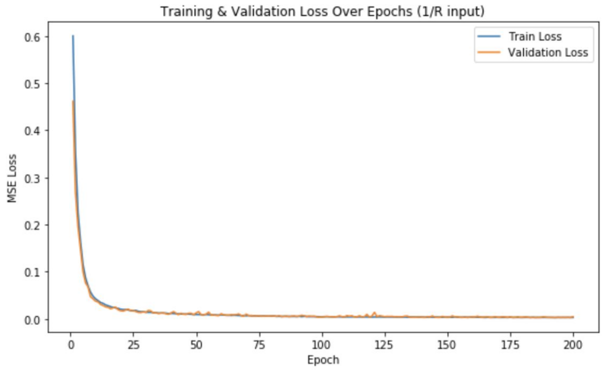

Error Distribution (raw R input)
Training
Validation
$E_{ab\ initio} - E_{pred}$ (eV)
Potential Energy (eV)


Error Distribution (R + 1/R input)
Training
Validation
$E_{ab\ initio} - E_{pred}$ (eV)
Potential Energy (eV)

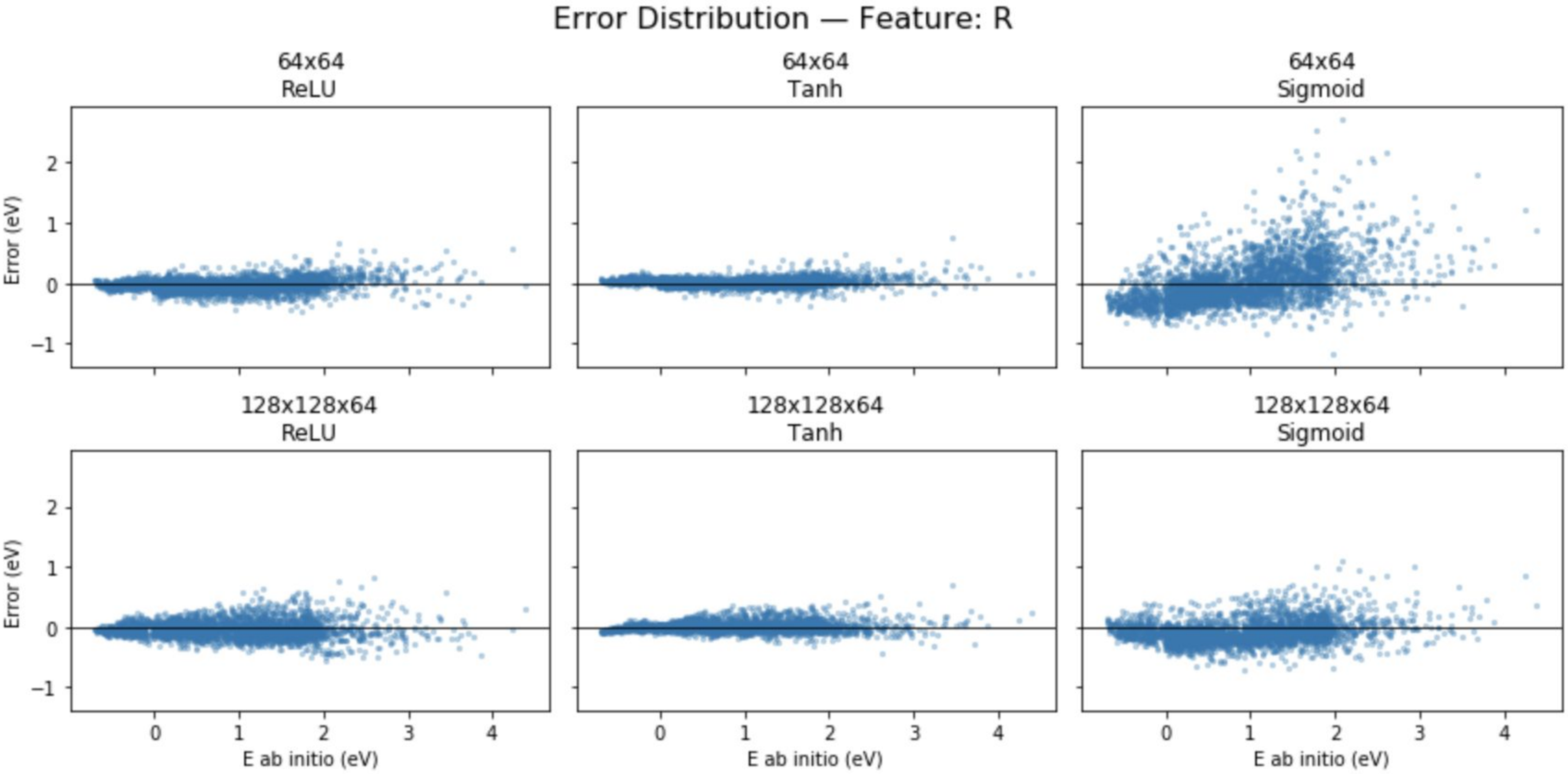

Error Distribution (1/R input)
Training
Validation
E_ab initio − E_pred (eV)
Potential Energy (eV)

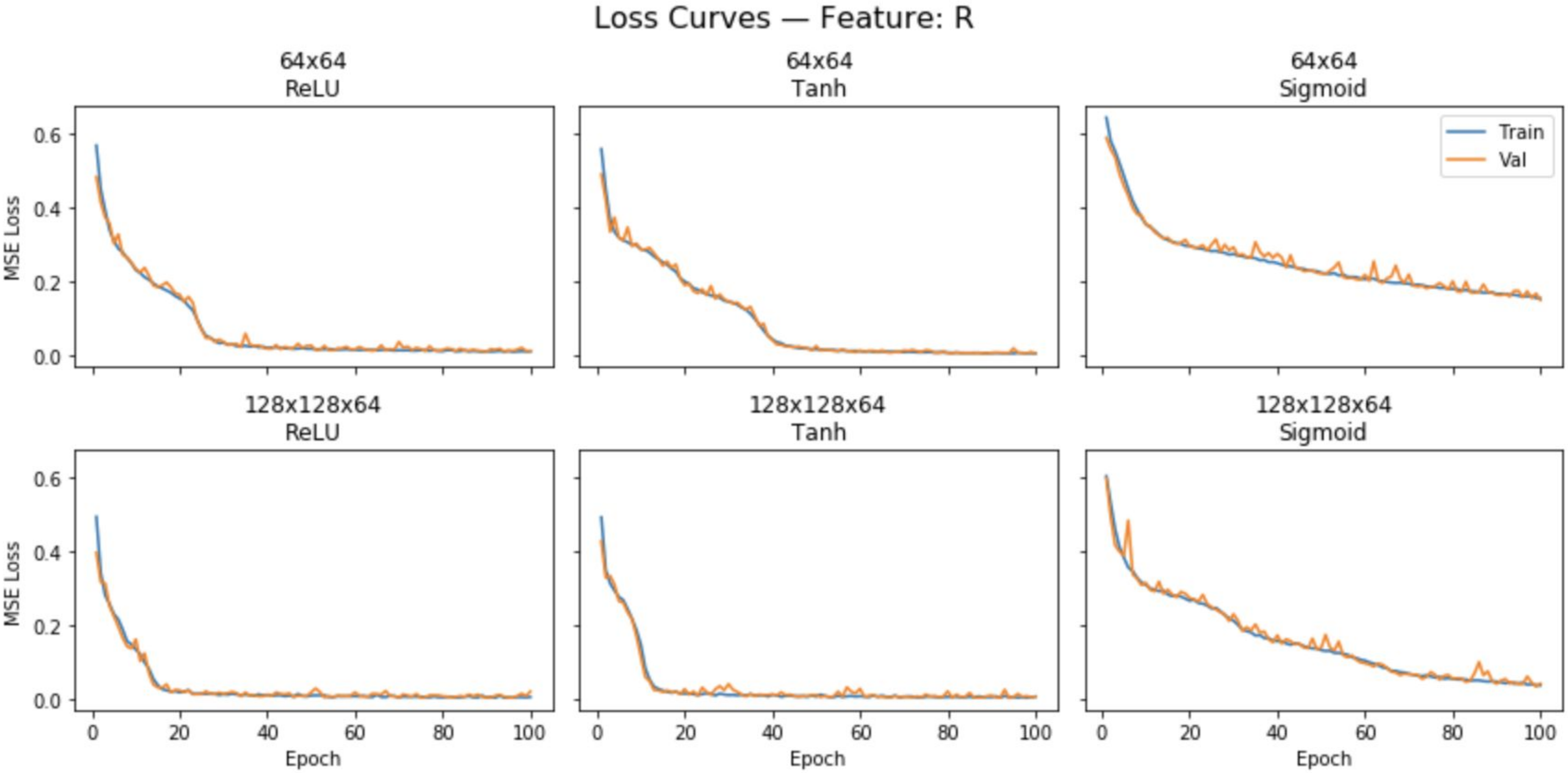

Training & Validation Loss Over Epochs (raw R input)
Train Loss
Validation Loss
MSE Loss
Epoch


Training & Validation Loss Over Epochs (R + 1/R input)
Train Loss
Validation Loss
MSE Loss
Epoch

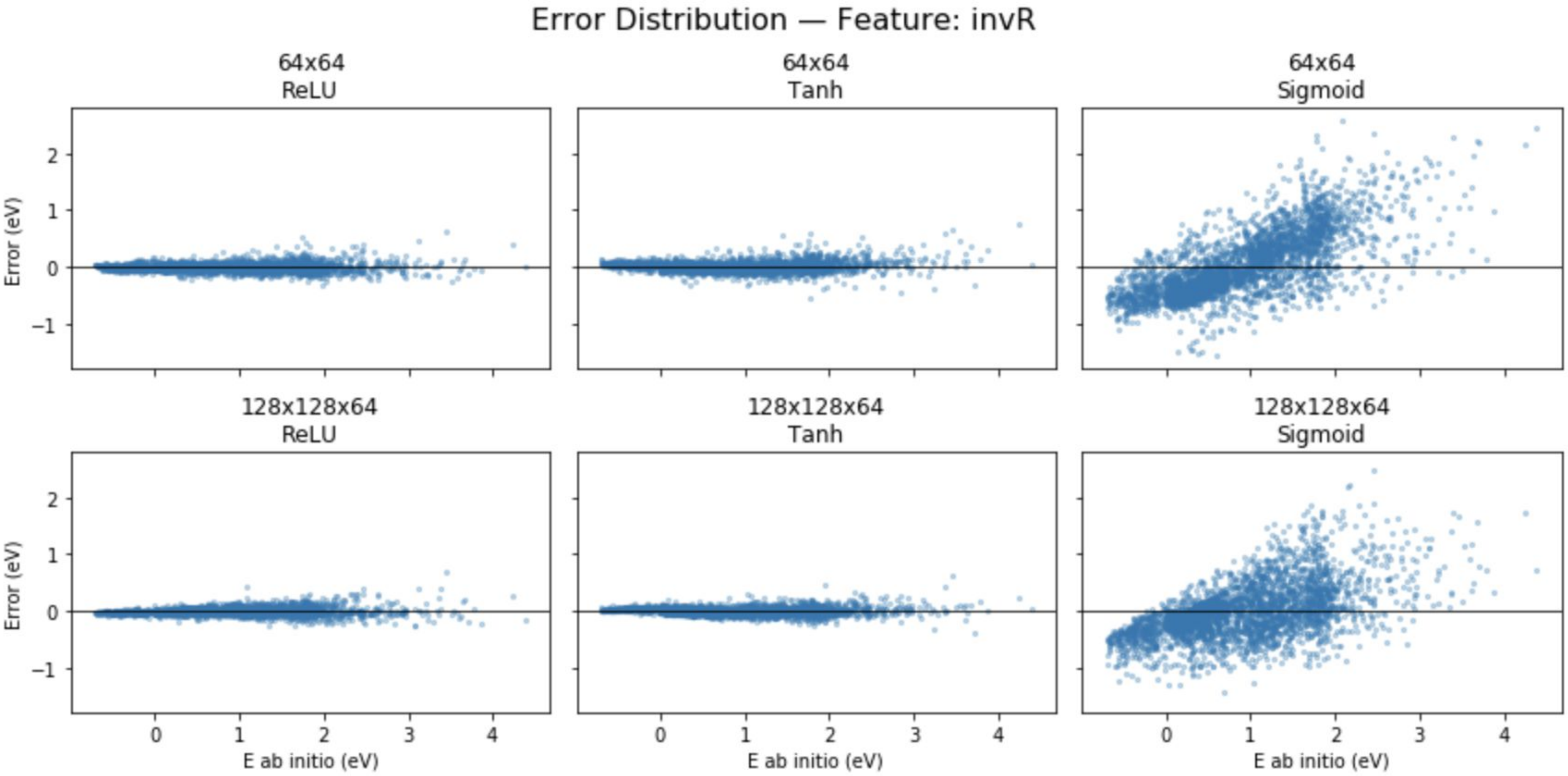

Training & Validation Loss Over Epochs (1/R input)
Train Loss
Validation Loss
MSE Loss
Epoch

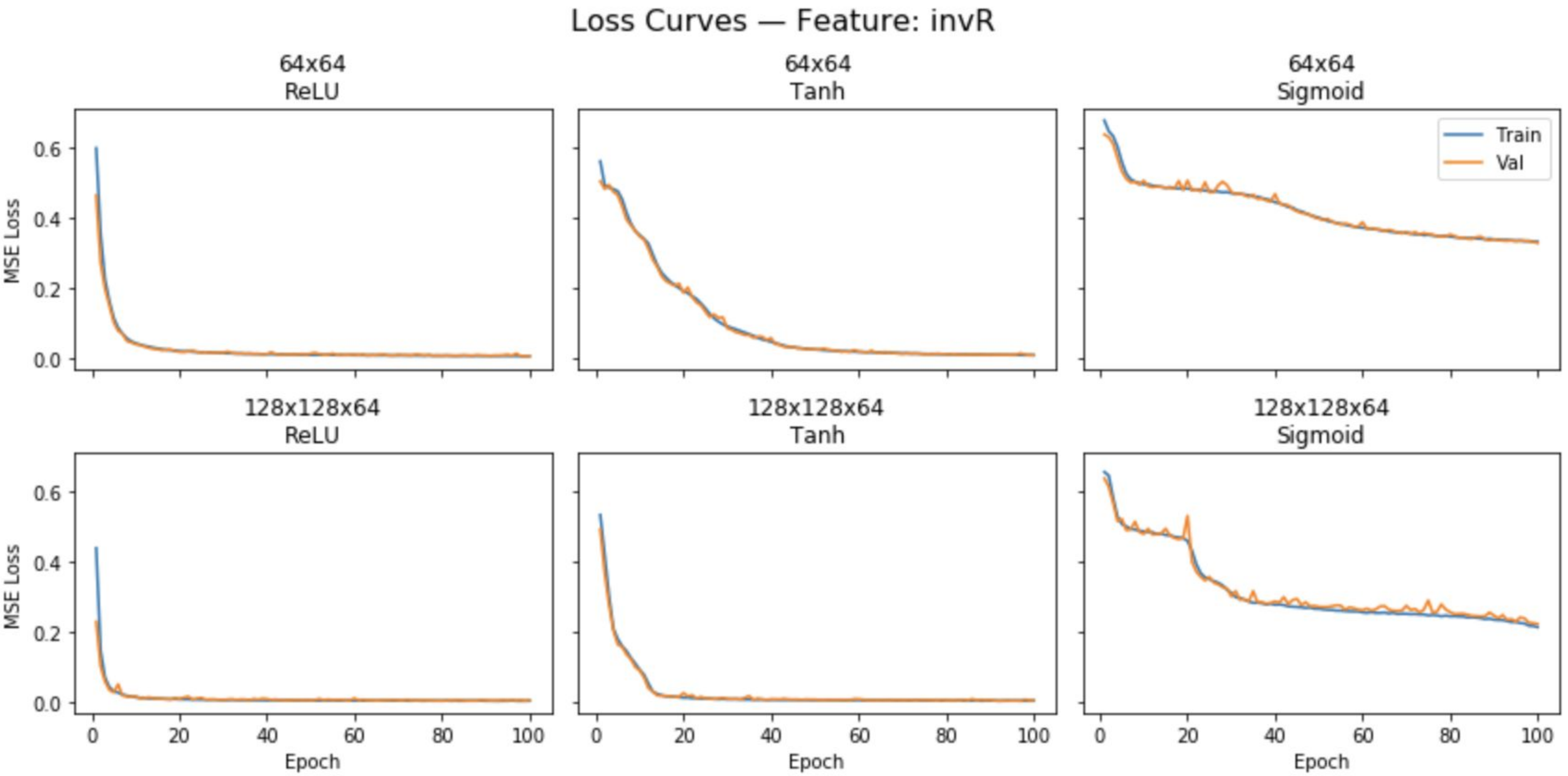
```python
# 2) Feature transforms
FEATURES = {
    'R':          lambda R: R,
    'invR':       lambda R: 1.0/R,
    'R+invR':     lambda R: np.hstack((R, 1.0/R)),
    'expR(a=1)': lambda R: np.exp(-1.0 * R),
}

# 3) Architectures & activations
ARCHS = {
    '64x64':       [64,64],
    '128x128x64': [128,128,64],
}
ACTIV = {
    'ReLU':    nn.ReLU,
    'Tanh':    nn.Tanh,
    'Sigmoid': nn.Sigmoid,
}
```

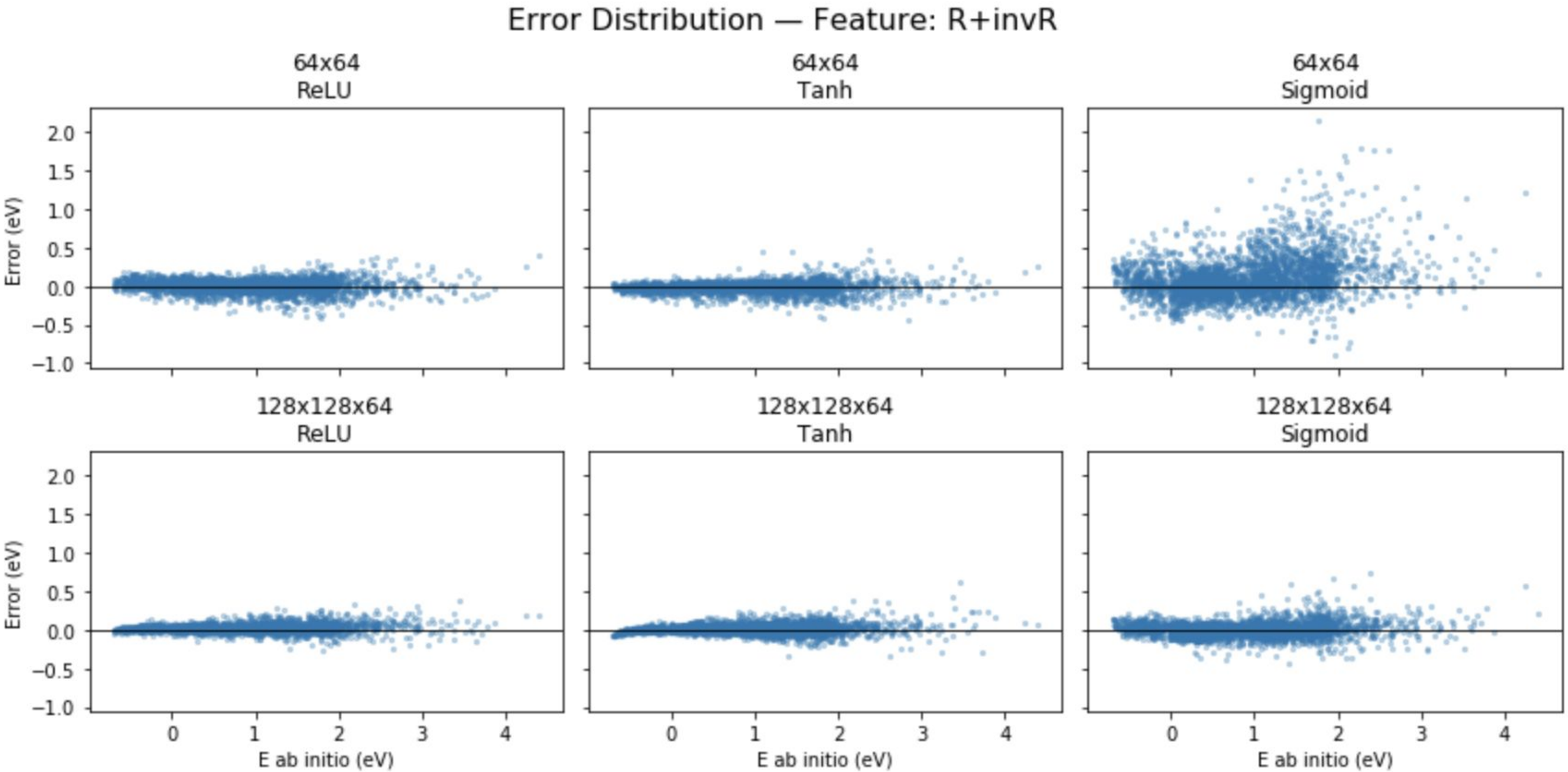

Error Distribution — Feature: R
64x64
ReLU
64x64
Tanh
64x64
Sigmoid
128x128x64
ReLU
128x128x64
Tanh
128x128x64
Sigmoid
Error (eV)
E ab initio (eV)

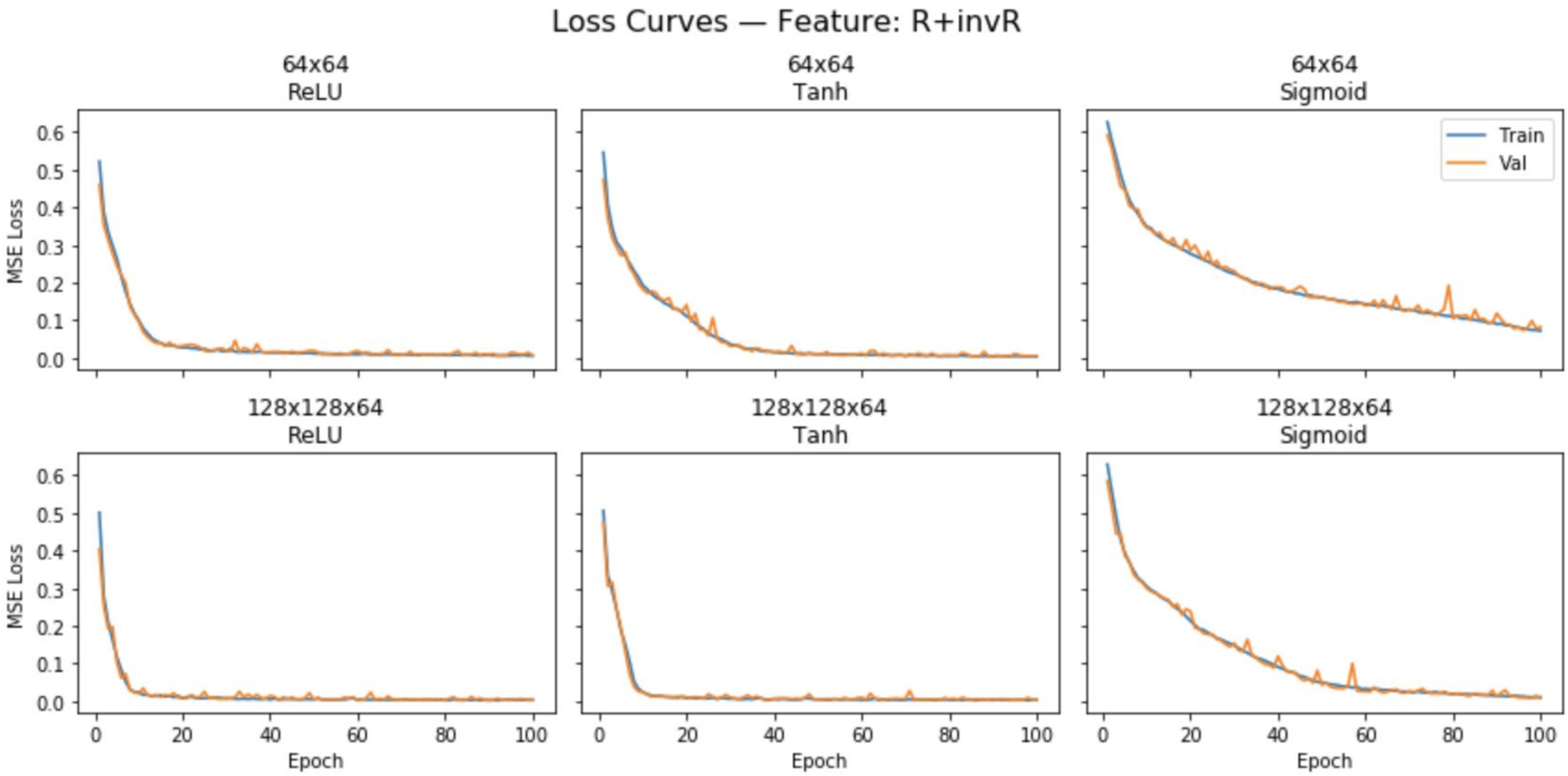


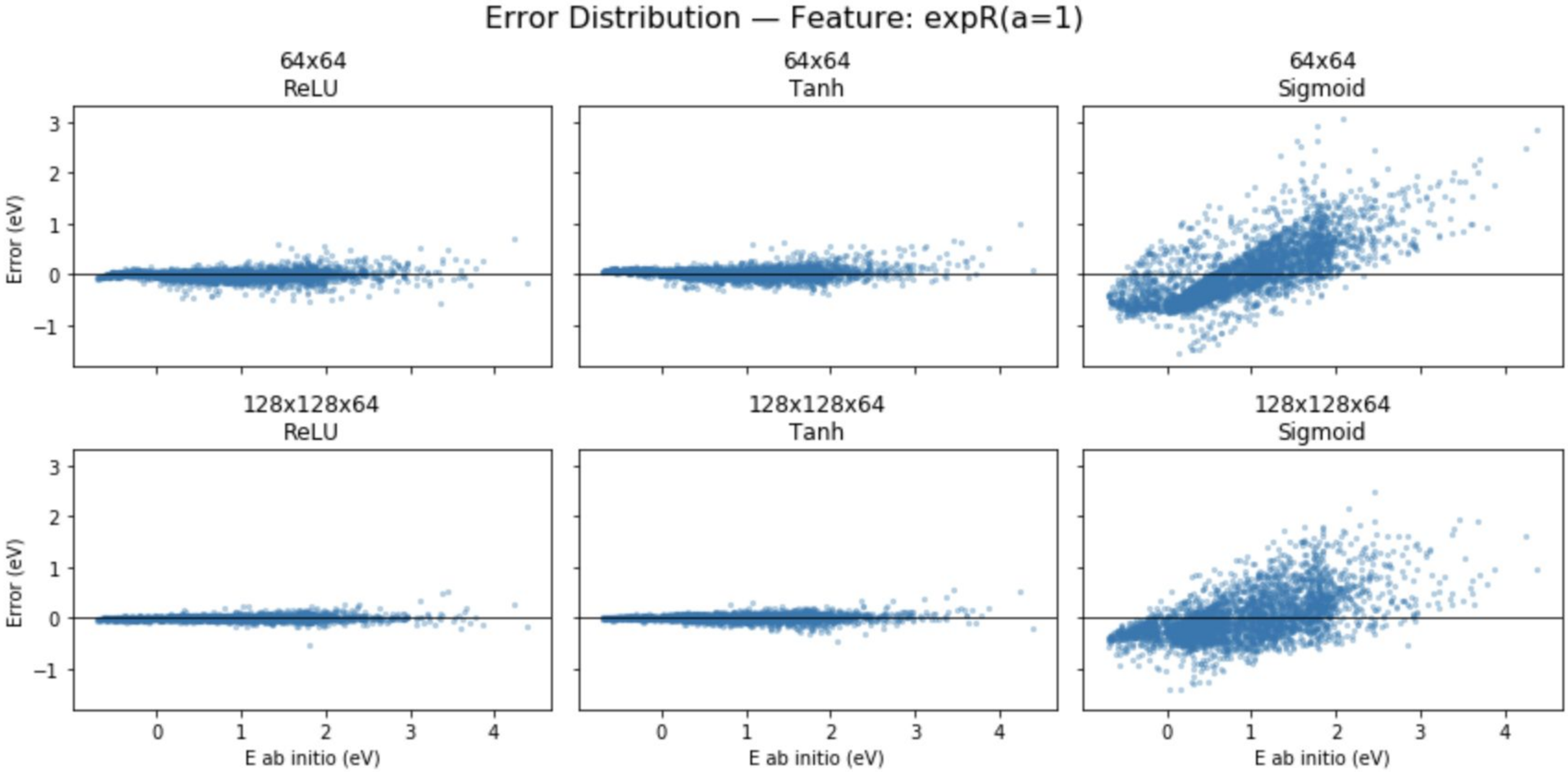
Loss Curves — Feature: R
64x64
ReLU
64x64
Tanh
64x64
Sigmoid
Train
Val
128x128x64
ReLU
128x128x64
Tanh
128x128x64
Sigmoid
MSE Loss
Epoch
0.0
0.2
0.4
0.6
0
20
40
60
80
100

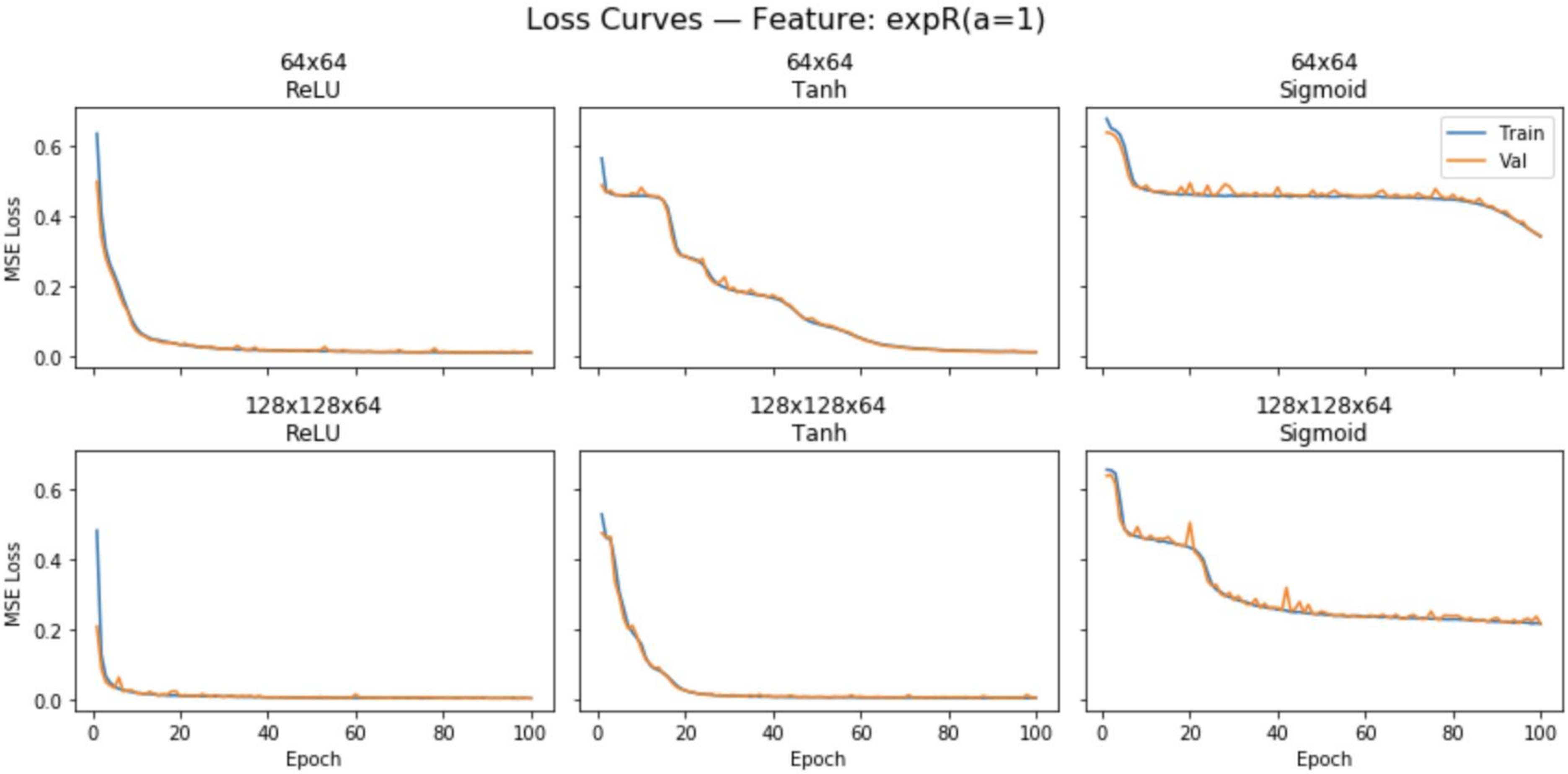

Error Distribution — Feature: invR
64x64
ReLU
64x64
Tanh
64x64
Sigmoid
128x128x64
ReLU
128x128x64
Tanh
128x128x64
Sigmoid
Error (eV)
Error (eV)
E ab initio (eV)
E ab initio (eV)
E ab initio (eV)


Loss Curves — Feature: invR
64x64
ReLU
64x64
Tanh
64x64
Sigmoid
128x128x64
ReLU
128x128x64
Tanh
128x128x64
Sigmoid
Train
Val
MSE Loss
Epoch
0.0
0.2
0.4
0.6
0
20
40
60
80
100


Error Distribution — Feature: R+invR
64x64
ReLU
64x64
Tanh
64x64
Sigmoid
128x128x64
ReLU
128x128x64
Tanh
128x128x64
Sigmoid
Error (eV)
E ab initio (eV)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4


Loss Curves — Feature: R+invR
64x64
ReLU
64x64
Tanh
64x64
Sigmoid
128x128x64
ReLU
128x128x64
Tanh
128x128x64
Sigmoid
Train
Val
MSE Loss
Epoch
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
20
40
60
80
100


Error Distribution — Feature: expR(a=1)
64x64
ReLU
64x64
Tanh
64x64
Sigmoid
128x128x64
ReLU
128x128x64
Tanh
128x128x64
Sigmoid
Error (eV)
E ab initio (eV)
3
2
1
0
−1
0
1
2
3
4



Loss Curves — Feature: expR(a=1)
64x64
ReLU
64x64
Tanh
64x64
Sigmoid
Train
Val
128x128x64
ReLU
128x128x64
Tanh
128x128x64
Sigmoid
MSE Loss
Epoch
0.0
0.2
0.4
0.6
0
20
40
60
80
100