Representation Learning for Long-Chain Hydrocarbon Adsorption in Zeolites

Yachan Liu,[†] Ping Yang,[†] Gustavo Perez,[‡] Aaron Sun,[‡] Wei Fan,[†] Subhransu Maji,[*,‡] and Peng Bai[*,†]

[†]*Department of Chemical Engineering, University of Massachusetts Amherst, Amherst, Massachusetts 01003, United States of America*

[‡]*College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst, Massachusetts 01003, United States of America*

E-mail: smaji@cs.umass.edu; pengbai@umass.edu

Phone: +1 413 545 6189. Fax: +1 413 545 1647

Details of Model Architecture

Crystal Graph Convolutional Neural Networks (CGCNN)[1] The CGCNN architecture used in this work consists of 1 embedding layer, 3 – 6 graph convolutional layers, 1 pooling layer, 1 hidden layer, and 1 output layer. The embedding layer is a fully connected (FC) layer with an input dimension of 92 and an output dimension of 64, denoted as $\mathrm{FC}(92, 64)$. Each convolutional layer utilizes a message-passing neural network[2] design and gated convolutions, consisting of a concatenation of node hidden vectors, neighbors hidden vectors and distance feature vector, a $\mathrm{FC}(169, 128)$ layer followed by batch normalization, a gated mechanism using softplus activation and sigmoid activation separately on the split core part and the filter part, batch normalization on the sum of element-wise multiplication of filter and core parts, and another softplus activation applied on updated node features. The pooling layer is inserted after all convolutional layers, aggregating node (atom) features into graph (crystal) features via normalized summation. After the pooling operation, a $\mathrm{FC}(64, 128)$ hidden layer followed by an output layer generates the prediction for the input crystal structure.

MatErials Graph Network (MEGNet)[3] The MEGNet architecture used in this work consists of 1 embedding block, 3 – 6 MEGNet blocks, 1 aggregation block, and 1 output block. The embedding block comprises FC layers to encode node ($89 \rightarrow 16 \rightarrow 64 \rightarrow 32$), edge ($100 \rightarrow 64 \rightarrow 32$), and state features ($2 \rightarrow 64 \rightarrow 32$). Each MEGNet block is formed by one graph convolutional layer and two FC layers ($32 \rightarrow 64 \rightarrow 32$) (except for the first MEGNet block). The graph convolutional layer applies three hidden layers as node ($128 \rightarrow 64 \rightarrow 64 \rightarrow 32$), edge ($96 \rightarrow 64 \rightarrow 64 \rightarrow 32$), and state ($96 \rightarrow 64 \rightarrow 64 \rightarrow 32$) update functions. The aggregation block contains two or one set2set networks[4] for node and edge, respectively, both of which employ a single-layer Long-Short Term Memory (LSTM) Recurrent Neural Network (RNN) with an input size of 32 and an output size of 64, to reduce the sets of atomic and bond vectors into a single vector. The final output block takes the concatenated vectors of node, bond, and state as input and processes them through three FC layers (160 → 64 → 32 → 1). All FC layers employ the modified softplus activation function.

Graph Neural Networks with 3-body Interactions (M3GNet)[5] The M3GNet architecture used in this work consists of 1 featurization block, 3 M3GNet blocks, 1 readout block, and 1 output block. The featurization block is formed by a graph featurizer and a three-body computation module. The graph featurizer encodes the atomic number of elements, and node (89 → 64) and edge (9 → 64) features, which are pair distances expanded in spherical Bessel functions with maximum orders of radial part $n = 3$ and angular part $l = 3$. The three-body computation module captures the angular dependencies in local atomic environments by the construction of a line graph, where a node corresponds to a bond in the original graph and an edge represents a valid triplet of atoms defined by a common central atom. For each triplet, the cosine of the bond angle is computed and expanded by spherical harmonics up to an angular order of $l = 3$. Simultaneously, the central bond length in each triplet is encoded using spherical Bessel functions up to radial order $n = 3$, and angular order $l = 3$. Each M3GNet block includes a three-body-to-bond module and a graph convolutional module. The three-body-to-bond module employs a $\mathrm{FC}(64, 9)$ layer followed by sigmoid activation for encoding directional contributions from neighbor atoms and a gated $\mathrm{FC}(9, 64)$ using SiLU and sigmoid activations separately for the main and gating paths to update bond-level features. The resulting enriched edge features, along with the node features, are then passed into the graph convolutional module. The graph convolutional module in each block updates node and edge features through gated MLPs with three hidden layers (192 → 64 → 64 → 64) followed by SiLU and sigmoid (only for the last layer in gating paths) activations, coupled with a $\mathrm{FC}(9, 64)$ for additional edge and node weights learned via linear projections from the radial basis inputs. Following the M3GNet blocks, the updated atomic features are aggregated into a graph-level representation via a weighted atom-wise readout block. This block applies two FC layers (64 → 64 → 64) followed by SiLU activations to node features and computes attention weights using a FC$(64, 1)$ layer followed by sigmoid activation, enabling adaptive contributions from different atoms. The final output block consists of three FC layers (64 → 64 → 64 → 1) with SiLU activations to sum the weighted information and output the target property.

Multi Atomic Cluster Expansion (MACE)[6] The MACE architecture used in this work consists of 1 embedding block, 1 interactions block, 1 product block, and 1 readout block. The embedding block initializes node features using the one-hot encoding of atomic numbers, edge features using radial embedding to encode the length of the edges as a 10-dimensional vector using a set of 10 Bessel functions with a predefined cutoff, and edge attributes using angular embedding to encode the direction of the edges as a 16-dimensional vector using spherical harmonics with an expansion order of 3. The interaction block computes equivariant atomic interactions by using two FC layers to map node features and edge features to two 128-dimensional spaces, which are then combined with the edge attributes to output a set of updated node features with a dimension of $(N_{\text{atom}}, 128, 16)$. The product block further refines node features to construct higher-order features by first forming tensor products of the features and then applying symmetric contractions. The last part of the product block is to mix everything linearly and create a new set of node features with an output dimension of 128. The readout block finally maps the node features to scalar properties using a FC$(128, 1)$ layer followed by SiLU activations. The interaction, product, and readout are repeated twice, and all the node features are summed to compute the final graph-level output.

Additional Tables

Table S1: Validation performance of 2D ResNet18, showing r^2 and MSE for input of (a) 2D distance images, (b) 3D volumetric grids, and (c) 2D multi-view distance images.

(a) 2D distance images				
	MSE	r^2		
	40.3	**0.758**		
(b) 3D volumetric grids				
	MSE	r^2		
With a linear adapter	22.7	0.864		
W/o linear adapters	**17.1**	**0.897**		
(c) 2D multi-view distance images				
Num of views	Three axes		A single fixed axis	
	MSE	r^2	MSE	r^2
10	21.2	0.873	29.5	0.823
9	20.8	0.875	28.9	0.826
8	20.0	0.880	**26.3**	**0.842**
7	**19.8**	**0.881**	32.5	0.805
6	20.1	0.879	27.8	0.833
5	22.2	0.866	31.1	0.813
4	21.8	0.869	33.6	0.799
3	24.2	0.855	38.5	0.769
2	26.2	0.842	45.2	0.728
1	34.2	0.795	56.0	0.664

Table S2: Validation performance of PointNet and EdgeConv trained on atoms within the unit cell. Atoms are duplicated, if applicable, to reach the specified number of points.

Num of points	PointNet		EdgeConv	
	r^2	MSE	r^2	MSE
128	0.391	101.6	0.264	122.7
256	0.418	97.1	0.320	113.4
512	0.437	93.9	0.335	110.8
1024	0.441	93.2	0.340	110.0
2048	0.435	94.1	0.326	112.4
4096	0.451	91.5		

Table S3: Validation performance of PointNet and EdgeConv trained on atoms within cubes of increasing size. Atoms are duplicated, if applicable, to reach the specified number of points.

Linear dimension (Å)	Num of points	PointNet		EdgeConv	
		r^2	MSE	r^2	MSE
30	256	0.218	130.4	0.175	137.5
	512	0.425	95.9	0.391	101.5
	1024	0.407	98.9	0.430	95.0
	2048	0.479	86.9	0.610	65.1
45	1024	0.235	127.6	0.300	116.7
	2048	0.382	103.1	0.536	77.4
60	1024	0.100	150.1	0.172	138.0
	2048	0.172	138.0	0.257	123.8

Table S4: Validation performance of 3D ResNet18 with single-channel distance grids trained from scratch using the full dataset or the transfer-learning subset for the adsorption of six alkane molecules.

	Full set		Subset	
	r^2	MSE	r^2	MSE
C18	0.970	4.9	0.897	3.8
C24	0.942	11.5	0.883	9.9
C30	0.809	37.5	0.738	52.0
2C17	0.965	5.8	0.887	4.9
4C17	0.963	5.9	0.882	6.0
22C16	0.958	6.9	0.918	7.3

Table S5: Validation accuracy of MEGNet with different model settings. The default setting and the best-performing setting are indicated in bold. $\boldsymbol{\Delta\mu = (r_{\text{cut}} + 1) / (N_b - 1)}$.

r_{cut}	N_b	N_{block}	r^2	MSE
4	**100**	**3**	0.241	127
4	40	3	0.235	128
4	20	3	0.024	163
4	10	3	0.214	131
6	100	3	0.224	129
7	100	3	0.554	74.4
8	100	3	**0.795**	**34.2**
9	100	3	0.630	61.7
10	100	3	0.356	108
8	100	4	0.781	36.6
8	100	5	0.315	114
8	100	6	0.304	116

Table S6: Validation accuracy of M3GNet with different model settings. The default setting and the best-performing setting are indicated in bold.

r_{cut}	$r_{3-\text{body}}$	r^2	MSE
4	4	0.392	101
5	**4**	0.426	95.7
6	4	0.610	65.0
7	4	0.795	34.2
8	4	0.857	23.9
9	4	0.875	20.9
10	4	**0.894**	**17.7**
10	3	0.878	20.3
10	5	0.879	20.2
10	6	0.882	19.6

Table S7: Validation accuracy of MACE with different model settings. The default setting and the best-performing setting are indicated in bold.

r_{cut}	r^2	MSE
4	0.305	116
5	0.544	76.0
6	0.700	50.0
7	0.794	34.3
8	**0.810**	**31.7**

Additional Figures

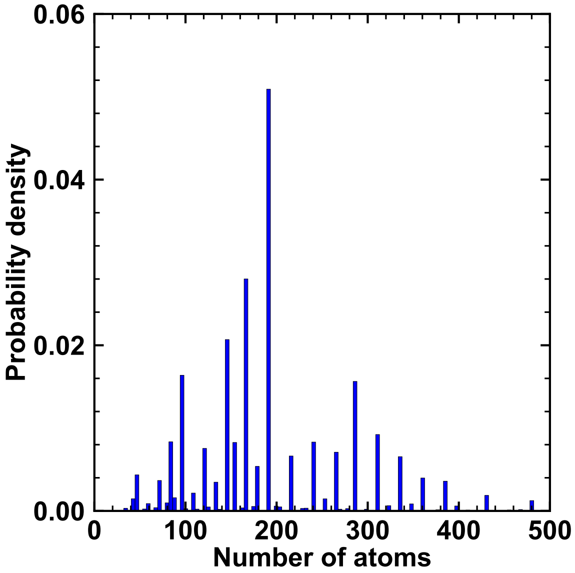


Figure S1: The distribution of the number of atoms in a unit cell. The minimum, maximum, mean, median and standard deviation are 24, 2352, 199, 192, and 99, respectively.

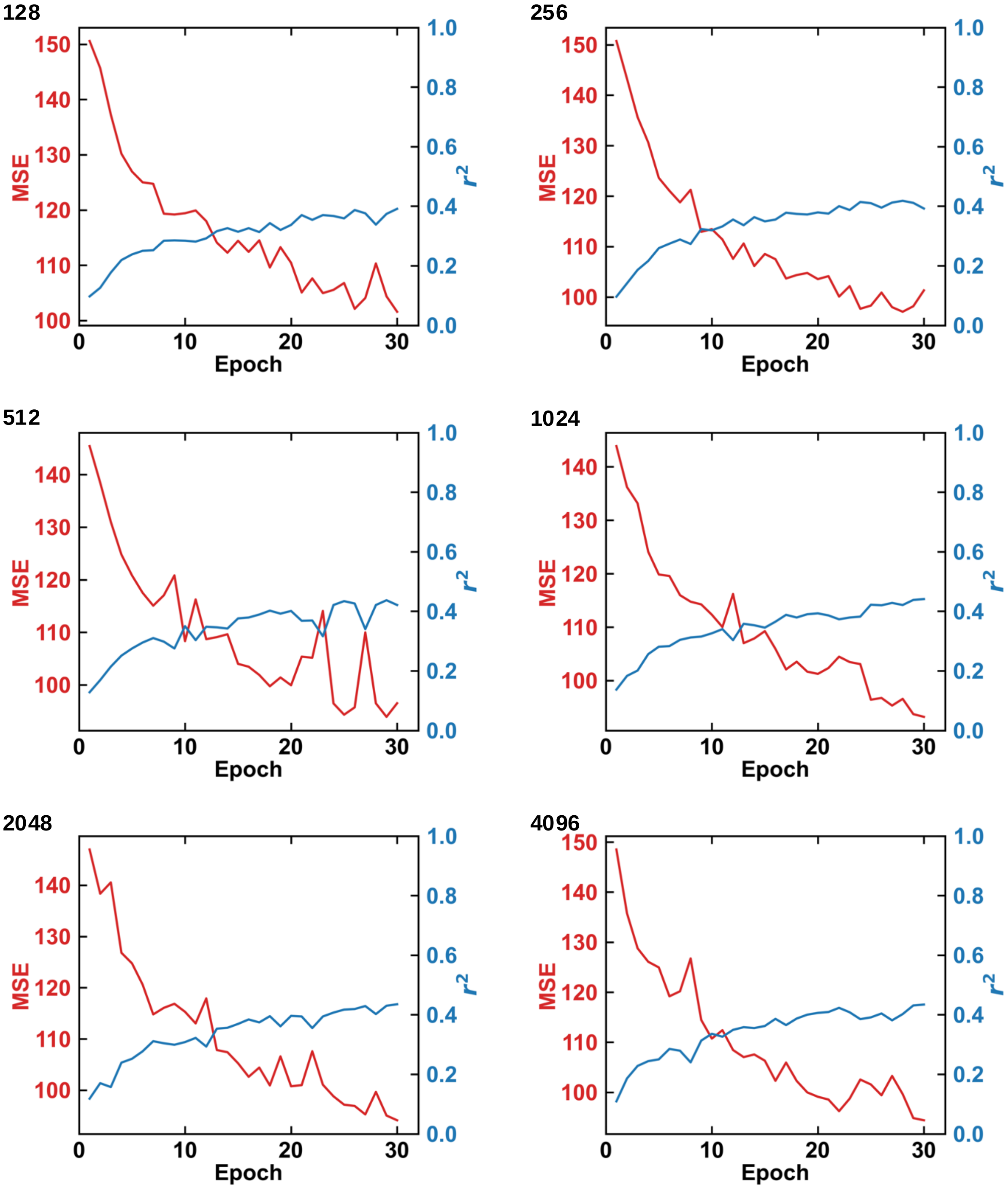


Figure S2: r^2 and MSE as a function of epochs during training of PointNet on atom-coordinate point clouds. The numbers on the top left corner of each panel indicate the number of points included from a unit cell.

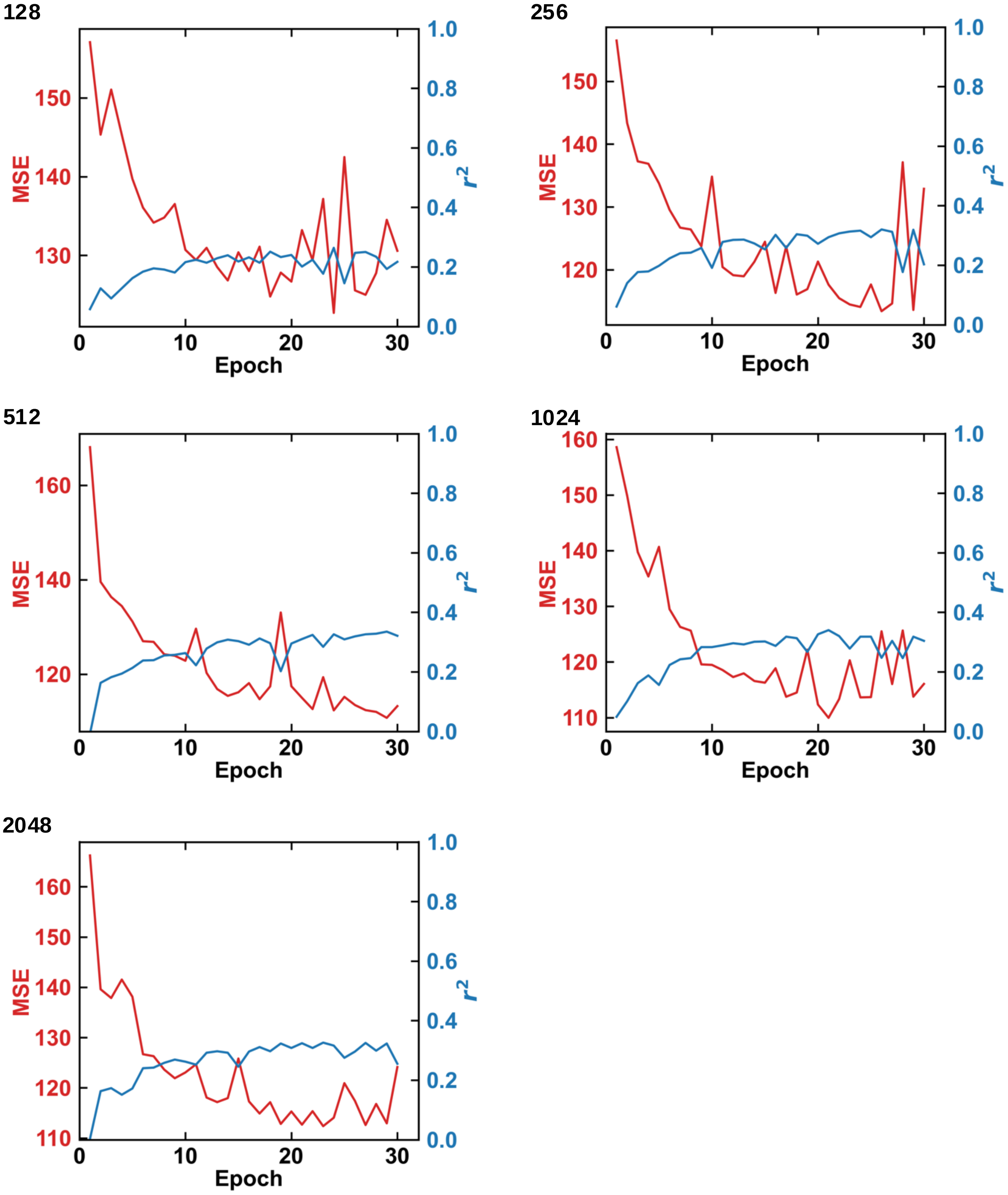
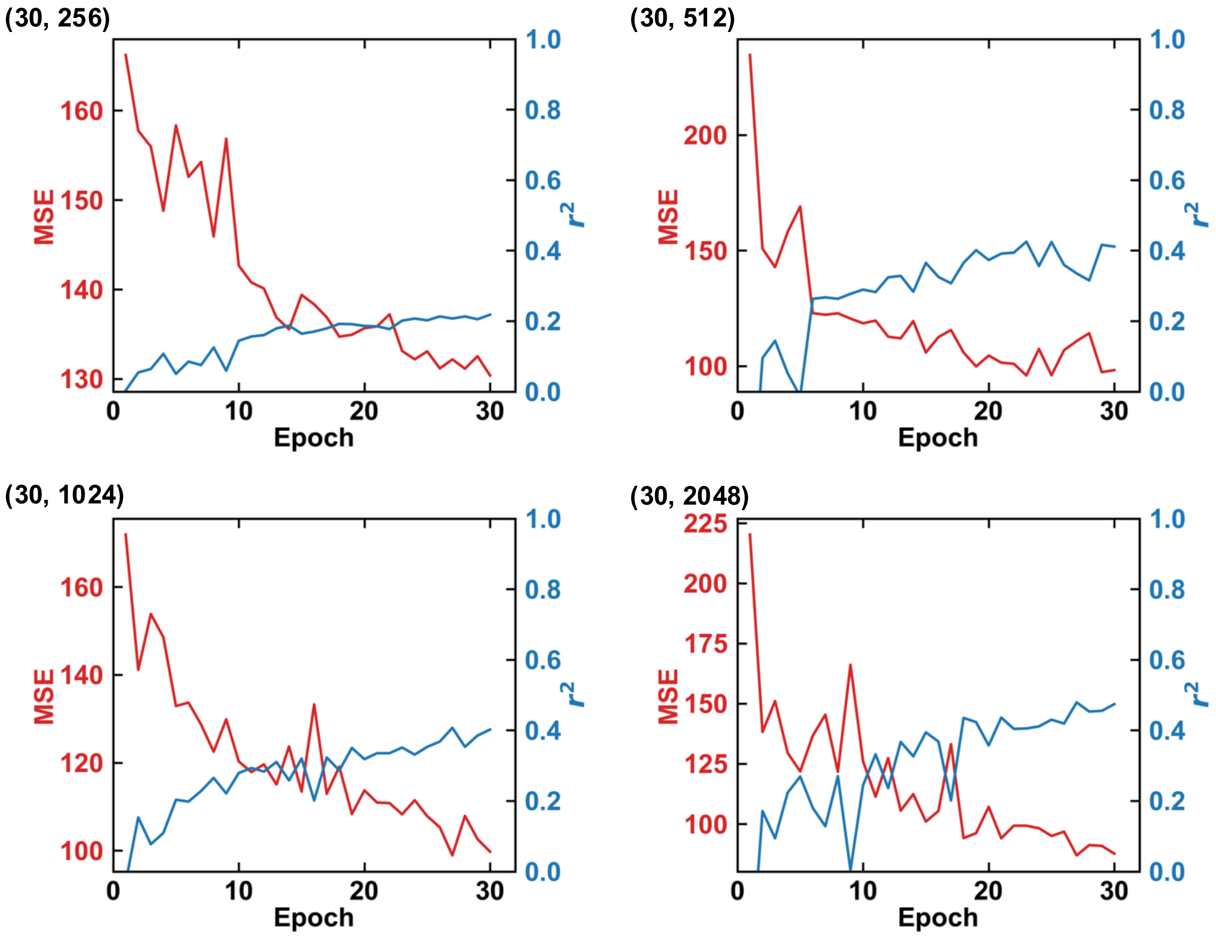


Figure S3: r^2 and MSE as a function of epochs during training of EdgeConv on atom-coordinate point clouds. The numbers on the top left corner of each panel indicate the number of points included from a unit cell.


(30, 256)
MSE
130
140
150
160
r^2
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
Epoch
(30, 512)
MSE
100
150
200
r^2
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
Epoch
(30, 1024)
MSE
100
120
140
160
r^2
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
Epoch
(30, 2048)
MSE
100
125
150
175
200
225
r^2
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
Epoch

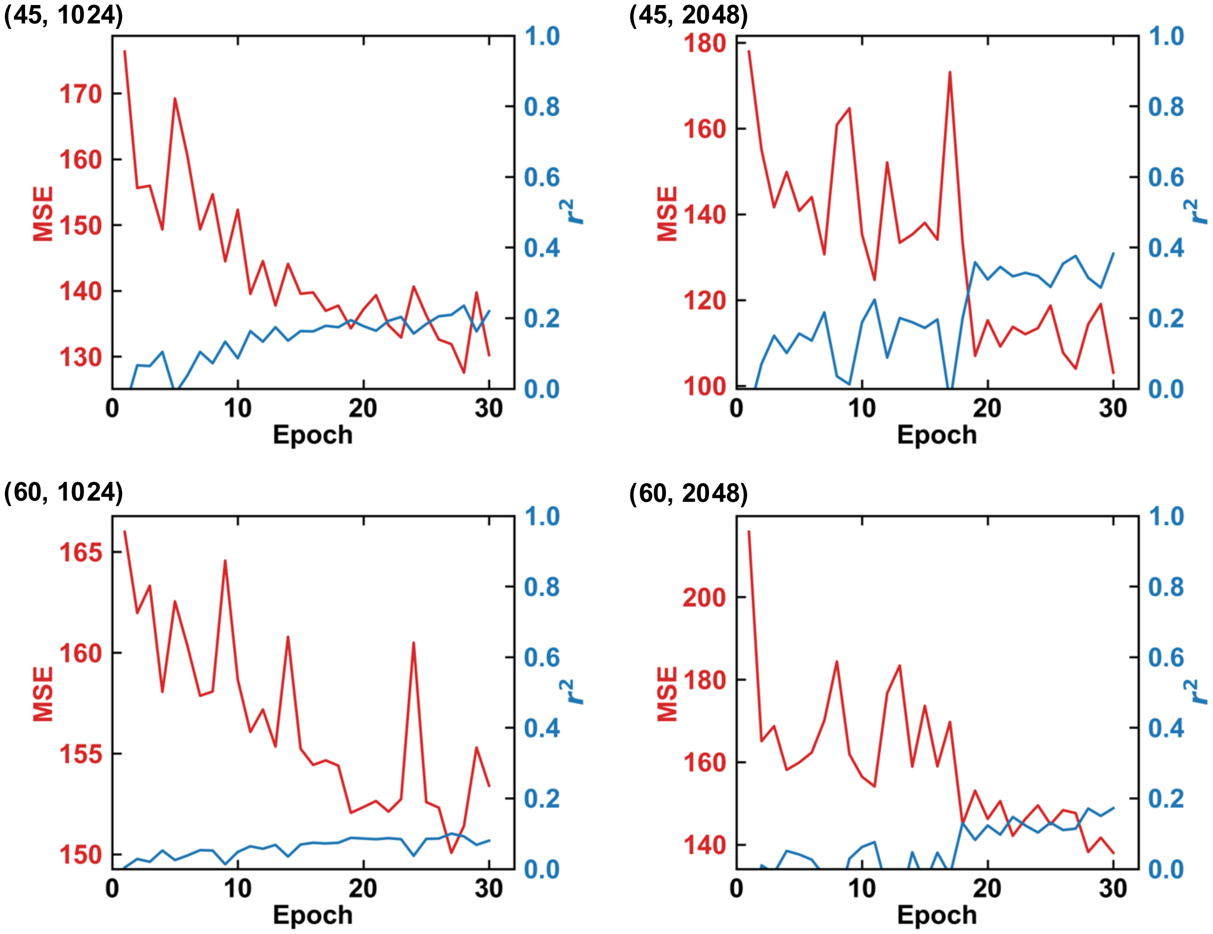
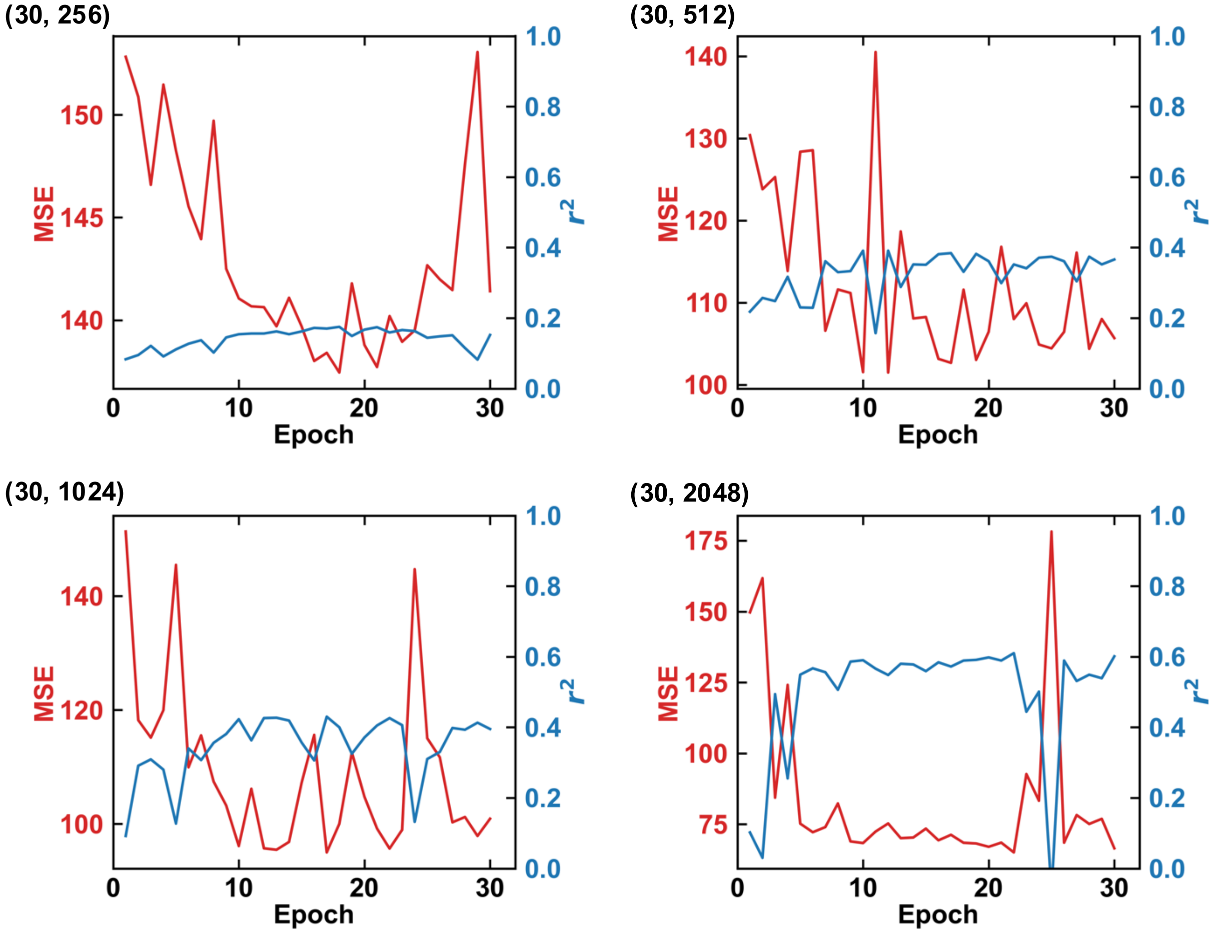


Figure S4: r^2 and MSE as a function of epochs during training of PointNet on atom-coordinate point clouds within a cube. The input representations are given in parentheses, (L in Å, number of points).


(30, 256)
(30, 512)
(30, 1024)
(30, 2048)
MSE
r^2
Epoch
0
10
20
30
150
145
140
140
130
120
110
100
140
120
100
175
150
125
100
75
0.0
0.2
0.4
0.6
0.8
1.0

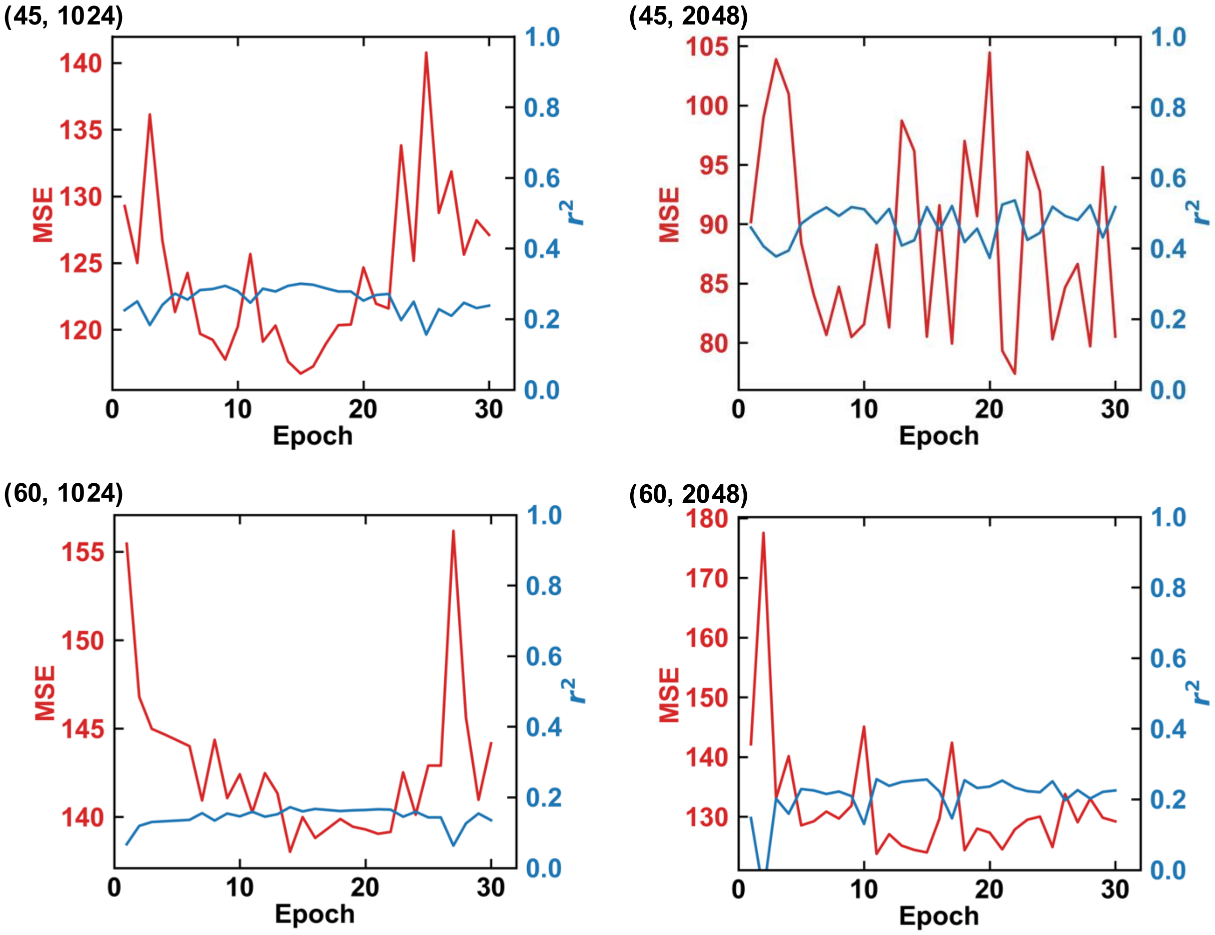


Figure S5: r^2 and MSE as a function of epochs during training of EdgeConv on atom-coordinate point clouds within a cube. The input representations are given in parentheses, (L in Å, number of points).

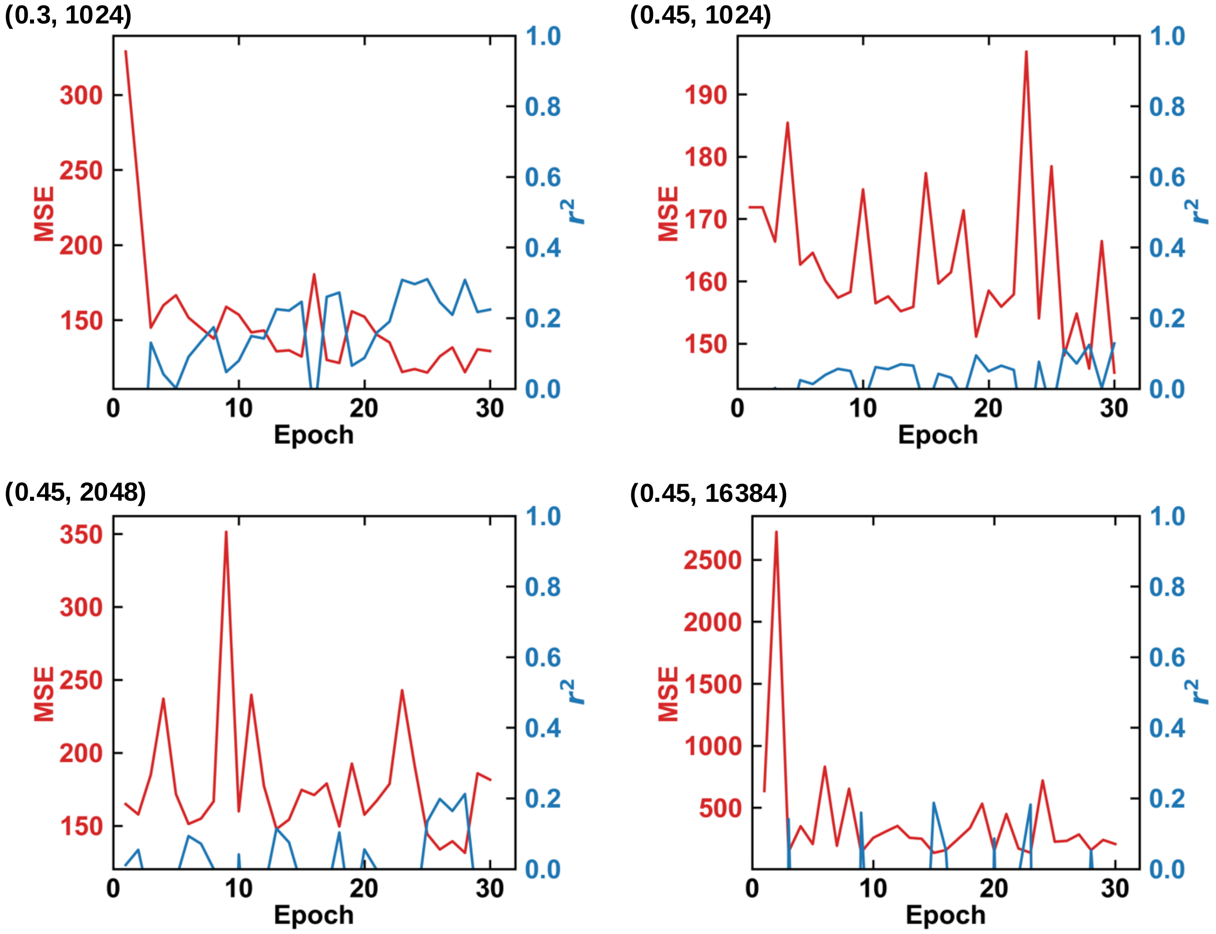
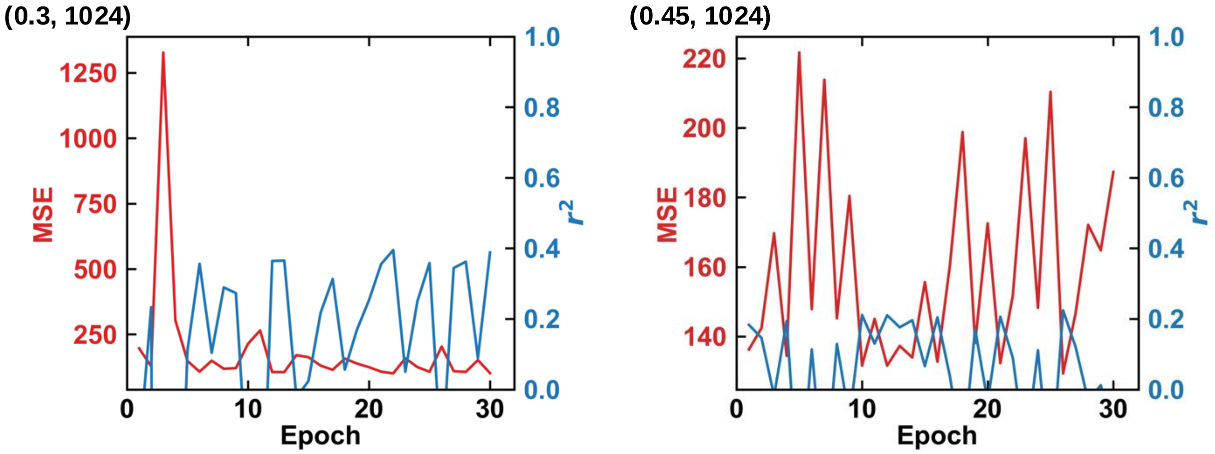


Figure S6: r^2 and MSE as a function of epochs duringthe training of PointNet on surface sampling point clouds within a cube. The input representations are given in parentheses, (Δd in Å, number of points).


(0.3, 1024)

(0.45, 1024)

Figure S7: r^2 and MSE as a function of epochs during training of EdgeConv on surface sampling point clouds within a cube. The input representations are given in parentheses, (Δd in Å, number of points).

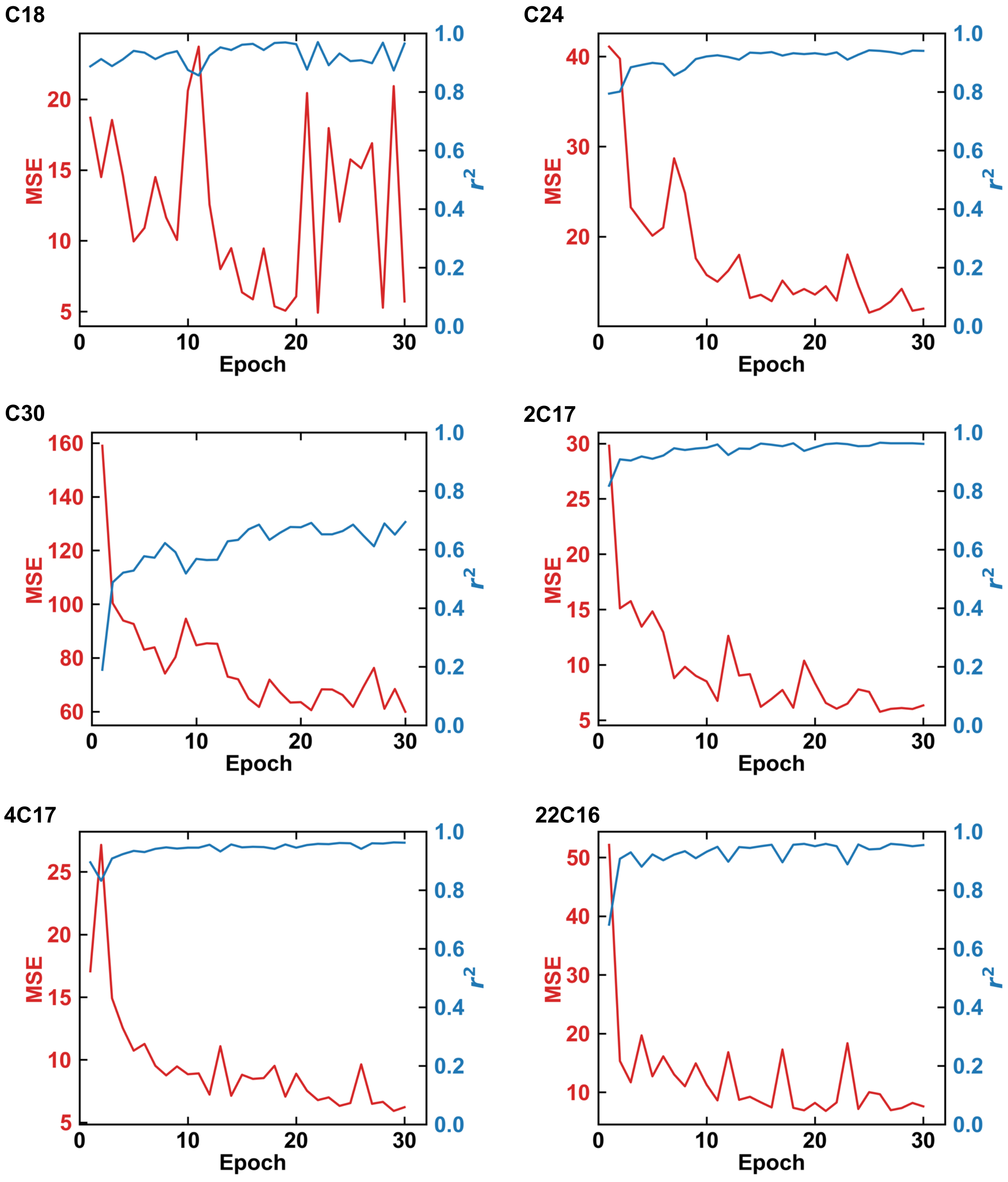


Figure S8: r^2 and MSE as a function of epochs during training of ResNet18 on the full training set.

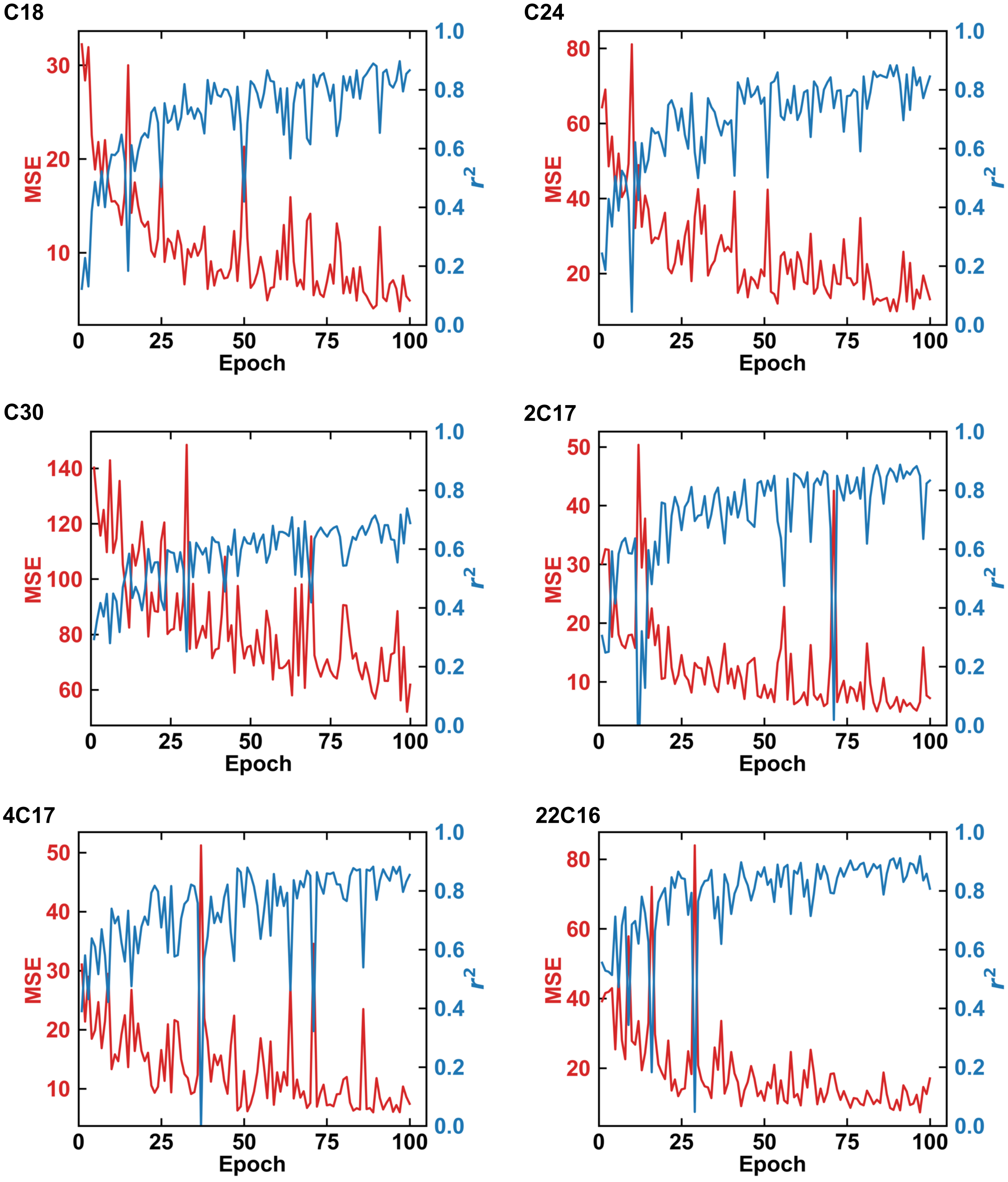


Figure S9: r^2 and MSE as a function of epochs during training of ResNet18 on the transfer-training subset.

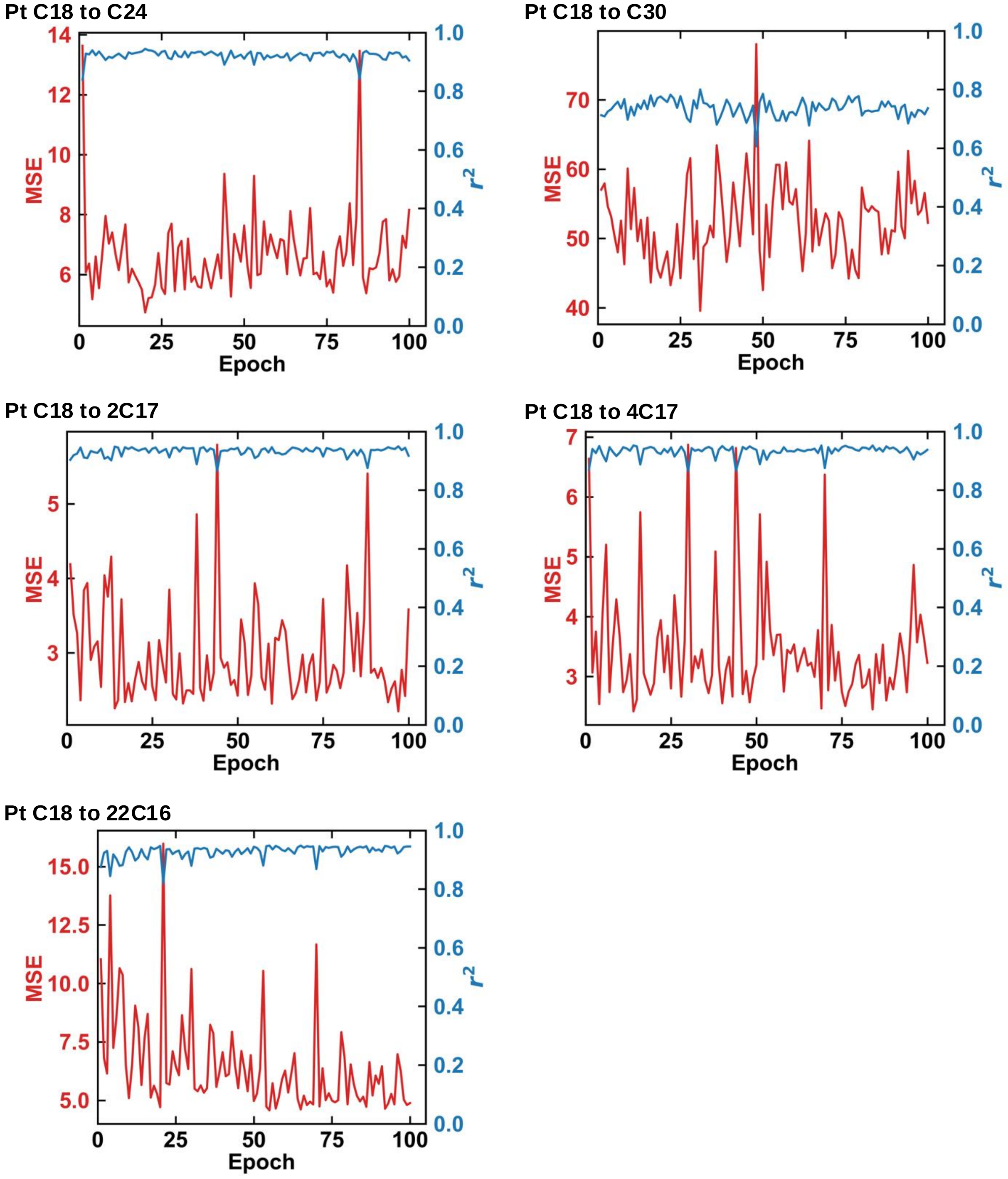
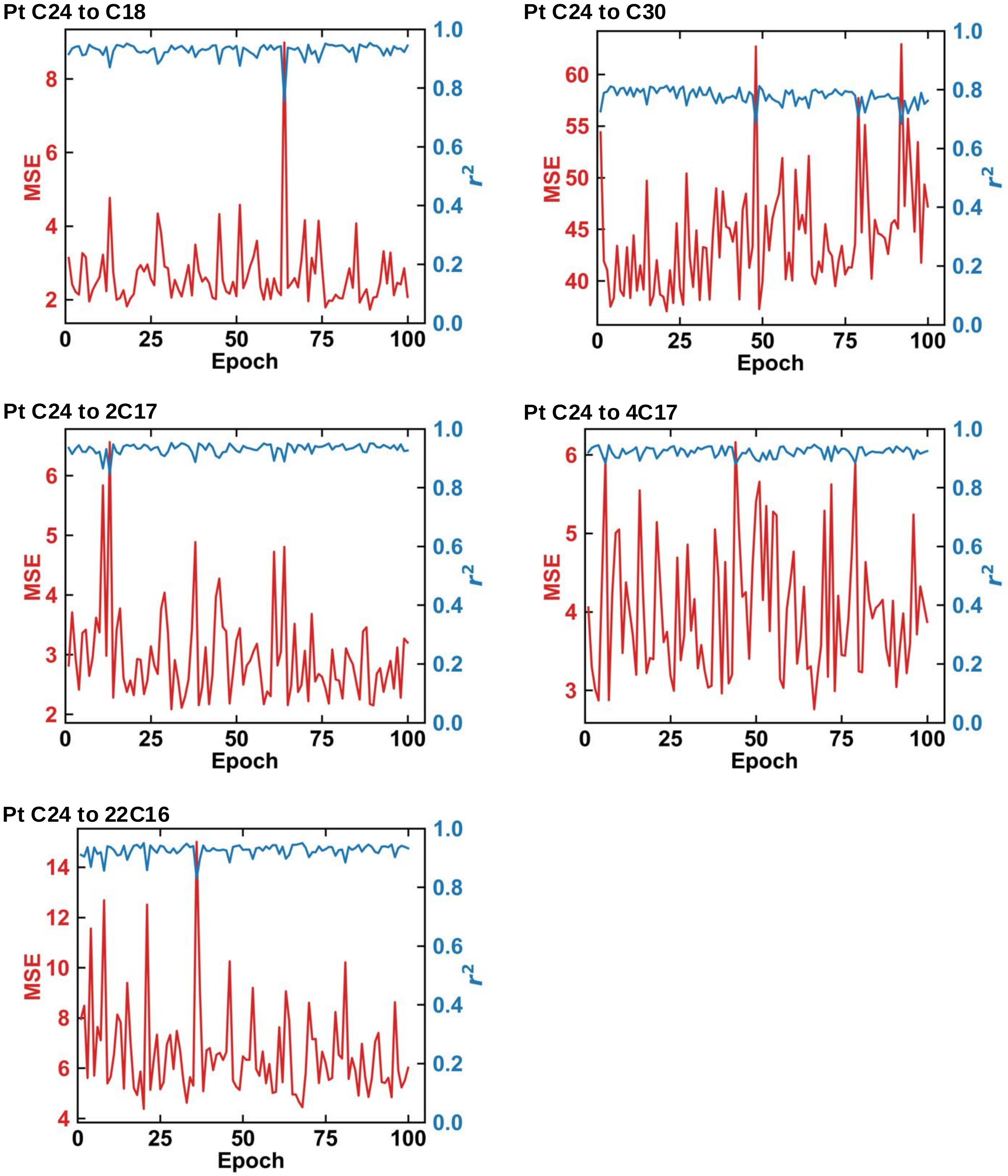


Figure S10: r^2 and MSE as a function of epochs during fine-tuning of ResNet18 pre-trained on C18.



Pt C24 to C18

Pt C24 to C30

Pt C24 to 2C17

Pt C24 to 4C17

Pt C24 to 22C16

Figure S11: r^2 and MSE as a function of epochs during fine-tuning of ResNet18 pre-trained on C24.

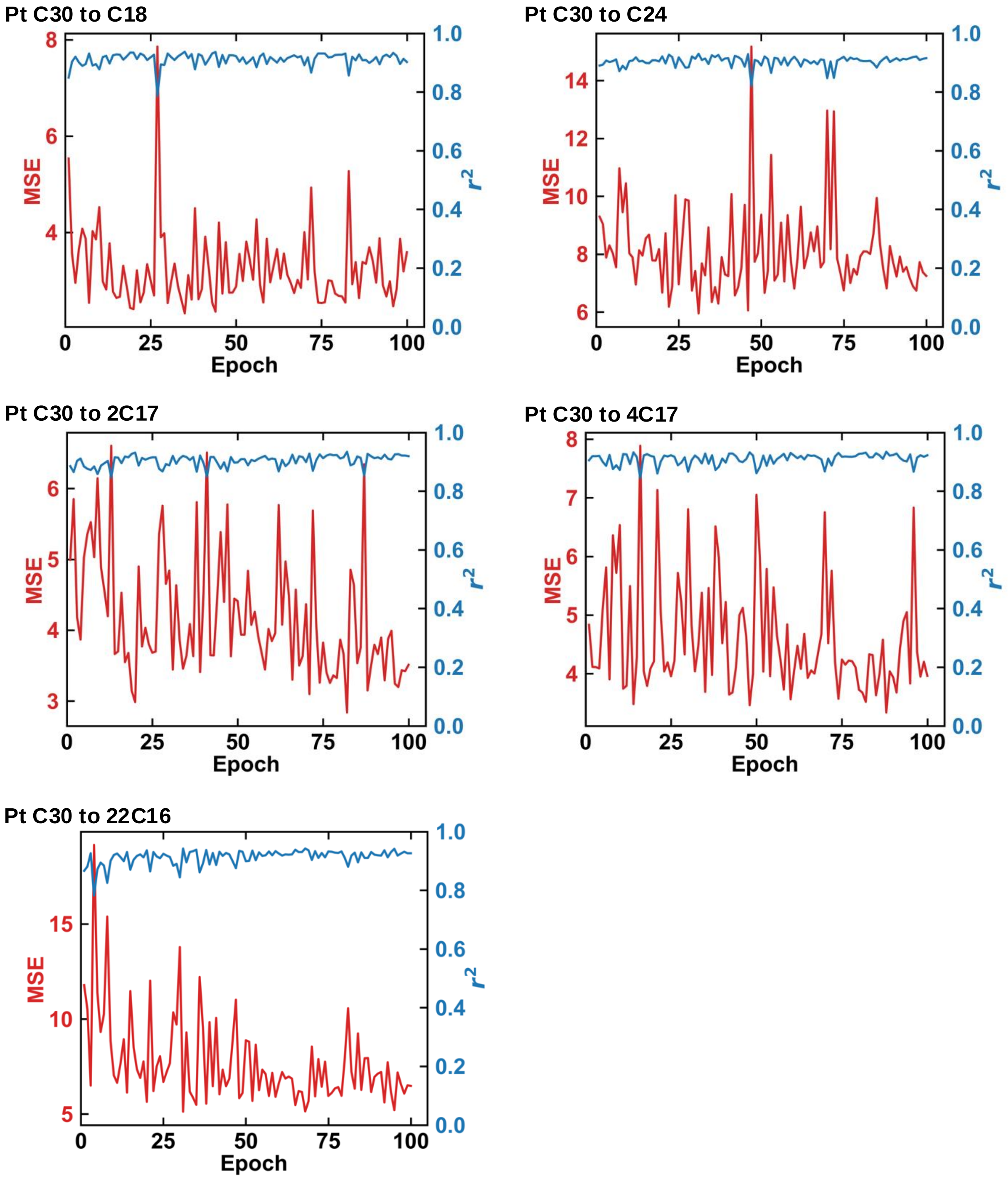
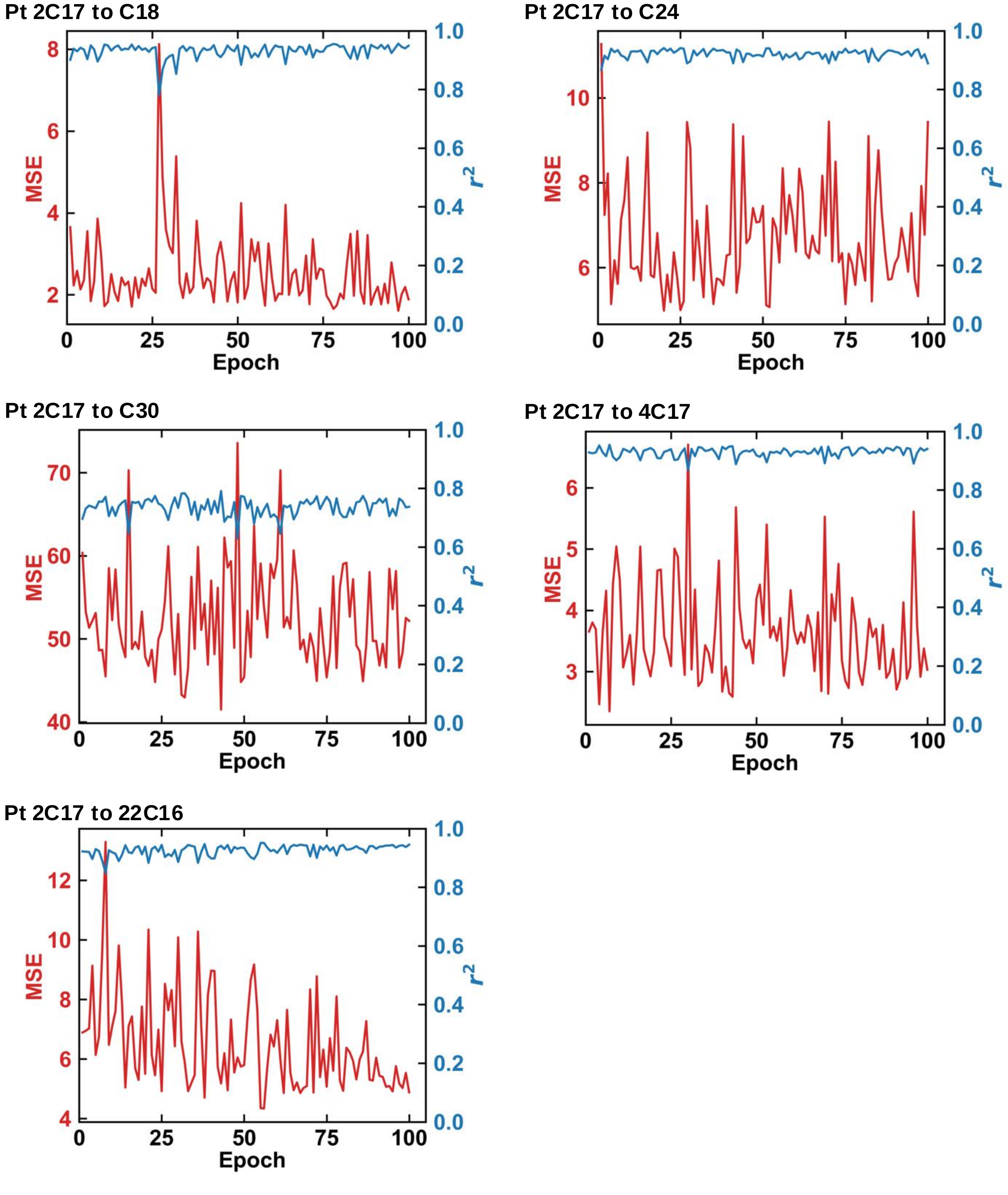


Figure S12: r^2 and MSE as a function of epochs during fine-tuning of ResNet18 pre-trained on C30.


Pt 2C17 to C18

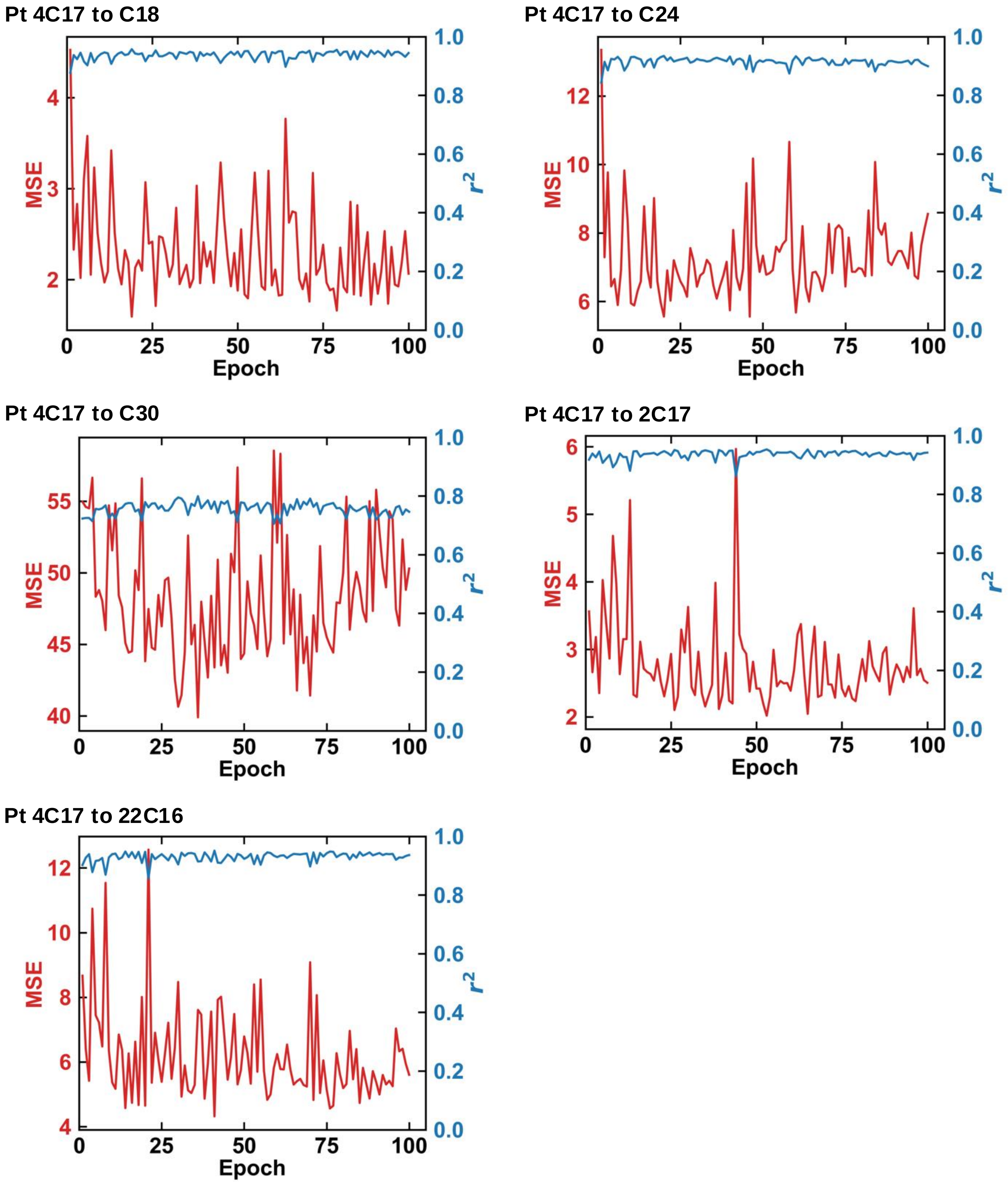
Pt 2C17 to C24

Pt 2C17 to C30

Pt 2C17 to 4C17

Pt 2C17 to 22C16

Figure S13: r^2 and MSE as a function of epochs during fine-tuning of ResNet18 pre-trained on 2C17.



Pt 4C17 to C18

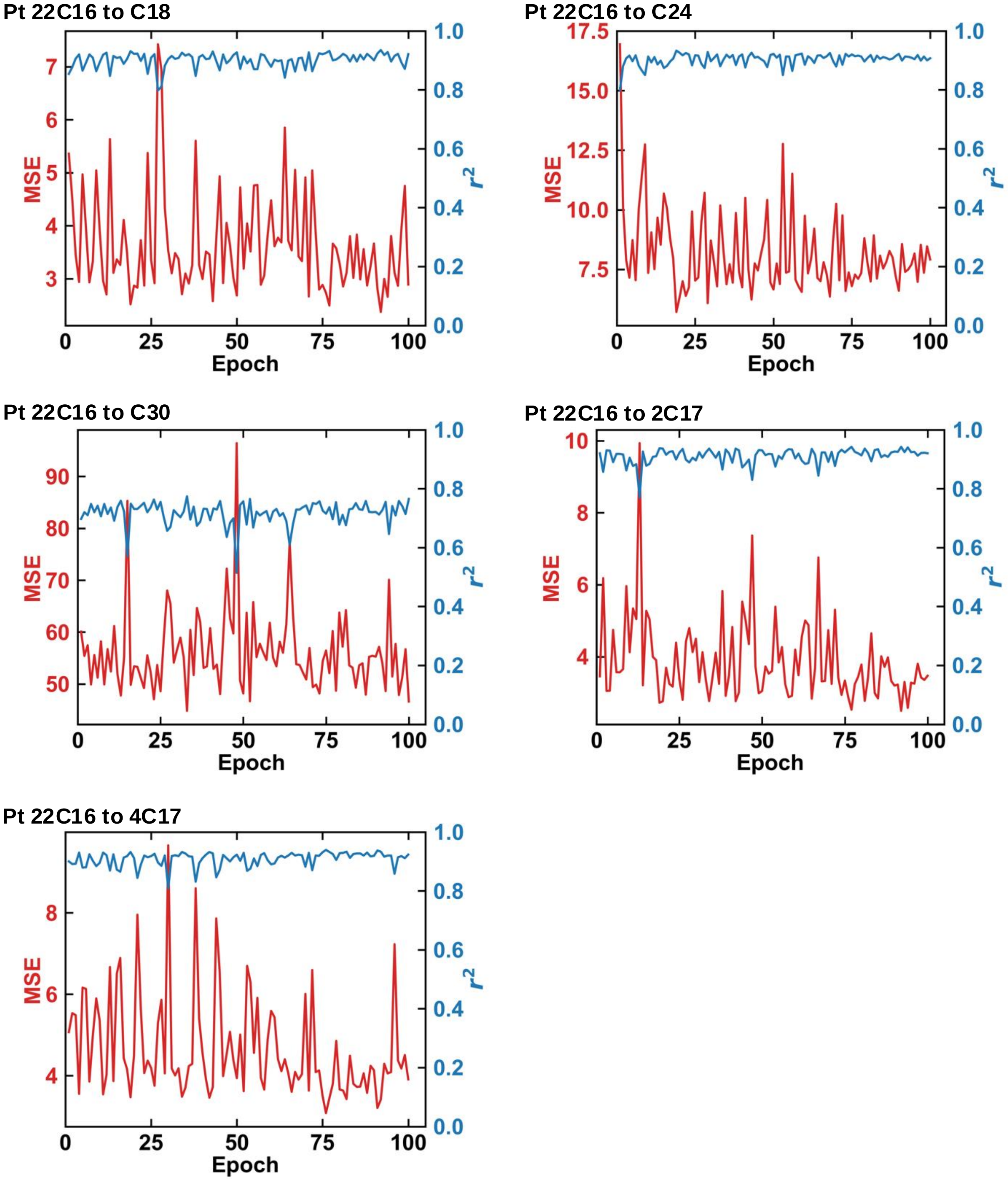
Pt 4C17 to C24

Pt 4C17 to C30

Pt 4C17 to 2C17

Pt 4C17 to 22C16

Figure S14: r^2 and MSE as a function of epochs during fine-tuning of ResNet18 pre-trained on 4C17.



Pt 22C16 to C18

Pt 22C16 to C24

Pt 22C16 to C30

Pt 22C16 to 2C17

Pt 22C16 to 4C17

Figure S15: r^2 and MSE as a function of epochs during fine-tuning of ResNet18 pre-trained on 22C16.

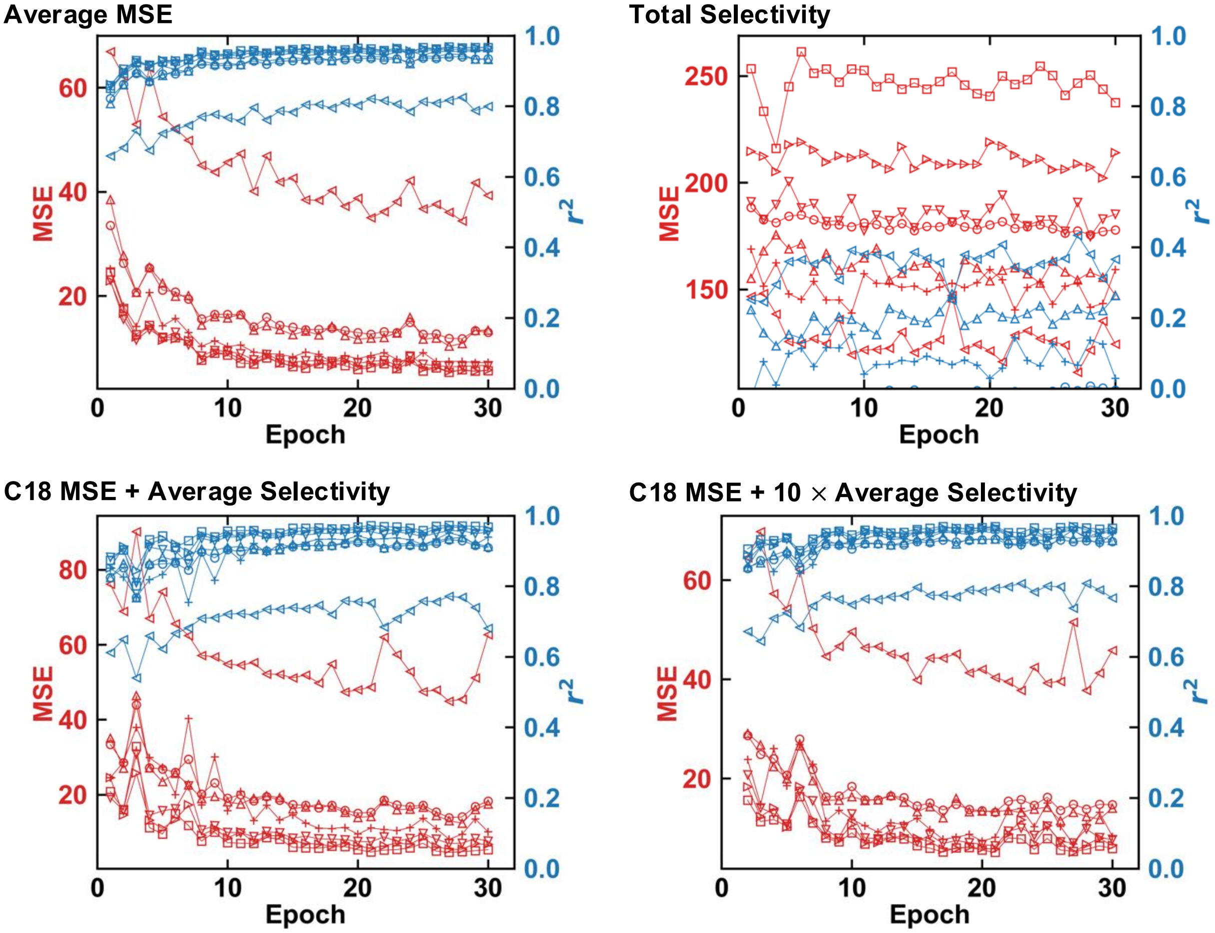


Figure S16: Individual r^2 and MSE of C18 (squares), C24 (up triangles), C30 (left triangles), 2C17 (right triangles), 4C17 (down triangles), 22C16 (pluses), composite r^2 (circles), and average MSE (circles) as a function of epochs during multi-task training of ResNet18 with different loss functions.

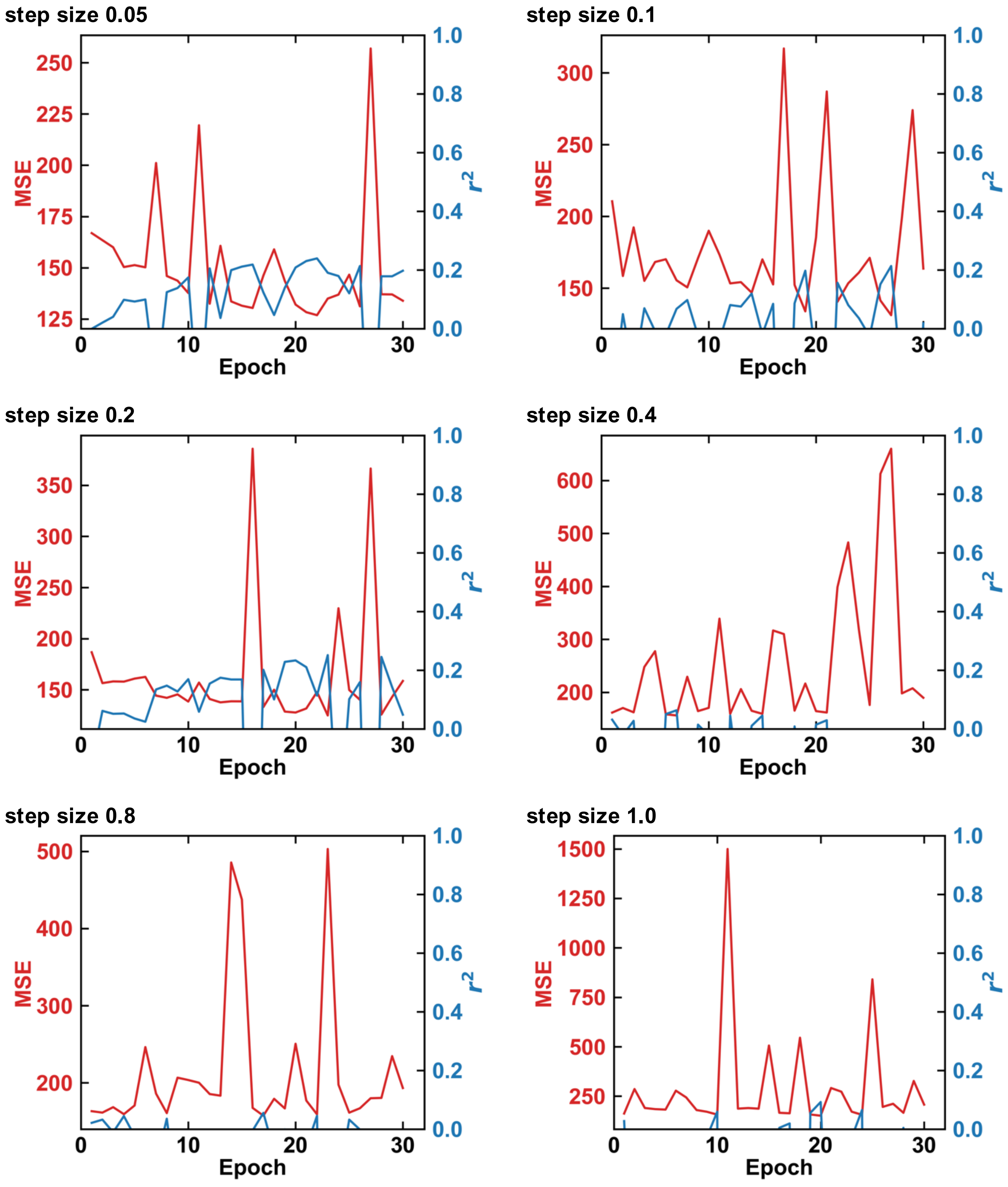


Figure S17: r^2 and MSE as a function of epochs during training of CGCNN with different step sizes, $\Delta\mu$.

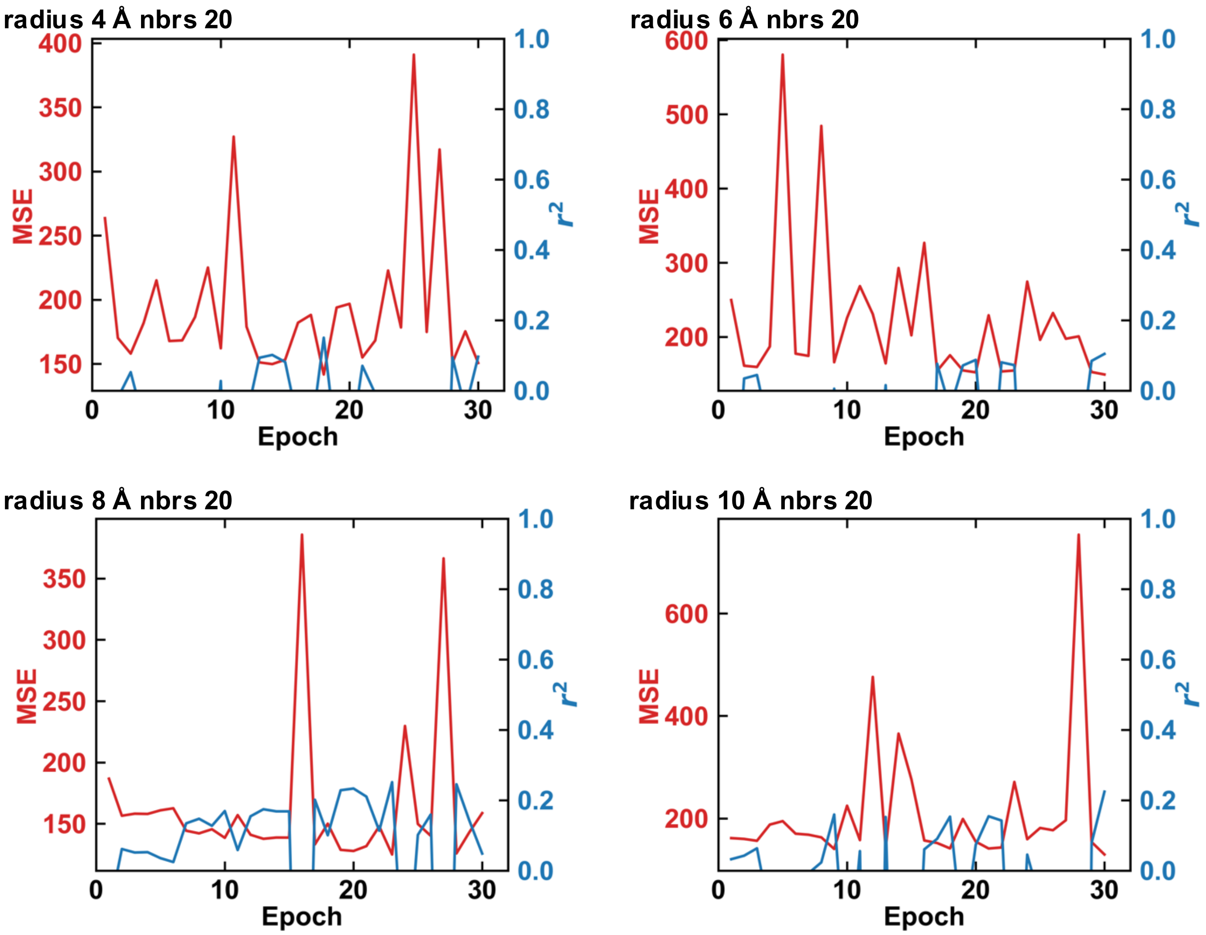


Figure S18: r^2 and MSE as a function of epochs during training of CGCNN with different cutoff radii, r_{cut}.

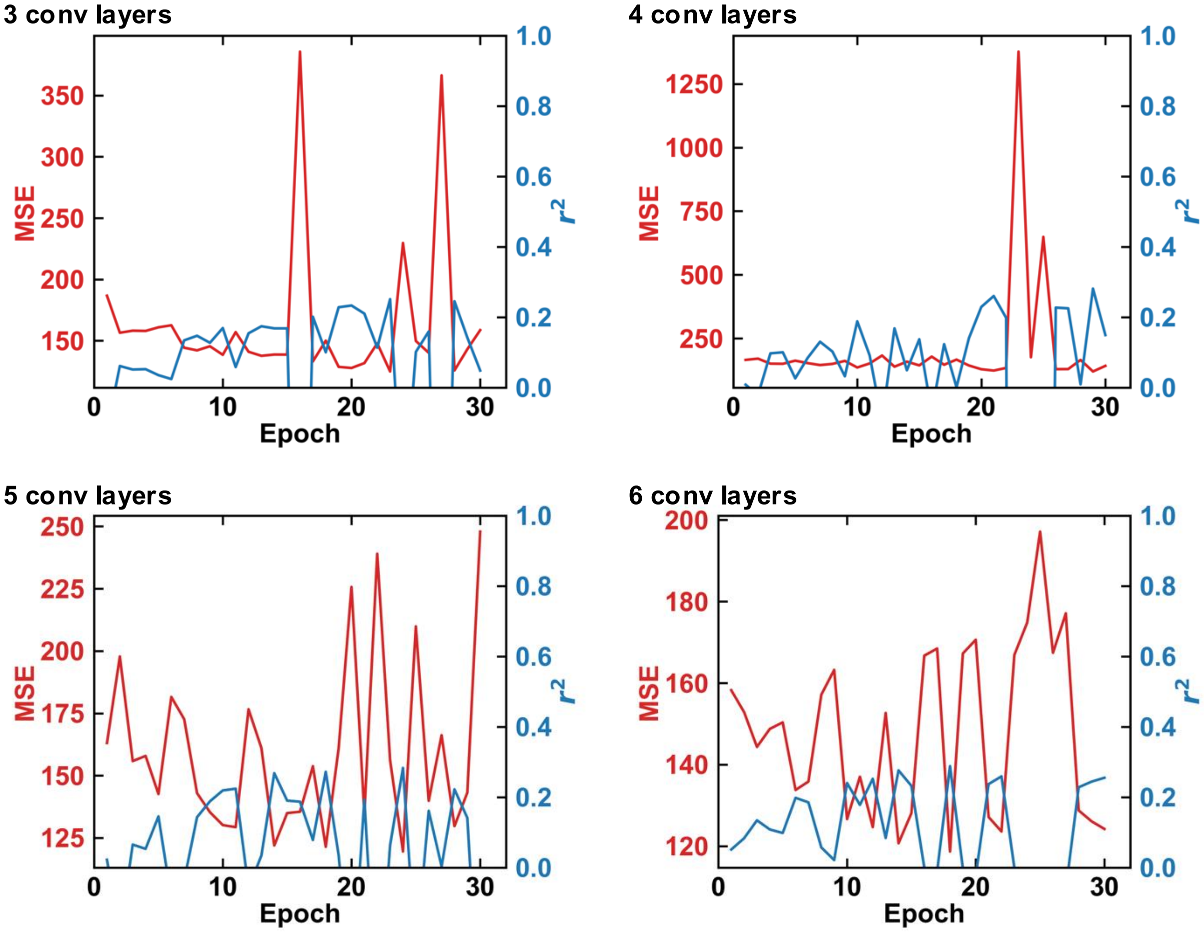
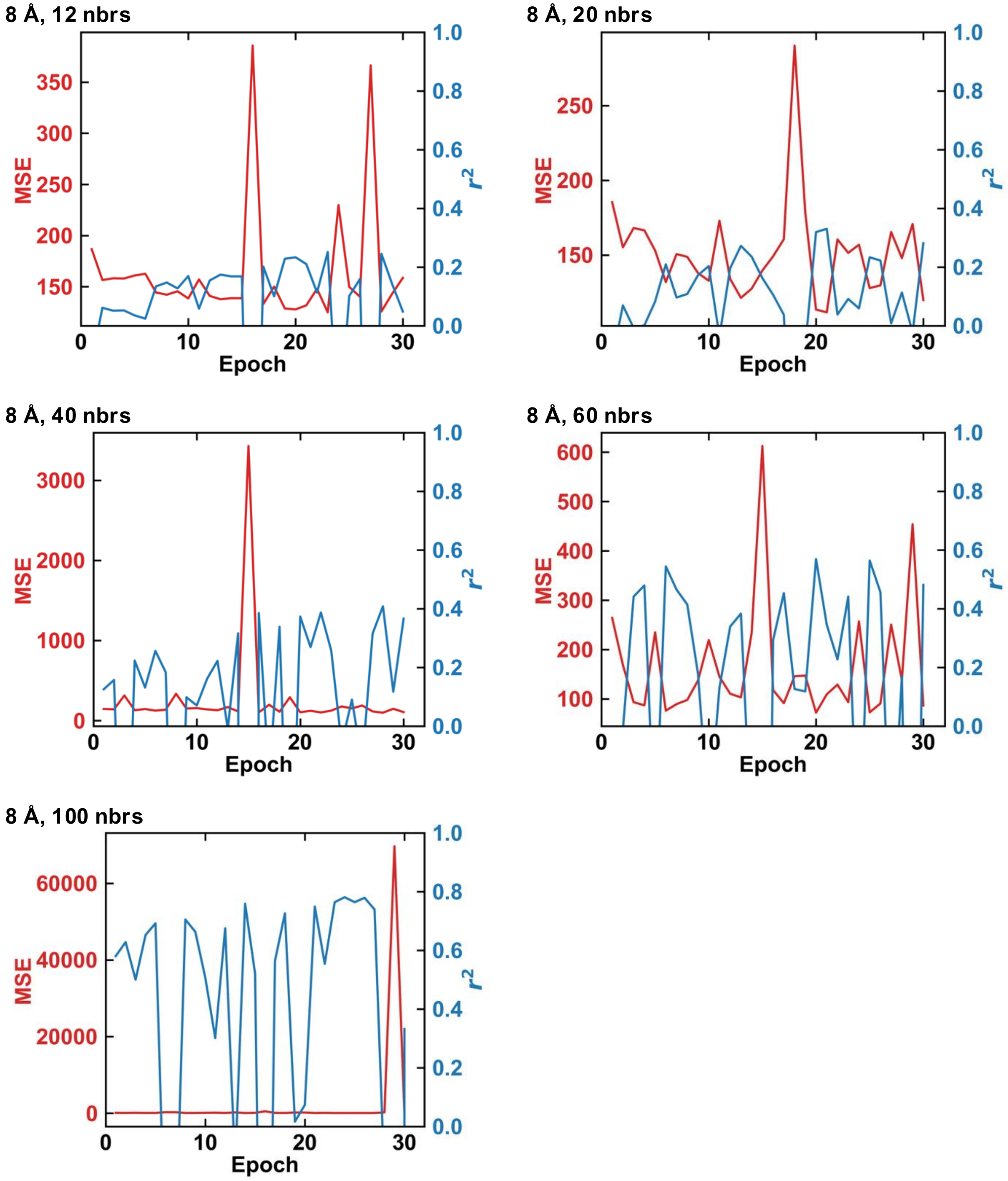


Figure S19: r^2 and MSE as a function of epochs during training of CGCNN with different numbers of convolutional layers, N_{conv}.



8 Å, 12 nbrs

8 Å, 20 nbrs

8 Å, 40 nbrs

8 Å, 60 nbrs

8 Å, 100 nbrs

Figure S20: r^2 and MSE as a function of epochs during training of CGCNN with different numbers of neighbor atoms, N_{neighbor}.

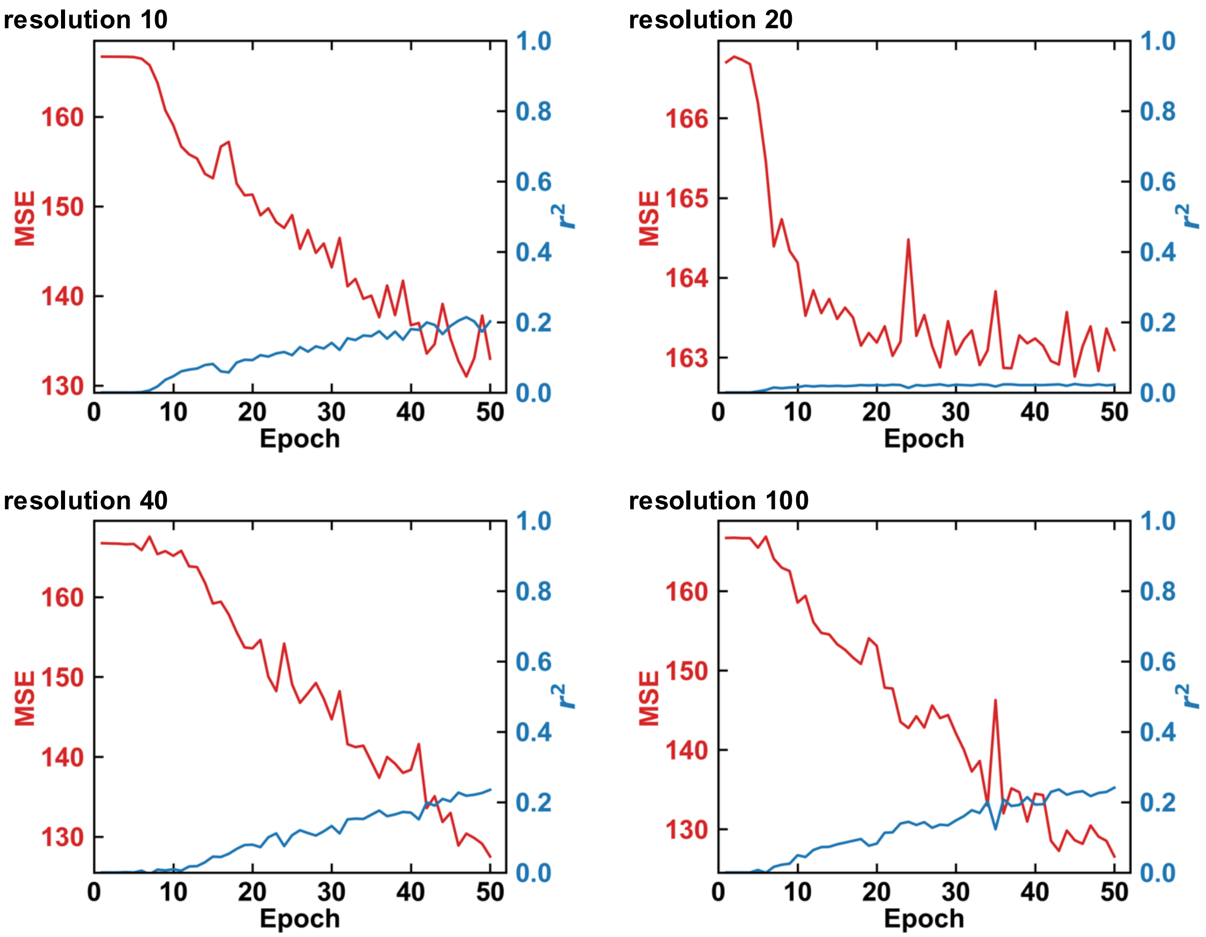


Figure S21: r^2 and MSE as a function of epochs during training of MEGNet with different numbers of basis functions, N_b.

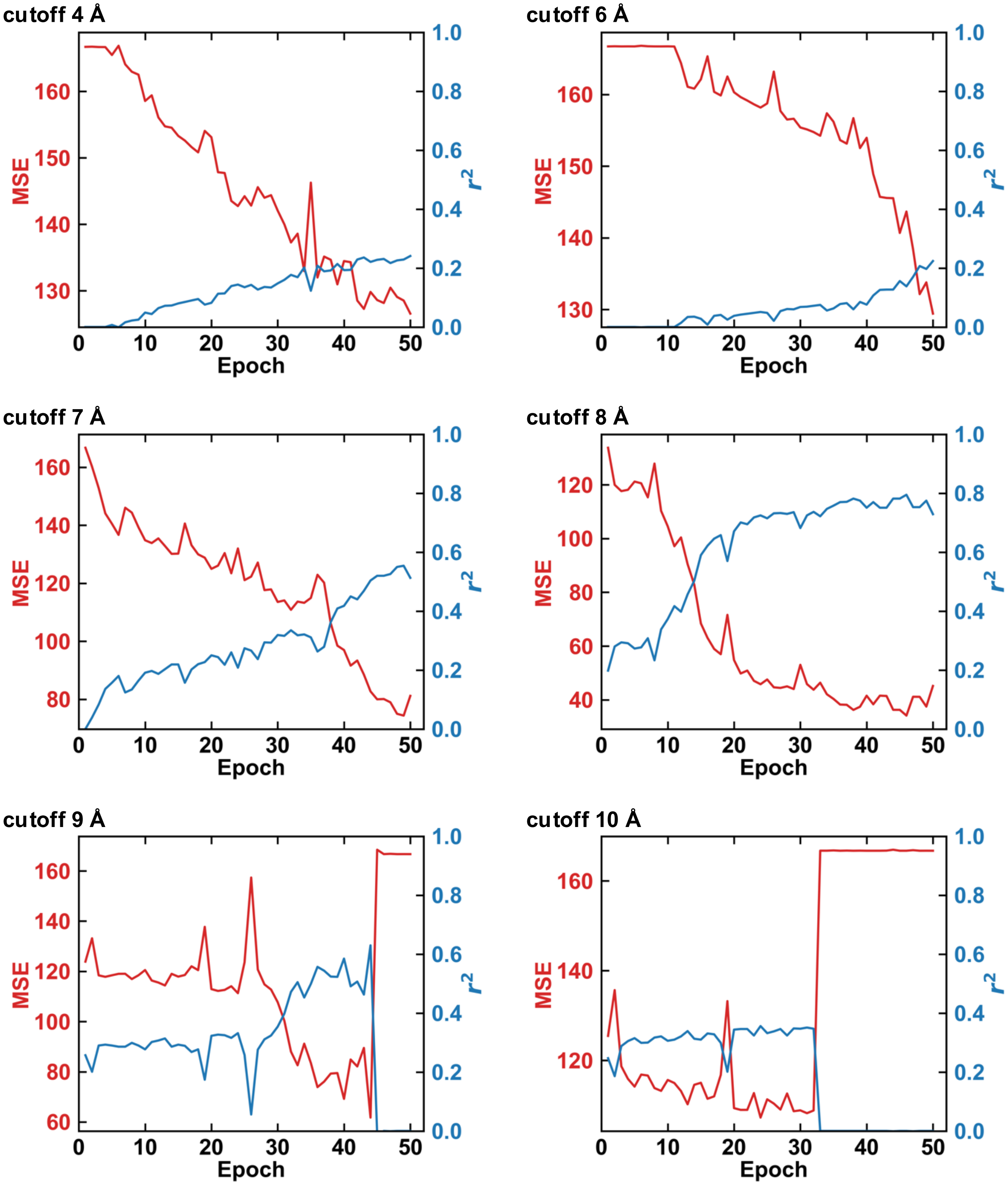
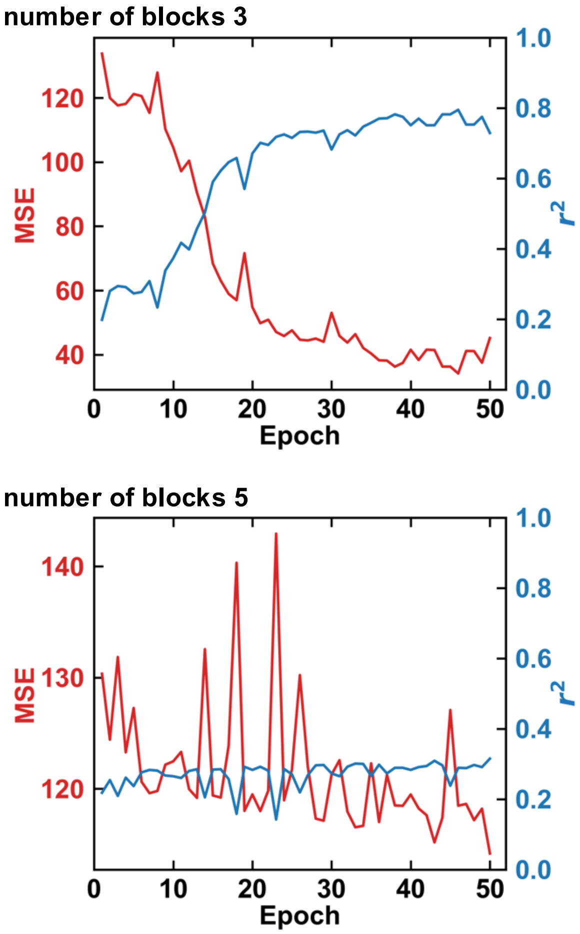


Figure S22: r^2 and MSE as a function of epochs during training of MEGNet with different cutoff radii, r_{cut}.



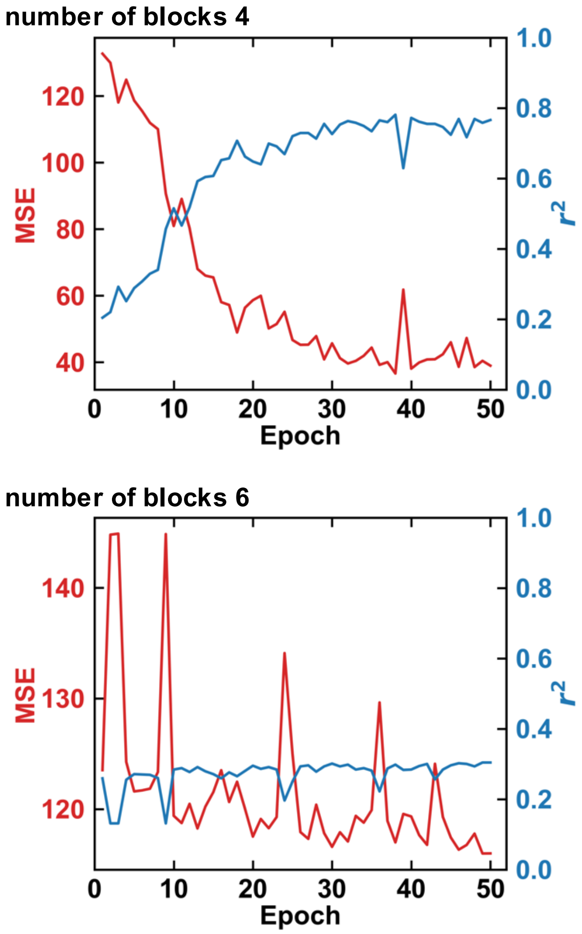
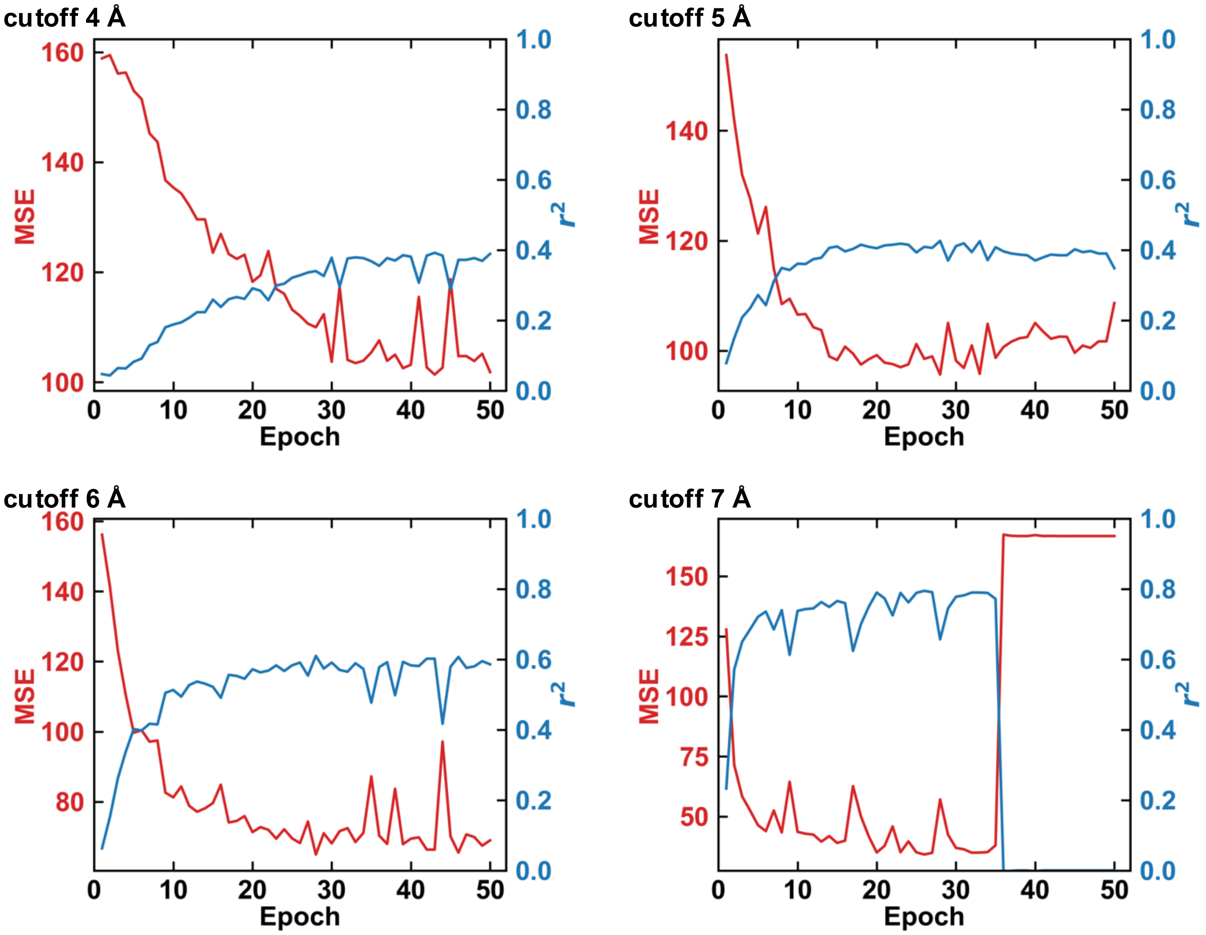


Figure S23: r^2 and MSE as a function of epochs during training of MEGNet with different numbers of MEGNet blocks, N_{block}.


cutoff 4 Å
MSE
160
140
120
100
r²
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
Epoch
cutoff 5 Å
MSE
140
120
100
r²
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
Epoch
cutoff 6 Å
MSE
160
140
120
100
80
r²
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
Epoch
cutoff 7 Å
MSE
150
125
100
75
50
r²
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
Epoch

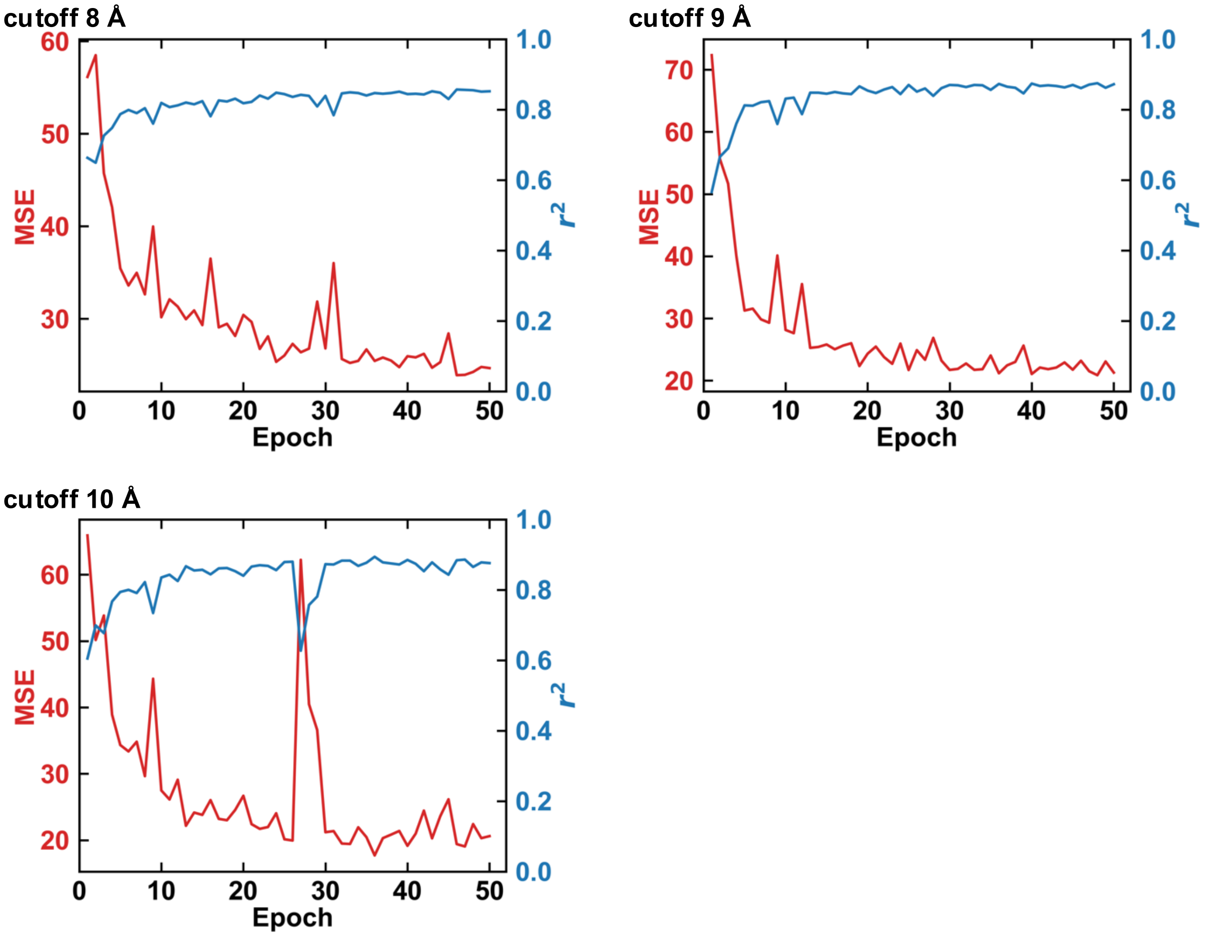


Figure S24: r^2 and MSE as a function of epochs during training of M3GNet with different cutoff radii, r_{cut}.

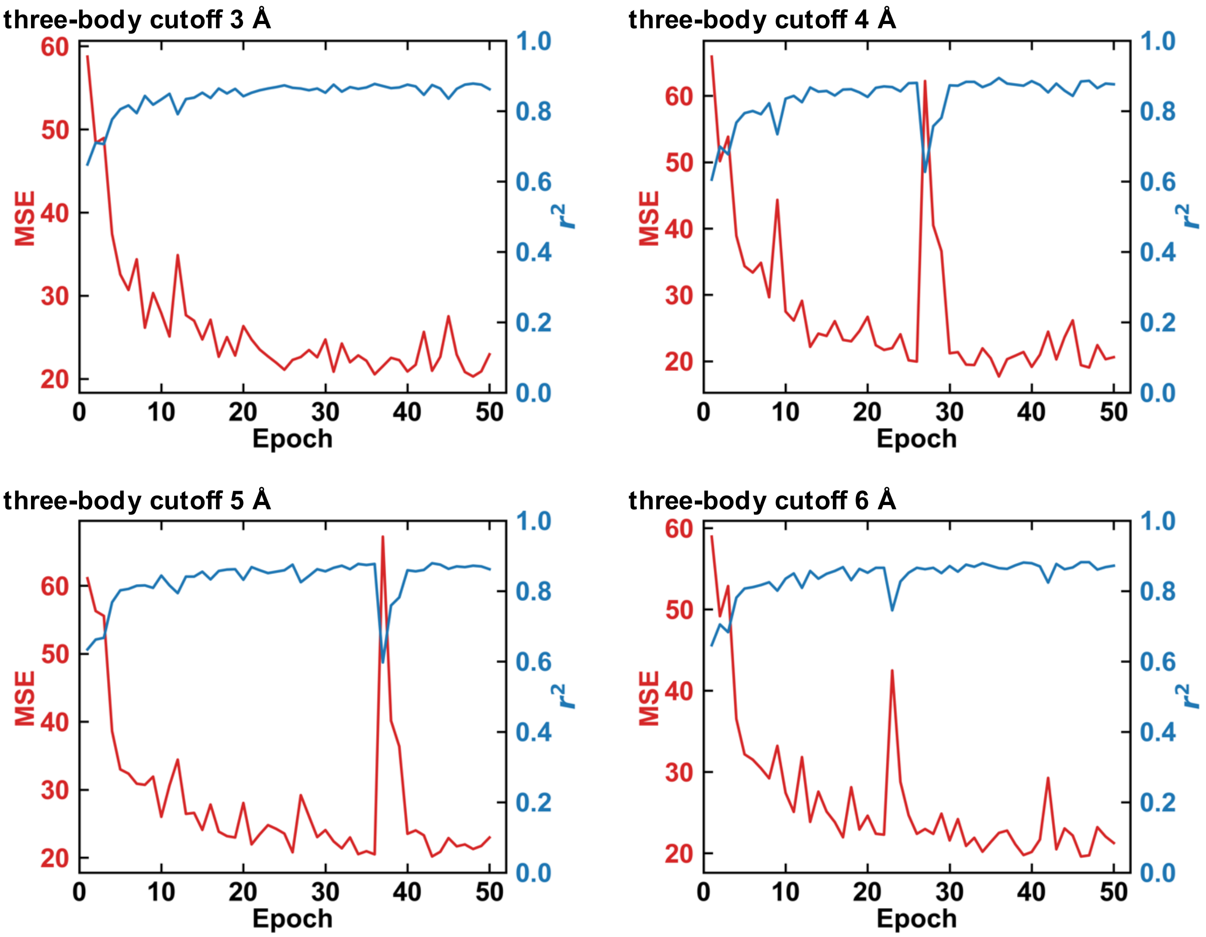


Figure S25: r^2 and MSE as a function of epochs during training of M3GNet with different three-body cutoffs, $r_{3-\mathrm{body}}$.

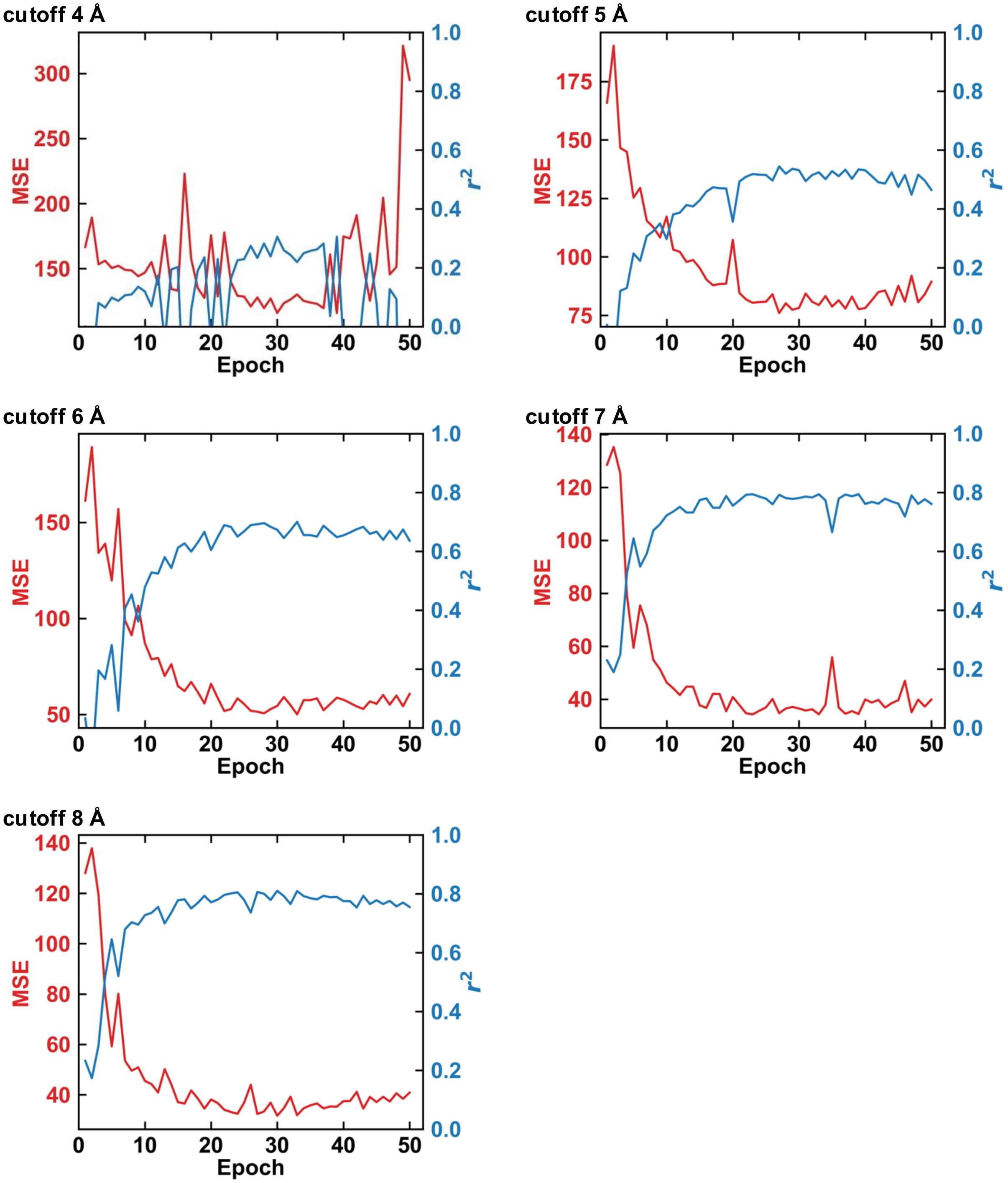


Figure S26: r^2 and MSE as a function of epochs during training of MACE with different cutoff radii, r_{cut}.

References

(1) Xie, T.; Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Physical Review Letters* **2018**, *120*, 145301.

(2) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. Proceedings of the 34th International Conference on Machine Learning. 2017; pp 1263–1272.

(3) Chen, C.; Ye, W.; Zuo, Y.; Zheng, C.; Ong, S. P. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials* **2019**, *31*, 3564–3572.

(4) Vinyals, O.; Bengio, S.; Kudlur, M. Order Matters: Sequence to Sequence for Sets. `http://arxiv.org/abs/1511.06391`.

(5) Chen, C.; Ong, S. P. A Universal Graph Deep Learning Interatomic Potential for the Periodic Table. *Nature Computational Science* **2022**, *2*, 718–728.

(6) Batatia, I.; Kovacs, D. P.; Simm, G. N. C.; Ortner, C.; Csanyi, G. MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields. Advances in Neural Information Processing Systems. 2022.