A Training and hyperparameters

Models were trained on NVIDIA A100 40 GB and NVIDIA RTX A6000 48 GB GPUs, except GemNet-OC-small which were trained on NVIDIA A100 80 GB and NVIDIA RTX A6000 48 GB. All models were trained on single GPUs, except for SchNet when trained on OC20-2M, which required 3 GPUs. Inference throughput was profiled on A100 40 GB GPUs, with reported values representing approximate numbers averaged across three evaluations. We provide detailed information about the hyperparameters we used for each model in Tables 5, 6, and 7.

Moreover, we summarize the KD weighting factors λ we used for each model configuration in Table 8.

Table 5: SchNet hyperparameters.

Hyperparameter	OC20	COLL
Hidden channels	1024	128
Filters	256	128
Interaction blocks	5	6
Gaussians	200	50
Cutoff	6.0	12.0
Batch size	192	32
Initial learning rate	10^{-4}	10^{-3}
Optimizer	AdamW	AdamW
Scheduler	LambdaLR	LinearWarmupExponentialDecay
Learning rate decay factor	0.1	0.01
Learning rate milestones	52083, 83333, 104166	-
Warmup steps	31250	3750
Warmup factor	0.1	-
Force Coefficient	100	100
Energy Coefficient	1	1
Number of epochs	30	500

Table 6: PaiNN hyperparameters. Slash-separated values indicate PaiNN versus PaiNN-small hyperparameters.

Hyperparameter	OC20	COLL
Hidden channels	512/256	256/128
Number of layers	6/4	6/4
Number of RBFs	128	128
Cutoff	12.0	12.0
Max. num. neighbors	50	50
Direct Forces	True	True
Batch size	32	32
Optimizer	AdamW	AdamW
AMSGrad	True	True
Initial learning rate	10^{-4}	10^{-3}
Scheduler	LambdaLR	LinearWarmupExponentialDecay
Warmup steps	None	3750
Learning rate decay factor	0.45	0.01
Learning rate milestones (steps)	160000, 320000, 480000, 640000	-
Force coefficient	100	100
Energy coefficient	1	1
EMA decay	0.999	0.999
Gradient clip norm threshold	10	10
Epochs	16	375

Table 7: GemNet-OC hyperparameters. Slash-separated values indicate GemNet-OC versus GemNet-OC-small hyperparameters.

Hyperparameter	OC20	COLL
No. spherical basis	7	7
No. radial basis	128	128
No. blocks	4/3	4
Atom embedding size	256/128	128
Edge embedding size	512/256	256
Triplet edge embedding input size	64	64
Triplet edge embedding output size	64	64
Quadruplet edge embedding input size	32	32
Quadruplet edge embedding output size	32	32
Atom interaction embedding input size	64	64
Atom interaction embedding output size	64	64
Radial basis embedding size	16	16
Circular basis embedding size	16	16
Spherical basis embedding size	32	32
No. residual blocks before skip connection	2	2
No. residual blocks after skip connection	2	2
No. residual blocks after concatenation	1	1
No. residual blocks in atom embedding blocks	3	3
No. atom embedding output layers	3	3
Cutoff	12.0	12.0
Quadruplet cutoff	12.0	12.0
Atom edge interaction cutoff	12.0	12.0
Atom interaction cutoff	12.0	12.0
Max interaction neighbors	30	30
Max quadruplet interaction neighbors	8	8
Max atom edge interaction neighbors	20	20
Max atom interaction neighbors	1000	1000
Radial basis function	Gaussian	Gaussian
Circular basis function	Spherical harmonics	Spherical Harmonics
Spherical basis function	Legendre Outer	Legendre Outer
Quadruplet interaction	True	True
Atom edge interaction	True	True
Edge atom interaction	True	True
Atom interaction	True	True
Direct forces	True	True
Activation	Silu	Silu
Optimizer	AdamW	AdamW
Scheduler	ReduceLROnPlateau	LinearWarmup ExponentialDecay
Force coefficient	100	100
Energy coefficient	1	1
EMA decay	0.999	0.999
Gradient clip norm threshold	10	10
Initial learning rate	5×10^{-4}	10^{-3}
Epochs	80/9	165

Table 8: Choice of the weighting factor λ of the KD loss for the different teacher-student configurations and KD strategies.

Teacher	Student	KD	OC20	COLL
GemNet-OC	PaiNN	*Vanilla (1)*	1.0	0.2
GemNet-OC	PaiNN	*Vanilla (2)*	500	100
GemNet-OC	PaiNN	*n2n*	10000	1000
GemNet-OC	PaiNN	*e2n*	1000	10
GemNet-OC	PaiNN	*v2v*	50000	100
GemNet-OC	GemNet-OC-small	*Vanilla (1)*	0.2	-
GemNet-OC	GemNet-OC-small	*Vanilla (2)*	10.0	-
GemNet-OC	GemNet-OC-small	*n2n*	1000.0	-
GemNet-OC	GemNet-OC-small	*e2e*	100000	-
PaiNN	PaiNN-small	*Vanilla (1)*	1	1
PaiNN	PaiNN-small	*Vanilla (2)*	200	100
PaiNN	PaiNN-small	*n2n*	100	100
PaiNN	PaiNN-small	*v2v*	1000	10000
PaiNN	SchNet	*Vanilla (1)*	0.1	1
PaiNN	SchNet	*Vanilla (2)*	0.1	100
PaiNN	SchNet	*n2n*	1000	100

B Full validation results on OC20

Tables 9-12 present the extended results on OC20 across the 4 separate S2EF validation sets.

Table 9: Evaluation results on the OC20 S2EF **in-distribution** validation set.

		OC20 S2EF Validation (**in-distribution**)			
	Model	Energy MAE meV ↓	Force MAE meV/Å ↓	Force cos ↑	EFwT % ↑
same	*S*: PaiNN-small	409	41.6	0.357	0.16
	T: PaiNN	358	38.5	0.390	0.25
	Vanilla (1)	426(-33.7%)	42.6(-32.3%)	0.35(-21.9%)	0.12(-44.4%)
	Vanilla (2)	396(25.7%)	45.7(-132.3%)	0.316(-126.9%)	0.11(-55.6%)
	n2n	**393 (31.2%)**	41.7(-2.5%)	**0.359 (4.7%)**	0.15(-7.8%)
	v2v	406(5.6%)	42.1(-16.9%)	0.357(-2.6%)	0.13(-32.6%)
	S: GemNet-OC-small	292	27.7	0.534	0.90
	T: GemNet-OC	226	22.5	0.610	1.09
	Vanilla (1)	292(0.1%)	27.7(0.0%)	0.535(1.1%)	0.92(1.8%)
	Vanilla (2)	283(13.5%)	27.7(-0.4%)	0.535(0.1%)	1.01(18.8%)
	n2n	**252 (61.1%)**	27.5(3.8%)	0.536(1.6%)	**1.09 (19.2%)**
	e2e	285(10.1%)	**26.4 (25.3%)**	**0.551 (22.5%)**	1.01(10.9%)
similar	*S*: SchNet	1237	62.2	0.214	0
	T: PaiNN	358	38.5	0.390	0.25
	Vanilla (1)	**1139 (10.6%)**	**52.9 (13.1%)**	**0.2422 (16%)**	0(0%)
	Vanilla (2)	1140(10.5%)	59.2(12.7%)	0.241(15.1%)	0(0%)
	n2n	1170(7%)	60(9.3%)	0.235(11.9%)	0(0%)
different	*S*: PaiNN	358	38.5	0.390	0.25
	T: GemNet-OC	226	22.5	0.61	1.89
	Vanilla (1)	356(1.7%)	38.3(1.1%)	0.392(1.2%)	0.258(0.5%)
	Vanilla (2)	357(0.7%)	43.5(-31.4%)	0.334(-25.4%)	0.210(-2.4%)
	n2n	**271 (66.0%)**	37.3(7.5%)	0.408(8.2%)	**0.477 (13.9%)**
	e2n	330(21.8%)	**36.3(14.0%)**	**0.419 (13.4%)**	0.371(7.4%)
	v2v	369(-8.2%)	37.2(8.0%)	0.409(8.9%)	0.217(-2.0%)

Table 10: Evaluation results on the OC20 S2EF **out-of-distribution (adsorbates)** validation set.

		OC20 S2EF Validation (**out-of-distribution (adsorbates)**)			
	Model	Energy MAE meV ↓	Force MAE meV/Å ↓	Force cos ↑	EFwT % ↑
same	*S*: PaiNN-small	469	47.9	0.334	0.03
	T: PaiNN	437	44.5	0.369	0.043
	Vanilla (1)	519(-156.2%)	47.4(14.7%)	0.334(-1.2%)	0.003(0%)
	Vanilla (2)	477(-23.6%)	54.5(-106.2%)	0.297(-109.9%)	0.002(-76.9%)
	n2n	443(80.9%)	**47.2 (19.8%)**	**0.337 (9.3%)**	0.003(-10%)
	v2v	**441 (87.5%)**	47.9(-1%)	0.337(8.68%)	**0.04 (81.6%)**
	S: GemNet-OC-small	325	31.0	0.521	0.190
	T: GemNet-OC	258	25.2	0.600	0.45
	Vanilla (1)	312(19.5%)	31.0(-0.7%)	0.522(2.1%)	0.19(-0.9%)
	Vanilla (2)	309(24.2%)	30.9(1.4%)	0.523(2.6%)	0.20(2.7%)
	n2n	**282 (63.7%)**	30.9(1.7%)	0.523(5.8%)	0.22(11.5%)
	e2e	315(14.8%)	**29.3 (28.8%)**	**0.542 (26.3%)**	**0.23 (16.5%)**
similar	*S*: SchNet	1344	58	0.196	0
	T: PaiNN	437	44.5	0.369	0.043
	Vanilla (1)	1247(10.7%)	64.5(-49.7%)	**0.221 (14.7%)**	0(0%)
	Vanilla (2)	**1245 (10.9%)**	64.5(-48.2)	0.22(14%)	0(0%)
	n2n	1286(6.3%)	65(-51.9%)	0.213(9.9%)	0(0%)
different	*S*: PaiNN	437	44.5	0.369	0.043
	T: GemNet-OC	258	25.2	0.6	0.45
	Vanilla (1)	424(7.2%)	44.5(-0.2%)	0.370(0.5%)	0.052(2.3%)
	Vanilla (2)	408(15.9%)	49.2(-24.3%)	0.315(-23.1%)	0.036(-1.7%)
	n2n	**321 (64.9%)**	43.2(6.9%)	0.387(7.8%)	**0.084 (10.0%)**
	e2n	407(16.9%)	**41.6 (15.3%)**	**0.498 (12.9%)**	0.081(9.3%)
	v2v	418(10.5%)	42.0(13%)	0.391(9.9%)	0.058(3.7%)

Table 11: Evaluation results on the OC20 S2EF **out-of-distribution (catalysts)** validation set.

		OC20 S2EF Validation (**out-of-distribution (catalysts)**)			
	Model	Energy MAE meV ↓	Force MAE meV/Å ↓	Force cos ↑	EFwT % ↑
same	*S*: PaiNN-small	467	42	0.341	0.13
	T: PaiNN	412	39.2	0.369	0.23
	Vanilla (1)	466(4.9%)	42.8(-28.4%)	0.336(-16.9%)	0.11(-20%)
	Vanilla (2)	439(52.4%)	45.4(-120.6%)	0.306(-120.8%)	0.12(-10%)
	n2n	**437 (56.3%)**	**42.0 (1%)**	**0.343 (8.2%)**	**0.14 (11.2%)**
	v2v	444(43.4%)	42.4(-12.8%)	0.342(3.6%)	0.12(-12%)
	S: GemNet-OC-small	335	28.9	0.506	0.85
	T: GemNet-OC	288	24.0	0.576	1.68
	Vanilla (1)	339(-9.3%)	29.0(-1.1%)	0.507(0.9%)	0.85(-0.4%)
	Vanilla (2)	318(35.4%)	28.9(-0.1%)	0.507(1.0%)	**1.05 (24.1%)**
	n2n	**309 (54.4%)**	28.8(2.0%)	0.508(2.6%)	1.02(20.5%)
	e2e	324(21.6%)	**27.6 (26.5%)**	**0.524 (25.1%)**	0.95(12.5%)
similar	*S*: SchNet	1205	61.6	0.205	0
	T: PaiNN	412	39.2	0.369	0.23
	Vanilla (1)	**1122 (10.6%)**	**58.7 (12.9%)**	**0.234 (15.9%)**	**0.01 (4.3%)**
	Vanilla (2)	1122(10.5%)	58.8(12.5%)	0.23(15.2%)	0(0%)
	n2n	1150(6.9%)	59.4(9.8%)	0.225(12.3%)	0(0%)
different	*S*: PaiNN	412	39.2	0.369	0.23
	T: GemNet-OC	288	24	0.576	1.68
	Vanilla (1)	423(-8.8%)	39.1(0.8%)	0.371(1.1%)	0.230(0.0%)
	Vanilla (2)	400(9.5%)	43.6(-29%)	0.320(-23.5%)	0.23(0.2%)
	n2n	**345 (54.0%)**	38.5(4.7%)	0.383(6.8%)	**0.433 (14%)**
	e2n	401(8.9%)	**37.4 (11.9%)**	**0.395 (12.3%)**	0.317(6.0%)
	v2v	424(-10.7%)	38.2(6.5%)	0.386(8.2%)	0.187(-3%)

Table 12: Evaluation results on the OC20 S2EF **out-of-distribution (both)** validation set.

		OC20 S2EF Validation (**out-of-distribution (both)**)			
	Model	Energy MAE meV ↓	Force MAE meV/Å ↓	Force cos ↑	EFwT % ↑
same	*S*: PaiNN-small	610	56.8	0.346	0.02
	T: PaiNN	554	59.2	0.379	0.03
	Vanilla (1)	648(-67.6%)	61.1(-179.2%)	0.056(-900.3%)	0.02(0%)
	Vanilla (2)	592(32.2%)	60.6(-158.3%)	0.310(-112.4%)	0.02(0%)
	n2n	557(94.2%)	**56.0 (33%)**	0.351(14.8%)	0.018(-15.8%)
	v2v	**547 (112.6%)**	56.5(12.3%)	**0.352 (17.3%)**	**0.025 (43.3)**
	S: GemNet-OC-small	424	37.4	0.533	0.11
	T: GemNet-OC	370	31.0	0.606	0.23
	Vanilla (1)	412(23.1%)	37.5(-1.8%)	0.534(1.2%)	0.11(-2.3%)
	Vanilla (2)	401(43.2%)	37.4(0.2%)	0.535(2.0%)	0.12(8.5%)
	n2n	**395 (53.7%)**	37.3(1.6%)	0.537(4.4%)	0.12(8.5%)
	e2e	412(22.2%)	**35.5 (29.5%)**	**0.554 (29.0%)**	**0.13 (19.9%)**
similar	*S*: SchNet	1450	78.4	0.202	0
	T: PaiNN	554	59.2	0.379	0.03
	Vanilla (1)	**1350 (11.2%)**	75.9(13%)	**0.227 (14.3%)**	**0.0025 (1.8%)**
	Vanilla (2)	1358(10.2%)	**75.7 (14.1%)**	0.226(13.8%)	0(0%)
	n2n	1396(6.1%)	76.2(11.5%)	0.220(10.4%)	0(0%)
different	*S*: PaiNN	554	59.2	0.379	0.03
	T: GemNet-OC	370	31	0.606	0.23
	Vanilla (1)	558(-2.3%)	53.8(19.0%)	0.380(0.6%)	0.031(-0.5%)
	Vanilla (2)	511(23.3%)	323.1(-935.7%)	0.326(-23.2%)	0.027(-2.6%)
	n2n	**448 (57.4%)**	52.4(24.1%)	0.394(2.4%)	0.056(12.1%)
	e2n	536(9.6%)	**50.1 (32.4%)**	**0.407 (12.5%)**	**0.057 (12.6%)**
	v2v	538(8.5%)	50.5(31.0%)	0.402(10.4%)	0.035(1.7%)

C Baseline results on COLL

In Table 13, we present the performance and inference throughput of the baseline models on COLL. As the systems are much smaller than those in OC20, the throughput is a lot larger than the one observed in Table 2. Qualitatively, however, we observe the same, clear trade-off between accuracy and throughput.

Table 13: Evaluation of the performance of the four baseline models on the COLL dataset.

	Inference Throughput	COLL test set			
Model	Samples / GPU sec. ↑	Energy MAE meV ↓	Force MAE meV/Å ↓	Force cos ↑	EFwT % ↑
SchNet	44000	146.5	121.2	0.970	2.75
PaiNN-small	29000	104.0	80.9	0.984	5.4
PaiNN	13000	85.8	64.1	0.988	10.1
GemNet-OC	3520	44.8	38.2	0.994	20.2

D Data augmentation

We investigated data augmentation as a way of distilling knowledge from GemNet-OC into PaiNN on the OC20 dataset.

D.1 Data jittering

To create additional data, we added noise to the atomic positions of the training samples and then used the teacher to label the newly derived samples. We tried two different approaches: Random noise, and optimizing the positions using gradient ascent such that the difference between the predictions of the student and teacher was maximized as done in [45]. Denoting the noise as δ, we obtain the noise atomic positions as $\boldsymbol{X}_\delta = \boldsymbol{X} + \delta$. Let the student and teacher models be denoted as f_s and f_t respectively, we then obtained the noise δ' by initializing this as $\mathbf{0}$ and

$$\begin{aligned}\mathcal{L}_\text{KD} &= \mathcal{L}_0(f_s(\boldsymbol{X}_\delta, \boldsymbol{z}), f_t(\boldsymbol{X}_\delta, \boldsymbol{z})) \\ \delta' &= \delta + \alpha \nabla_\delta \mathcal{L}_\text{KD}.\end{aligned}$$

However, this becomes computationally expensive, as it requires additional gradients for $\nabla_\delta \mathcal{L}_\text{KD}$. Hence, we settled on using a single step, where we fixed the norm of δ to avoid going too far away from the real structure. We experimented with different norms, with the smallest being 0.1 Å. We compared this to using random directions with a fixed norm, and we did not see any improvements when using the more computationally expensive gradient ascent approach. In both cases, the noise was added to all the samples in the batch.

D.2 Synthetic data

Combined dataset 2M+d1M. We generated 1M synthetic samples by first drawing 100k random adsorbate and catalyst combinations (systems) and then running relaxations with a pre-trained GemNet-OC model. Out of these relaxations with 100 steps on average (200 max), we randomly draw approx. 10% to obtain 1M samples.

In the next step, we combine the 1M samples with the 2M OCP dataset, which is based on DFT relaxations. Directly working with this combined dataset means iterating over a 1-to-2 ratio of samples from each subset in an epoch of 3M samples. To control this ratio, we define the target ratio of samples from the synthetic dataset during training $\alpha_{target} \in [0, 1]$. Setting $\alpha_{target} = 0.5$ means that per epoch we iterate over the 1M dataset 1.5 times and over the 2M dataset 0.75 times.

Different weighting in loss depending on origin. Next to specifying a sampling ratio of samples from the synthetic dataset, we can also specify how to weight the contribution of samples to the loss based on their origin (DFT or synthetic). To achieve this, we specify the weighting ratio of synthetic to DFT samples $r_\text{s/dft} = w_\text{s}/w_\text{dft} \in \mathbb{R}^+$. In each batch, we compute a weighting factor in front of the synthetic w_s and DFT w_dft samples satisfying the conditions

$$w_\text{s} \cdot \alpha_\text{batch} + w_\text{dft} \cdot (1 - \alpha_\text{batch}) = 1, \tag{9}$$

where α_batch is the ratio of synthetic to DFT samples in a batch. Hence, when we combine DFT and synthetic data - $\alpha_\text{batch} \in (0, 1)$, we derive the following weights:

$$w_\text{dft} = (1 - \alpha_\text{batch} + \alpha_\text{batch} \cdot r_\text{s/dft})^{-1}, \tag{10}$$

$$w_\text{s} = r_\text{s/dft} \cdot w_\text{dft}. \tag{11}$$

Likewise, when $\alpha_\text{batch} = \{0, 1\}$ - i.e., we either train on DFT or synthetic data exclusively, the corresponding weighting coefficient (w_s or w_dft) is naturally equal to 1.

E Hyperparameter studies

We additionally investigated different aspects of our KD protocols, which we present below. We have performed these experiments when distilling GemNet-OC into PaiNN on the OC20-2M dataset.

E.1 Effect of losses

In our experiments, we have used MSE as the loss $\mathcal{L}_{\text{feat}}$. However, in the general framework in Equation (3), there are other choices that are possible, e.g., more advanced losses like Local Structure Preservation [34] and Global Structure Preservation (GSP) [35]. We therefore initially experimented with using these alternative losses when distilling GemNet-OC into PaiNN on OC20. However, the initial experiments showed that MSE worked well, and in particular, a lot better than the more advanced GSP and LSP losses. We therefore settled on using MSE as our $\mathcal{L}_{\text{feat}}$. In Table 14, we present the average performance over all four validation splits when using these different losses in the *n2n* and *e2n* settings.

We also experimented with trying to optimize the CKA directly (as we saw that the CKA similarity improved when using distillation), but it did not work and we did not pursue it any further.

Table 14: Comparing different loss functions $\mathcal{L}_{\text{feat}}$ with GemNet-OC as teacher and PaiNN as student on OC20, using the *n2n* and *e2n* KD protocols. In the case of LSP + *e2n*, the model completely failed when predicting forces on the *ood (both)* validation set, leading to a force MAE of $464\ \text{meV}/\text{Å}$, which is almost a factor 10 larger than the other models on the same split. We have therefore written this value as "-". When evaluating a checkpoint for a model which was trained half as long, the error on this split was $53.7\ \text{meV}/\text{Å}$, and the average over all four validation splits was $44.8\ \text{meV}/\text{Å}$.

		OC20 validation set			
	Loss	Energy MAE meV ↓	Force MAE meV/Å ↓	Force cos ↑	EFwT % ↑
n2n	MSE	346	42.8	0.393	0.262
	GSP	427	46.1	0.356	0.124
	LSP	398	44.8	0.367	0.159
e2n	MSE	430	41.3	0.405	0.195
	GSP	463	45.2	0.363	0.119
	LSP	441	-	0.380	0.134

E.2 Effect of transformations

We have evaluated using different transformations $\mathcal{M}_{\text{s}}$, i.e., transformations of the student features before applying the loss $\mathcal{L}_{\text{feat}}$. We tried either using the identity function, (i.e., not using a transformation at all), a linear transformation (i.e., multiplication with matrix and adding a bias vector), or using a multilayer perceptron (MLP) with one hidden layer. We conducted our experiments when distilling G We found that using an MLP worsened the results, and for *e2n*, there was not a big difference between using a linear layer and no transformation at all. For *n2n*, the node features in PaiNN and GemNet-OC are of different dimensions, and we can therefore not use the identity transformation when using the MSE loss.

The results from the experiments are presented in Table 15.

E.3 Effect of feature selection

GemNet-OC consists of an initial embedding layer, followed by a series of interaction layers. The result of each embedding/interaction layer is used as input into the next layer, while a copy is also processed by an "output layer". To make the final prediction, the results of the different output layers are concatenated and processed by a final MLP. This means that, for each embedding/interaction layer, we have two features that could potentially be distilled: the feature used as input for the next layer, or the result of the output layer. Additionally, we could use features from inside the final MLP which make the prediction by processing the concatenated output features.

Table 15: Comparing different transformation functions $\mathcal{M}$s (Identity, a linear layer and an MLP with one hidden layer) with GemNet-OC as teacher and PaiNN as student, using the *n2n* and *e2n* KD protocols. As the node features in GemNet-OC and PaiNN are of different dimensions, we cannot use the identity transformation when using *n2n*.

		OC20 validation set			
	Loss	Energy MAE meV ↓	Force MAE meV/Å ↓	Force cos ↑	EFwT % ↑
n2n	Identity	-	-	-	-
	Linear	346	42.8	0.393	0.262
	MLP	363	45.1	0.367	0.147
e2n	Identity	430	41.3	0.405	0.195
	Linear	418	41.3	0.405	0.207
	MLP	427	43.2	0.387	0.161

Initially, we used the feature after the final interaction layer when distilling knowledge from GemNet-OC. However, we found that using the feature just before the final linear layer in the final MLP gave a drastic improvement in performance. We, therefore, set out to investigate how the choice of features impacted the results in more detail.

We performed these experiments when distilling GemNet-OC into PaiNN on OC20 using our *n2n* strategy, and the results presented here are on a set of 30 thousand samples sampled from the in-distribution validation set. We did not perform any extensive hyperparameter tuning, but chose λ such that $\lambda\mathcal{L}_{\text{KD}}$ (the distillation loss term) was initially roughly the same for all choices.

Choice of GemNet-OC layer. In Figure 4, we present the training curves when fixing the choice of feature in PaiNN and varying the choice of features in GemNet-OC. The overall trend is that closer to the output is better: even using the features from the early output layers is better than using features from later interaction layers. Our results suggest that for forces, it is better to use features from earlier output layers. However, we think this could be due to the choice of λ, as we have empirically found that the weighting of the loss term in *n2n* could offer a trade-off between energy and force performance.

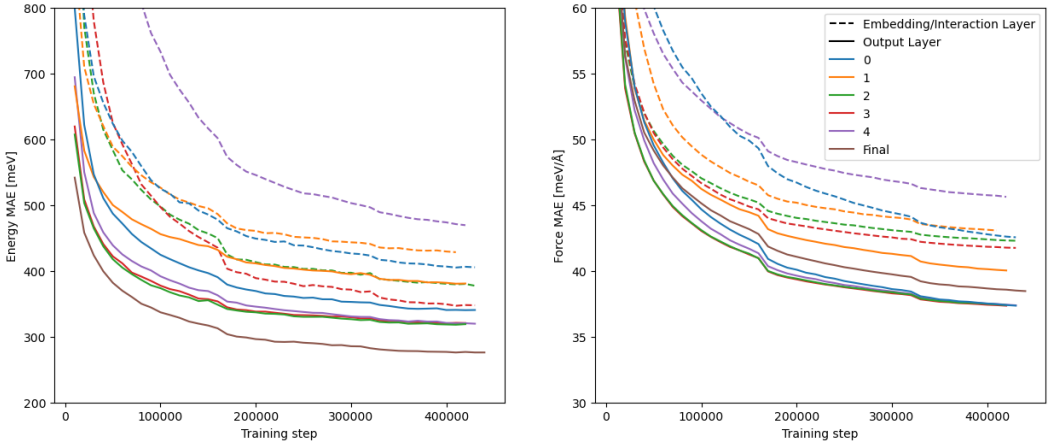

Figure 4: Evaluation error as we vary the features to distill from in the teacher model - energy MAE (*left*), and force MAE (*right*). *n2n* KD from GemNet-OC to PaiNN. Performance is evaluated on a validation subset comprising 30k samples. The numbers 0 to 4 indicate at what stage the feature has been extracted, with 0 meaning after the embedding layer, and 1 to 4 after the corresponding interaction layer. Solid and dashed lines indicate if the feature is the result of an embedding/interaction layer, or an output layer, respectively. "Final" refers to the feature extracted right before the final linear prediction layer.

Choice of PaiNN layer. PaiNN consists of a sequence of blocks, where each block consists of a message layer and an update layer. In Figure 5, we present training curves when fixing the choice of features in GemNet-OC (the feature just before the final linear layer), and varying the choice of features in PaiNN (choice of block, and either the feature after the corresponding message or update layer). The results here indicate that using deeper features leads to better results.

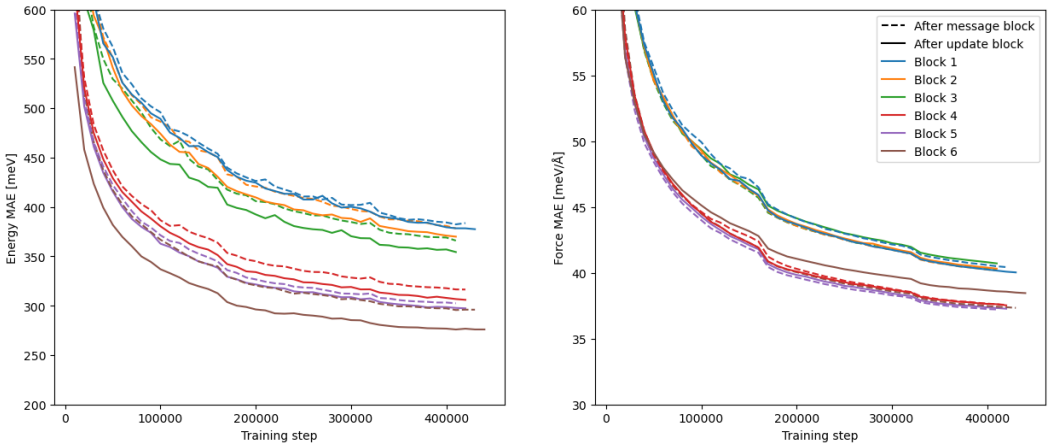

Figure 5: Evaluation error as we vary the features to distill into in the student model - energy MAE (*left*), and force MAE (*right*). *n2n* KD from GemNet-OC to PaiNN. Performance is evaluated on a validation subset comprising 30k samples. The different colors indicate after which block the features have been extracted, and dashed and solid lines indicate if features were extracted after the message or update layers, respectively.

Conclusion. The conclusion we draw from this study is that using features as close to the output as possible improves KD performance in our setup. However, these results are only empirical, and more investigation could be done. For example, if it is possible to beforehand determine which pairs of features should be used (and not having to rely on trial-and-error).

F Error bars

To get an idea of the stability of our KD protocols, we perform additional experiments distilling GemNet-OC into PaiNN using three different seeds and compute standard deviations. We present these results in Table 16.

Table 16: Performance of KD from GemNet-OC into PaiNN across 3 different seeds, averaged over all validation splits. The numbers are presented as mean $\pm$ one standard deviation. The missing force error for the baseline model is due to one of the seeds completely failing on the out-of-distribution (both) split, drastically increasing the error. The other two seeds had force MAEs of 43.8 and 45.3 $\mathrm{meV/Å}$, respectively.

	OC20 validation set			
Loss	Energy MAE $\mathrm{meV}\downarrow$	Force MAE $\mathrm{meV/Å}\downarrow$	Force cos $\uparrow$	EFwT $\%\uparrow$
None (baseline)	440 ± 8	-	0.376 ± 0.0018	0.143 ± 0.0051
n2n	346 ± 0.7	43.2 ± 0.6	0.392 ± 0.0017	0.256 ± 0.011

G Explainability

We utilize CKA similarity scores to monitor the effect of KD throughout our studies. Here, we present a selection of the analyses we have performed, summarizing how KD influences the similarity between teacher and student models across teacher-student configurations (Figure 6); across KD protocols (Figure 7); and feature selections (Figure 8). We found out that such similarity metrics can be effectively used to examine and profile different KD approaches, as well as as a potential debugging tool. We also explored the utility of CKA (in conjunction with measures of the predictive ability of individual features) as a means to inform the design of (optimal) KD strategies and feature selection protocols *a priori*, but the results were not conclusive to include here.

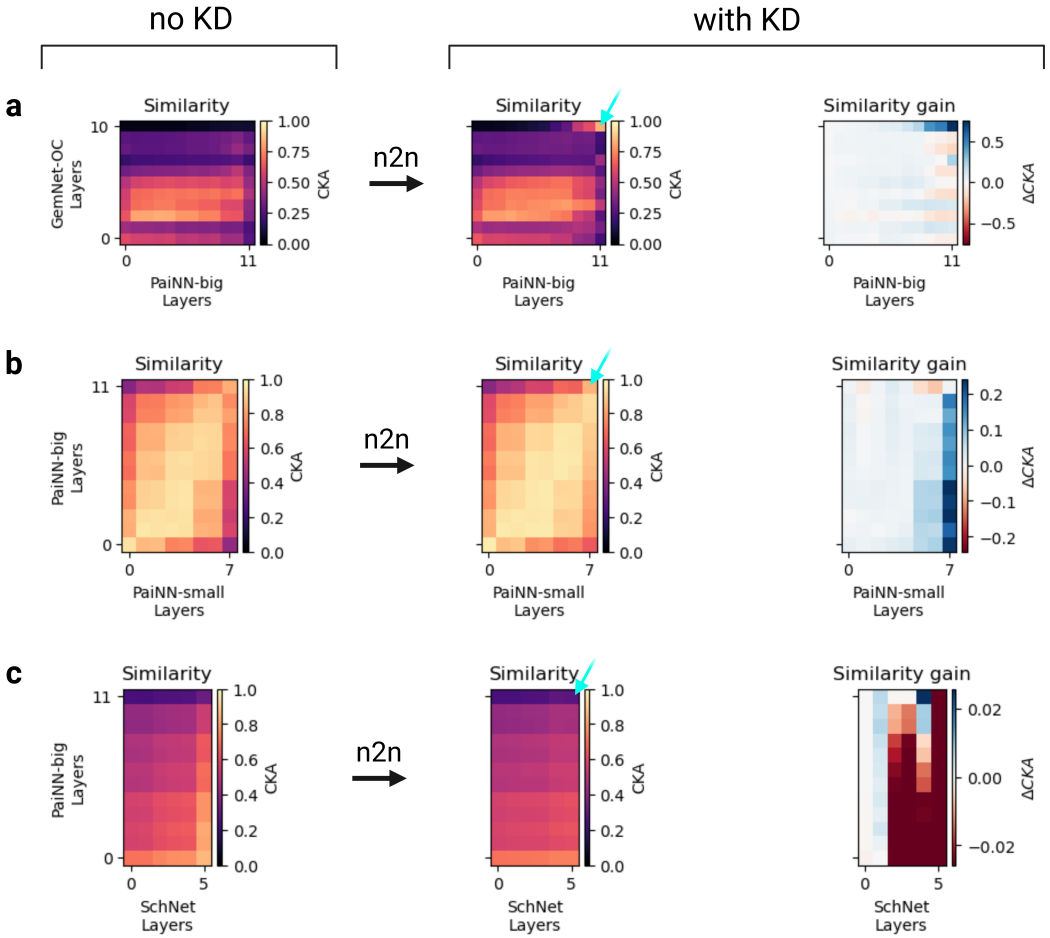

Figure 6: We explore the effect of *n2n* KD on the feature similarity between different student-teacher configurations: (a) GemNet-OC -> PaiNN; (b) PaiNN -> PaiNN-small; (c) PaiNN -> SchNet. The layer pair that was used in each experiment is indicated with a ↙. Note the scale difference in the *Similarity gain* plots.

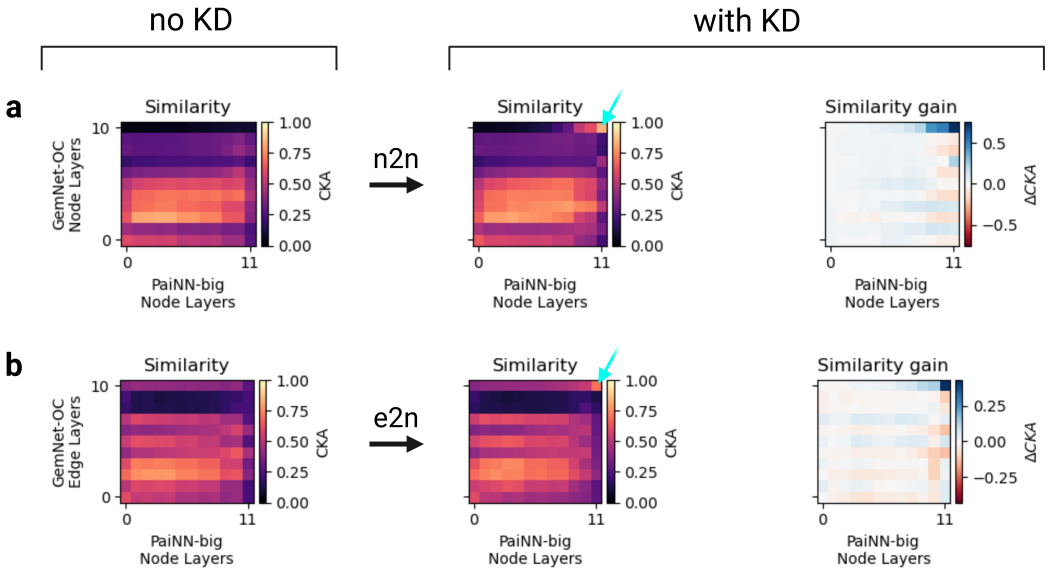

Figure 7: We explore the effect of different KD strategies - *n2n* and *e2n* KD on the feature similarity between the student and the teacher models. This is computed for GemNet-OC -> PaiNN: (a) *n2n*; (b) *e2n*. The layer pair that was used in each experiment is indicated with a ↙. Note the scale difference in the *Similarity gain* plots.

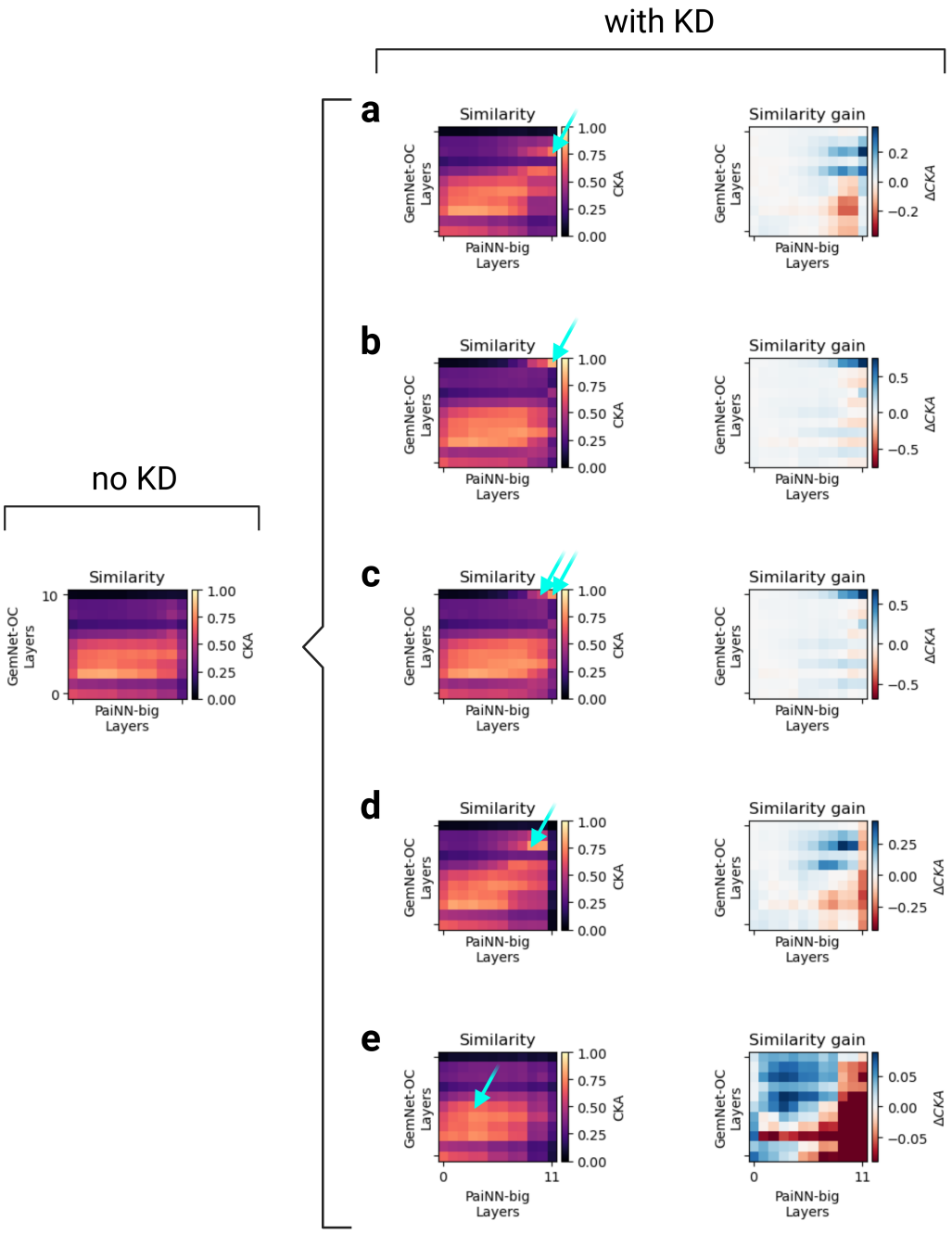

Figure 8: We explore the effect of feature selection in KD on feature similarity between the student and the teacher models: (a) H4->U6; (b) CONCAT+MLP->U6; (c) CONCAT+MLP->M6+U6; (d) H4->U5; (e) X2->M2. The layer pair that was used in each experiment is indicated with a ↙. Note the scale difference in the *Similarity gain* plots.

H Training times

One caveat of knowledge distillation is that it inherently increases the training time of the student model. In our offline KD setup, we need to perform additional forward passes through the teacher to extract representations to distill to the student. However, it is important to note that, despite increasing the computational time per training step, we observed that models trained with KD can outperform their baseline counterparts even when compared at the same training time point (Figure 9), despite the latter having been trained for more steps/epoch in total. This means that, all in all, we can use KD to enhance the predictive accuracy in models without necessarily impacting training times.

However, we make the following remark. In this experiment, we utilized publicly available pre-trained Gemnet-OC model weights, and therefore did not have to train the teacher model ourselves. However, when access to a pre-trained teacher model is not available, one should also account for the time required to train the teacher.

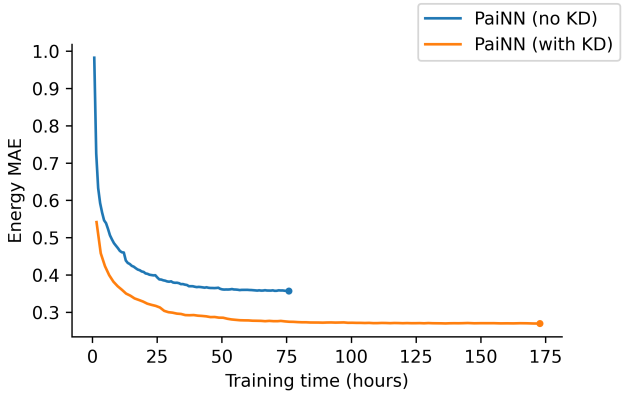

Figure 9: Energy validation error of PaiNN without (*blue*) and with (*orange*) knowledge distillation from GemNet-OC, trained for the same number of steps (1 million). Validation on a random sample of size 30k samples from the in-distribution OC20 validation set.