Coherent energy and force uncertainty in deep learning force fields

Peter Bjørn Jørgensen*
Alexandra Institute
2300 Copenhagen, Denmark
peterbjorgensen@gmail.com

Jonas Busk
Technical University of Denmark
2800 Kongens Lyngby

Ole Winther
Technical University of Denmark
2800 Kongens Lyngby

Mikkel N. Schmidt
Technical University of Denmark
2800 Kongens Lyngby

Abstract

In machine learning energy potentials for atomic systems, forces are commonly obtained as the negative derivative of the energy function with respect to atomic positions. To quantify aleatoric uncertainty in the predicted energies, a widely used modeling approach involves predicting both a mean and variance for each energy value. However, this model is not differentiable under the usual white noise assumption, so energy uncertainty does not naturally translate to force uncertainty. In this work we propose a machine learning potential energy model in which energy and force aleatoric uncertainty are linked through a spatially correlated noise process. We demonstrate our approach on an equivariant messages passing neural network potential trained on energies and forces on two out-of-equilibrium molecular datasets. Furthermore, we also show how to obtain epistemic uncertainties in this setting based on a Bayesian interpretation of deep ensemble models.

1 Introduction

In recent years, the use of machine learning force fields have enabled studies of atomic systems that are otherwise out of reach because of the computational complexity of traditional electronic structure methods such as density functional theory (DFT). Deep learning force fields are typically trained on large datasets of molecular energies and forces computed using DFT, and to increase data efficiency training data can be collected sequentially, guided by the model uncertainty. Thus, robust uncertainty estimates are crucial to optimize data collection and generally increase interpretability of predictions.

In machine learning force fields, forces are often obtained by computing the partial derivatives of the potential energy surface with respect to the atom positions, which ensures that the derived force field is conservative. Futhermore, when the potential energy model is invariant to rotations and translations, the force field is equivariant to rotations.

Mean-variance networks and deep ensemble models [8] are commonly used to obtain uncertainty estimates from deep learning models [3, 2]. However, given a mean-variance network for the energy, deriving the force uncertainty through the derivative is not possible under the usual assumption of white/uncorrelated noise. This difficulty has been noted by several authors [4, 3]. Gasteiger et al. [4] writes: *"There is thus no general way of estimating σ_F for these kinds of models. Instead, we have to rely on σ_E as the uncertainty measure and hope that it correlates with the force error."* Notice

*Majority of the work was done while still at Technical University of Denmark.

Presented at Advancing Molecular Machine Learning - Overcoming Limitations [ML4Molecules], ELLIS workshop, VIRTUAL, December 8, 2023, unofficial NeurIPS 2023 side-event.

that the force standard deviation σ_F and the energy standard deviation σ_E have different physical units, so while hoping that the two might be correlated, we would at least have to correct the units to obtain meaningful uncertainties. Concurrently with our work, Carrete et al. [3] found that the force variance can not be derived directly from the mean-variance network for the energy because we need a differentiable function for the covariance of the energy observations at two arbitrary points at close proximity. They therefore workaround the problem by expanding the model to predict a separate σ_F for each individual atom as also done in [2].

In this work we propose to relax the white noise assumption for the noise and derive closed form expressions for the mean and variance of the forces directly from the potential energy model. We show that a parameter naturally arises, which can be trained or adjusted post-hoc and can be understood as the squared inverse length scale of the noise process.

2 White noise model

Given a sequence of atomic numbers $\boldsymbol{z} = (z_1, \ldots, z_i, \ldots, z_N) \in \mathbb{N}$ and corresponding atomic positions $\boldsymbol{r} = (\vec{r}_1, \ldots \vec{r}_i, \ldots, \vec{r}_N) \in \mathbb{R}^D$, the underlying assumption of the machine learning potential energy model is that the energy observations are obtained as:

$$E_{\text{obs}}(\boldsymbol{z}, \boldsymbol{r}) = E_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{r}) + \rho_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{r})\epsilon\,, \tag{1}$$

where $E_{\boldsymbol{\theta}}(\cdot)$ and $\rho^2_{\boldsymbol{\theta}}(\cdot)$ are the mean and variance outputs, respectively, of a machine learning potential with parameters $\boldsymbol{\theta}$ and ϵ is zero mean, random noise of unit variance. The model is roto-translationally invariant when

$$E_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{r}) = E_{\boldsymbol{\theta}}(P\boldsymbol{z}, RP\boldsymbol{r} + t) \tag{2}$$

for any permutation P, rotation R and translation t. The typical assumption is that ϵ is white noise, i.e. its autocorrelation function $\mathbb{E}\left[\epsilon(\boldsymbol{z}, \boldsymbol{r})\epsilon(\boldsymbol{z}', \boldsymbol{r}')\right]$ is 1 when $(\boldsymbol{z}, \boldsymbol{r})$ and $(\boldsymbol{z}', \boldsymbol{r}')$ represent the same configuration, $(\boldsymbol{z}, \boldsymbol{r}) = (P\boldsymbol{z}', RP\boldsymbol{r}' + t)$, and 0 otherwise.

We say that a function $f(t)$ is differentiable at a point $t = t_0$ if the limit exists:

$$f'(t_0) = \lim_{\Delta t \to 0} \frac{f(t_0 + \Delta t) - f(t_0)}{\Delta t}. \tag{3}$$

To extend the notion of a derivative to a stochastic process, we can say that a random process $X(t)$ is differentiable in the mean-squared sense at a point $t = t_0$ if there exists a stochastic process $X'(t)$ such that

$$\lim_{\Delta t \to 0} \mathbb{E}\left[\left(\frac{X(t_0 + \Delta t) - X(t_0)}{\Delta t} - X'(t_0)\right)^2\right] = 0. \tag{4}$$

If we attempt to take the partial derivative of equation 1 with respect to one of the atomic coordinates, we will see that the limit in equation 4 does not exist because of the discontinuity of the autocorrelation function of $\epsilon(\boldsymbol{z}, \boldsymbol{r})$ with respect to $\boldsymbol{r}$. A typically used workaround is to model the force uncertainty independently from the energy uncertainty, i.e. we assume that the force observations are obtained by a separate noise process, such that the observed force on the ith atom in the dth spatial dimension is coming from this process:

$$f_{\text{obs},i,d}(\boldsymbol{z}, \boldsymbol{r}) = -\frac{\partial E_{\boldsymbol{\theta}}}{\partial r_{i,d}}(\boldsymbol{z}, \boldsymbol{r}) + \omega_{\boldsymbol{\theta},i}(\boldsymbol{z}, \boldsymbol{r})\epsilon_{i,d}\,, \tag{5}$$

where $\omega^2_{\boldsymbol{\theta},i}$ is the variance output from the deep learning potential for each atom and $\epsilon_{i,d}$ is zero mean, random noise of unit variance. A simplifying assumption is that the noise level is the same in all the spatial dimensions, but the model could be extended to different noise levels in each direction using equivariant vectorial outputs from the model.

3 Colored noise model

The form of the energy predictions of this model is the same as in equation 1 and we use the same symbols for the outputs of the machine learning potential, even though they are not directly transferable from one setting to the other. The energy observations are thus obtained as:

$$E_{\text{obs}}(\boldsymbol{z}, \boldsymbol{r}) = E_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{r}) + \rho_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{r})\eta\,, \tag{6}$$

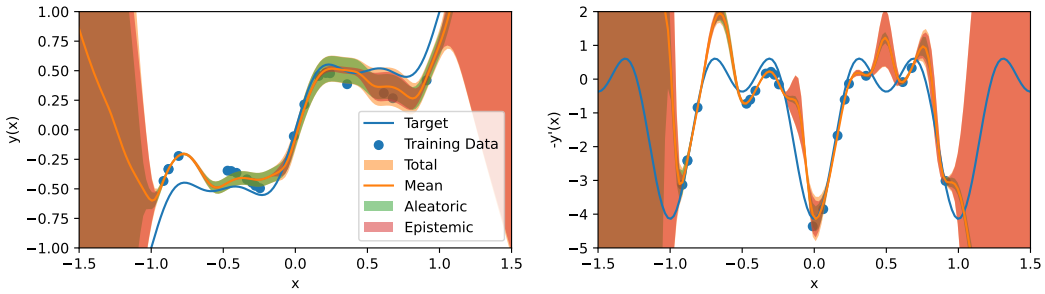

Figure 1: Ensemble model (with 10 instances) fitted to energy (left) and force (right) observations with correlated noise.

where η is a stochastic process with a differentiable autocorrelation function $\mathbb{E}_{\eta\eta}\left[\eta(\boldsymbol{z},\boldsymbol{r})\eta(\boldsymbol{z}',\boldsymbol{r}')\right]$ function with respect to the atom positions $\boldsymbol{r}$ and $\boldsymbol{r}'$ and η is zero mean and unit variance for all $\boldsymbol{z}$ and $\boldsymbol{r}$. The mean and variance for $E_{\text{obs}}(\boldsymbol{z},\boldsymbol{r})$ are still given directly from the machine learning potential outputs $E_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{r})$ and $\rho^2_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{r})$, respectively. Another simplifying assumption is that the noise process η is wide-sense stationary, i.e. its mean does not change with $\boldsymbol{z},\boldsymbol{r}$ and the autocorrelation function is (locally) a function of the difference between the atom positions $\mathbb{E}_{\eta\eta}\left[\eta(\boldsymbol{z},\boldsymbol{r})\eta(\boldsymbol{z}',\boldsymbol{r}')\right] = R^{\boldsymbol{z}\boldsymbol{z}'}_{\eta\eta}(\boldsymbol{r}-\boldsymbol{r}')$. The force mean is the same as in equation 5, namely $-\frac{\partial E_{\boldsymbol{\theta}}}{\partial r_{i,d}}(\boldsymbol{z},\boldsymbol{r})$, while we get the following expression for the force variance (derived in appendix A):

$$\text{Var}\left(-\frac{\partial E_{\text{obs}}}{\partial r_{i,d}}(\boldsymbol{z},\boldsymbol{r})\right) = -\frac{\partial^2 R^{\boldsymbol{z}\boldsymbol{z}'}_{\eta\eta}(\Delta\boldsymbol{r})}{\partial \Delta r^2_{i,d}}\Big|_{\Delta\boldsymbol{r}=\boldsymbol{0}} \rho^2_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{r}) + R^{\boldsymbol{z}\boldsymbol{z}'}_{\eta\eta}(\boldsymbol{0})\left(\frac{\partial \rho_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{r})}{\partial r_{i,d}}\right)^2. \tag{7}$$

Notice that the force variance only depends on the noise autocorrelation through its value at $\Delta\boldsymbol{r}=0$ and its second derivative at $\Delta\boldsymbol{r}=0$. Since η is zero mean and unit variance we get $R^{\boldsymbol{z}\boldsymbol{z}'}_{\eta\eta}(\boldsymbol{0})=1$. If η were a Gaussian process with the exponentiated quadratic kernel $R(\boldsymbol{r},\boldsymbol{r}') = \exp\left(\frac{||\boldsymbol{r}-\boldsymbol{r}'||^2}{2\ell^2}\right)$ we would refer to $\gamma = -\frac{\partial^2 R^{\boldsymbol{z}\boldsymbol{z}'}_{\eta\eta}(\Delta\boldsymbol{r})}{\partial \Delta r^2_{i,d}}\Big|_{\Delta\boldsymbol{r}=\boldsymbol{0}} = \frac{1}{\ell^2}$ as the inverse *length scale* squared of the kernel. Hence, when we train our neural network potential we can treat the inverse length scale as a hyperparameter $\hat{\gamma}$ and estimate it from the training data. The force variance thus becomes:

$$\text{Var}\left(-\frac{\partial E_{\text{obs}}}{\partial r_{i,d}}(\boldsymbol{z},\boldsymbol{r})\right) = \hat{\gamma}\rho^2_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{r}) + \left(\frac{\partial \rho_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{r})}{\partial r_{i,d}}\right)^2. \tag{8}$$

If we drop the wide-sense stationary assumption the model can be generalised to multiple inverse length scales and we would predict the γ parameters as another output of the model for each atom, i.e. $\hat{\gamma}_{\boldsymbol{\theta},i}(\boldsymbol{z},\boldsymbol{r})$, but we would also need to predict another quantity of the noise process, specifically $\zeta = \frac{\partial R^{\boldsymbol{z}\boldsymbol{z}'}_{\eta\eta}(\boldsymbol{r},\boldsymbol{r}')}{\partial r'_{i,d}}\Big|_{\boldsymbol{r}'=\boldsymbol{r}}$ (see equation 25 in appendix A). To get epistemic uncertainty predictions from the model we use a Bayesian interpretation of deep ensemble models [16, 6, 5]. See Appendix B.

4 Toy Example

In this example we fit an ensemble of multilayer perceptron neural networks to a synthetic 1-dimensional dataset that is generated using the same underlying assumptions as the model. The noise free energy function is $y(x) = x + 0.3\sin(2\pi x) + 0.1\sin(4\pi x)$. The additive noise is sampled from a Gaussian process with kernel $k(a,b) = \alpha^2\left(\exp\left(-\frac{(a-b)^2}{2\ell^2}\right) + \delta(a-b)10^{-4}\right)$ with $\alpha = 0.2$ and $\ell = 0.5$. We sample 20 training examples uniformly between -1 and 1 and the resulting predictions are shown in figure 1. The individual models are trained with maximum likelihood using a normal distribution parameterised with the mean and variance expressions derived in section 3. As expected, the predicted mean function does not follow the true energy function because the added noise function is a single realisation from a Gaussian process prior. The aleatoric uncertainty is small when the samples are widely spaced, but increases when the samples are close together, which allows the

Table 1: Test results of ensemble models ($M = 5$) with different noise assumptions trained on the ANI-1x (A1x) and Transition1x (T1x) datasets. Energy errors are averaged over molecules, while force errors are computed component-wise and averaged over the spatial dimensions and atoms.

Data	**Model**	**Energy (eV)**						
		MAE↓	RMSE↓	NLL↓	RZV	ENCE↓	CV	RMV
A1x	Vanilla	0.013	0.031	-2.13	1.83	0.77	1.05	0.016
	White	0.010	0.028	-3.13	0.69	0.31	1.54	0.033
	Colored	0.012	0.026	-2.56	0.42	0.56	0.97	0.048
T1x	Vanilla	0.033	0.060	-0.83	2.10	1.06	0.73	0.029
	White	0.039	0.064	-1.67	1.04	0.18	0.47	0.051
	Colored	0.037	0.063	-1.75	0.93	0.21	0.48	0.054
		Forces (eV)						
		MAE↓	RMSE↓	NLL↓	RZV	ENCE↓	CV	RMV
A1x	Vanilla	0.018	0.037	-1.74	1.89	0.76	1.31	0.027
	White	0.017	0.041	-2.67	0.70	0.29	1.32	0.052
	Colored	0.018	0.039	-2.51	0.66	0.32	1.13	0.051
T1x	Vanilla	0.037	0.075	-1.56	1.54	0.47	1.08	0.057
	White	0.038	0.076	-1.86	0.86	0.18	0.84	0.076
	Colored	0.038	0.075	-1.78	0.79	0.24	0.76	0.074

model to trade precision for smoothness. We also see that the epistemic uncertainty is large in regions where there are no training samples and decreases as we get closer to the training samples, while the aleatoric uncertainty extrapolates poorly outside of the training interval with high uncertainty for negative x and small uncertainty for positive x.

5 Application to Molecular Data

We apply the proposed method to an ensemble of equivariant message passing neural networks, specifically an extension of the PaiNN [12] model. The proposed method is evaluated on two publicly available datasets designed specifically for the development and evaluation of ML potentials, ANI-1x [15] and Transitions1x [11]. The datasets include different compositions and out-of-equilibrium structures and contain a wide distribution of energies and forces. See Appendix C for details.

For comparison we also train a *vanilla* ensemble model, consisting of 5 models without variance outputs, but it still has epistemic uncertainty given by the variances across the ensemble. We also compare with a *white noise* ensemble model, that uses the usual assumption of uncorrelated noise. The results are summarised in Table 1. The symbol ↓ means that lower values are better. The metrics are described in detail in Appendix E. In general we see that the accuracy in terms of mean absolute error (MAE) and root mean squared error (RMSE) of all three model variants are in the same ballpark and comparable to previous work [2], i.e. we do not need to sacrifice prediction accuracy to include aleatoric uncertainty in the model ensemble. On average the vanilla ensemble underestimates the errors (RZV>1) while the other two models overestimate the errors (except white noise model in T1x energy) and have lower (better) negative log-likelihood. The white noise model is better calibrated (lower ENCE and NLL) than the colored noise model. The coefficient of variation (CV) is lower for the colored noise model. Deriving the energy and force uncertainty from a single output perhaps makes the model less expressive. However, the reliability diagrams Appendix E shows that the uncertainties are well-behaved and it should be possible to improve the uncertainty estimates by calibrating the uncertainties on the validation set as in [2].

6 Conclusion

We have presented a method for training single and ensemble models with coherent uncertainty estimates for energy and forces of atomic systems. This allows coherent training of energy and force uncertainty from a single output of the model. However, a parameter related to the length scale of the

noise arises, which needs to be tuned or learned. In this way, our method is akin to Gaussian process regression, but importantly it does not require the design of a kernel function and is therefore easier to apply with existing state of the art deep learning potentials. The method is not limited to deep learning force fields, but can be valuable in other domains where both the energy and gradients are observed. We hope that this perspective on uncertainty quantification will contribute to further development of machine learning force fields with theoretically grounded energy and force uncertainties.

References

[1] Jonas Busk et al. "Calibrated uncertainty for molecular property prediction using ensembles of message passing neural networks". In: *Machine Learning: Science and Technology* 3.1 (Dec. 2021), p. 015012. DOI: 10.1088/2632-2153/ac3eb3. URL: https://dx.doi.org/10.1088/2632-2153/ac3eb3.

[2] Jonas Busk et al. "Graph Neural Network Interatomic Potential Ensembles with Calibrated Aleatoric and Epistemic Uncertainty on Energy and Forces". In: *Physical Chemistry Chemical Physics* 25.37 (2023), pp. 25828–25837. DOI: 10.1039/D3CP02143B.

[3] Jesús Carrete et al. *Deep Ensembles vs. Committees for Uncertainty Estimation in Neural-Network Force Fields: Comparison and Application to Active Learning*. Feb. 2023. DOI: 10.48550/arXiv.2302.08805. arXiv: 2302.08805 [physics].

[4] Johannes Gasteiger et al. *Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules*. Apr. 2022. arXiv: arXiv:2011.14115. (Visited on 03/23/2023).

[5] F. K. Gustafsson, M. Danelljan, and T. B. Schon. "Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2020, pp. 1289–1298. DOI: 10.1109/CVPRW50498.2020.00167. URL: https://doi.ieeecomputersociety.org/10.1109/CVPRW50498.2020.00167.

[6] Lara Hoffmann and Clemens Elster. *Deep Ensembles from a Bayesian Perspective*. Nov. 2021. DOI: 10.48550/arXiv.2105.13283. arXiv: 2105.13283 [cs, stat].

[7] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Jan. 2017. DOI: 10.48550/arXiv.1412.6980. arXiv: 1412.6980 [cs]. (Visited on 10/09/2023).

[8] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in neural information processing systems* 30 (2017), pp. 6402–6413.

[9] Dan Levi et al. "Evaluating and Calibrating Uncertainty Prediction in Regression Tasks". In: *Sensors* 22.15 (2022). ISSN: 1424-8220. DOI: 10.3390/s22155540.

[10] Pascal Pernot. "Prediction uncertainty validation for computational chemists". In: *The Journal of Chemical Physics* 157.14 (2022), p. 144103. DOI: 10.1063/5.0109572. eprint: https://doi.org/10.1063/5.0109572. URL: https://doi.org/10.1063/5.0109572.

[11] Mathias Schreiner et al. "Transition1x - a dataset for building generalizable reactive machine learning potentials". In: *Scientific Data* 9.1 (Dec. 2022), p. 779. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01870-w. URL: https://doi.org/10.1038/s41597-022-01870-w.

[12] Kristof Schütt, Oliver Unke, and Michael Gastegger. "Equivariant message passing for the prediction of tensorial properties and molecular spectra". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 9377–9388.

[13] Maximilian Seitzer et al. "On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=aPOpXlnV1T.

[14] Nicki Skafte, Martin Jørgensen, and Søren Hauberg. "Reliable Training and Estimation of Variance Networks". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019. (Visited on 10/09/2023).

[15] Justin S. Smith et al. "Less is more: Sampling chemical space with active learning". In: *The Journal of Chemical Physics* 148.24 (2018), p. 241733. DOI: 10.1063/1.5023802. eprint: https://doi.org/10.1063/1.5023802.

[16] Andrew Gordon Wilson and Pavel Izmailov. “Bayesian Deep Learning and a Probabilistic Perspective of Generalization”. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS’20. Red Hook, NY, USA: Curran Associates Inc., Dec. 2020, pp. 4697–4708. ISBN: 978-1-71382-954-6.

A Covariance of energy and force observations

We can write the energy function in the following form:

$$E(x) = \mu_\theta(x) + \varepsilon(x)\sigma_\theta(x)\,, \tag{9}$$

where $\varepsilon(x)$ is a stochastic process with zero mean, unit variance and differentiable autocorrelation function $R_{\varepsilon\varepsilon}(x, x') = \mathbb{E}\left[\varepsilon(x)\varepsilon(x')\right]$. The parameters θ are random variables but not dependent on x. The covariance function is:

$$\begin{aligned}
\mathrm{Cov}(E(x), E(x')) &= \mathbb{E}\left[(\mu_\theta(x) + \varepsilon(x)\sigma_\theta(x))(\mu_\theta(x') + \varepsilon(x')\sigma_\theta(x'))\right] \\
&\quad - \mathbb{E}\left[\mu_\theta(x)\right]\mathbb{E}\left[\mu_\theta(x')\right] && (10)\\
&= \mathbb{E}\left[(\mu_\theta(x)\mu_\theta(x') + \mu_\theta(x)\varepsilon(x')\sigma_\theta(x') + \mu_\theta(x')\varepsilon(x)\sigma_\theta(x) + \varepsilon(x)\varepsilon(x')\sigma_\theta(x)\sigma_\theta(x')\right] \\
&\quad - \mathbb{E}\left[\mu_\theta(x)\right]\mathbb{E}\left[\mu_\theta(x')\right] && (11)\\
&= \mathbb{E}\left[(\mu_\theta(x)\mu_\theta(x') + \cancelto{\mathbb{E}(\varepsilon)=0}{\mu_\theta(x)\varepsilon(x')\sigma_\theta(x')} + \cancelto{\mathbb{E}(\varepsilon)=0}{\mu_\theta(x')\varepsilon(x)\sigma_\theta(x)} + \varepsilon(x)\varepsilon(x')\sigma_\theta(x)\sigma_\theta(x')\right] \\
&\quad - \mathbb{E}\left[\mu_\theta(x)\right]\mathbb{E}\left[\mu_\theta(x')\right] && (12)\\
&= R_{\varepsilon\varepsilon}(x, x')\mathbb{E}\left[\sigma_\theta(x)\sigma_\theta(x')\right] + \mathrm{Cov}(\mu_\theta(x), \mu_\theta(x')) && (13)
\end{aligned}$$

$$\begin{aligned}
\mathrm{Var}(E(x)) &= \lim_{x'\to x} \mathrm{Cov}(E(x), E(x')) && (14)\\
&= R_{\varepsilon\varepsilon}(x, x)\mathbb{E}\left[\sigma_\theta^2(x)\right] + \mathrm{Var}(\mu_\theta(x))\,. && (15)
\end{aligned}$$

To do the covariance of the force we use that:

$$\mathrm{Cov}\left(\frac{\partial E(x)}{\partial x}, \frac{\partial E(x')}{\partial x'}\right) = \frac{\partial^2 \mathrm{Cov}(E(x), E(x'))}{\partial x \partial x'}\,. \tag{16}$$

First take the partial derivative with respect to x:

$$\begin{aligned}
\mathrm{Cov}\left(\frac{\partial E(x)}{\partial x}, E(x')\right) &= \frac{\partial \mathrm{Cov}(E(x), E(x'))}{\partial x} && (17)\\
&= \frac{\partial R_{\varepsilon\varepsilon}(x, x')\mathbb{E}\left[\sigma_\theta(x)\sigma_\theta(x')\right]}{\partial x} + \mathrm{Cov}\left(\frac{\partial \mu_\theta(x)}{\partial x}, \mu_\theta(x')\right) && (18)\\
&= \frac{\partial R_{\varepsilon\varepsilon}(x, x')}{\partial x}\mathbb{E}\left[\sigma_\theta(x)\sigma_\theta(x')\right] && (19)\\
&\quad + R_{\varepsilon\varepsilon}(x, x')\mathbb{E}\left[\frac{\partial \sigma_\theta(x)}{\partial x}\sigma_\theta(x')\right] && (20)\\
&\quad + \mathrm{Cov}\left(\frac{\partial \mu_\theta(x)}{\partial x}, \mu_\theta(x')\right)\,. && (21)
\end{aligned}$$

We have used the product rule for differentiation $f(x) = u(x)v(x) \Rightarrow f'(x) = u(x)v'(x) + u'(x)v(x)$. Now also take the partial derivative with respect to x':

$$\begin{aligned}
\mathrm{Cov}\left(\frac{\partial E(x)}{\partial x}, \frac{\partial E(x')}{\partial x'}\right) &= \frac{\partial^2 \mathrm{Cov}(E(x), E(x'))}{\partial x \partial x'} && (22)\\
&= \frac{\partial^2 R_{\varepsilon\varepsilon}(x, x')}{\partial x \partial x'}\mathbb{E}\left[\sigma_\theta(x)\sigma_\theta(x')\right] \\
&\quad + \frac{\partial R_{\varepsilon\varepsilon}(x, x')}{\partial x}\mathbb{E}\left[\frac{\sigma_\theta(x)\partial\sigma_\theta(x')}{\partial x'}\right] \\
&\quad + \frac{\partial R_{\varepsilon\varepsilon}(x, x')}{\partial x'}\mathbb{E}\left[\frac{\partial\sigma_\theta(x)\sigma_\theta(x')}{\partial x}\right] \\
&\quad + R_{\varepsilon\varepsilon}(x, x')\mathbb{E}\left[\frac{\partial\sigma_\theta(x)\partial\sigma_\theta(x')}{\partial x\partial x'}\right] \\
&\quad + \mathrm{Cov}\left(\frac{\partial \mu_\theta(x)}{\partial x}, \frac{\partial \mu_\theta(x')}{\partial x'}\right)\,. && (23)
\end{aligned}$$

To compute the variance we let $x' \to x$ and evaluate the expression in equation 23.

$$\begin{aligned}
\mathrm{Var}\left(\frac{\partial E(x)}{\partial x}\right) &= \frac{\partial^2 \mathrm{Cov}(E(x), E(x'))}{\partial x \partial x'} && (24)\\
&= \frac{\partial^2 R_{\varepsilon\varepsilon}(x,x')}{\partial x \partial x'}\Big|_{x'=x} \mathbb{E}\left[\sigma_\theta(x)^2\right] \\
&\quad + 2\frac{\partial R_{\varepsilon\varepsilon}(x,x')}{\partial x'}\Big|_{x'=x} \mathbb{E}\left[\frac{\sigma_\theta(x)\partial\sigma_\theta(x')}{\partial x'}\Big|_{x'=x}\right] \\
&\quad + R_{\varepsilon\varepsilon}(x,x)\mathbb{E}\left[\left(\frac{\partial\sigma_\theta(x)}{\partial x}\right)^2\right] \\
&\quad + \mathrm{Var}\left(\frac{\partial\mu_\theta(x)}{\partial x}\right). && (25)
\end{aligned}$$

If we further assume that the noise process $\varepsilon(x)$ is wide sense stationary, i.e. its autocorrelation function function is a function of the difference between the inputs $R_{\varepsilon\varepsilon}(x,x') = R_{\varepsilon\varepsilon}(x-x')$, we can simplify the expression further. Since $R_{\varepsilon\varepsilon}(x-x')$ has a maximum at 0 it's first partial derivatives will be 0 and for an even function $f(x)$ we have that:

$$\frac{\partial^2 f(x-x')}{\partial x \partial x'}\Big|_{x'=x} = -\frac{\partial^2 f(x)}{\partial x^2}\Big|_{x=0} \tag{26}$$

Under the wide sense stationary noise assumption the variance of the gradient can therefore be simplified as:

$$\mathrm{Var}\left(\frac{\partial E(x)}{\partial x}\right) = -\frac{\partial^2 R_{\varepsilon\varepsilon}(x)}{\partial x^2}\Big|_{x=0}\mathbb{E}\left[\sigma_\theta^2(x)\right] + R_{\varepsilon\varepsilon}(0)\mathbb{E}\left[\left(\frac{\partial\sigma_\theta(x)}{\partial x}\right)^2\right] + \mathrm{Var}\left(\frac{\partial\mu_\theta(x)}{\partial x}\right). \tag{27}$$

A.1 First principles proof for the variance of the gradient

Again look at equation 9. It can be shown that:

$$\mathrm{Var}\left(\frac{\partial E(x)}{\partial x}\right) = \mathrm{Var}\left(\frac{\partial\mu_\theta(x)}{\partial x}\right) + \mathrm{Var}\left(\frac{\partial(\varepsilon(x)\sigma_\theta(x))}{\partial x}\right), \tag{28}$$

by showing that the covariance between the first and second term is 0. The first term (variance of the gradient of the mean) is straightforward to compute. Here we will focus on the second term. The average of the gradient is 0, so the variance is equal to the second moment:

$$\begin{aligned}
\mathrm{Var}\left(\frac{\partial(\varepsilon(x)\sigma_\theta(x))}{\partial x}\right) &= \mathbb{E}\left[\left(\frac{\partial(\varepsilon(x)\sigma_\theta(x))}{\partial x}\right)^2\right] && (29)\\
&= \mathbb{E}\left[\lim_{\Delta x\to 0}\left(\frac{\sigma_\theta(x+\Delta x)\varepsilon(x+\Delta x) - \sigma_\theta(x)\varepsilon(x)}{\Delta x}\right)^2\right] && (30)\\
&= \mathbb{E}\left[\lim_{\Delta x\to 0}\frac{\sigma_\theta^2(x+\Delta x)\varepsilon^2(x+\Delta x) + \sigma_\theta^2(x)\varepsilon^2(x) - 2\sigma_\theta(x+\Delta x)\varepsilon(x+\Delta x)\sigma_\theta(x)\varepsilon(x)}{\Delta x^2}\right]. && (31)
\end{aligned}$$

Interchange the order of expectation and limit:

$$= \lim_{\Delta x\to 0}\frac{\left\langle\sigma_\theta^2(x+\Delta x)\right\rangle R_{\varepsilon\varepsilon}(0) + \left\langle\sigma_\theta^2(x)\right\rangle R_{\varepsilon\varepsilon}(0) - 2\left\langle\sigma_\theta(x+\Delta x)\sigma_\theta(x)\right\rangle R_{\varepsilon\varepsilon}(\Delta x)}{\Delta x^2}, \tag{32}$$

where $\langle\cdot\rangle = \mathbb{E}_{p(\theta)}[\cdot]$ and we have used wide sense stationary property of $\varepsilon(x)$, thus $\mathbb{E}\left[\varepsilon^2(x)\right] = R_{\varepsilon\varepsilon}(0)\forall x \in \mathbb{R}$. We now add $-2\left\langle\sigma_\theta(x+\Delta x)\right\rangle\sigma_\theta(x)R_{\varepsilon\varepsilon}(0) + 2\left\langle\sigma_\theta(x+\Delta x)\right\rangle\sigma_\theta(x)R_{\varepsilon\varepsilon}(0) = 0$ to the numerator to complete the square.

$$= \lim_{\Delta x \to 0} \frac{\left\langle (\sigma_\theta(x+\Delta x) - \sigma_\theta(x))^2 \right\rangle R_{\varepsilon\varepsilon}(0) + 2\left\langle \sigma_\theta(x+\Delta x)\sigma_\theta(x)\right\rangle (R_{\varepsilon\varepsilon}(0) - R_{\varepsilon\varepsilon}(\Delta x))}{\Delta x^2} \tag{33}$$

$$= R_{\varepsilon\varepsilon}(0)\mathbb{E}_{p(\theta)}\left[\lim_{\Delta x \to 0}\left(\frac{\sigma_\theta(x+\Delta x) - \sigma_\theta(x)}{\Delta x}\right)^2\right] + \lim_{\Delta x \to 0} \frac{2\left\langle \sigma_\theta(x+\Delta x)\sigma_\theta(x)\right\rangle (R_{\varepsilon\varepsilon}(0) - R_{\varepsilon\varepsilon}(\Delta x))}{\Delta x^2}\,. \tag{34}$$

The first term is $R_{\varepsilon\varepsilon}(0)\mathbb{E}\left[\left(\frac{\partial \sigma_\theta(x)}{\partial x}\right)^2\right]$. Let us now rearrange the second term:

$$\lim_{\Delta x \to 0} \frac{2\left\langle \sigma_\theta(x+\Delta x)\sigma_\theta(x)\right\rangle (R_{\varepsilon\varepsilon}(0) - R_{\varepsilon\varepsilon}(\Delta x))}{\Delta x^2} \tag{35}$$

$$= \lim_{\Delta x \to 0} -\frac{\left\langle \sigma_\theta(x+\Delta x)\sigma_\theta(x)\right\rangle (2R_{\varepsilon\varepsilon}(\Delta x) - 2R_{\varepsilon\varepsilon}(0))}{\Delta x^2} \tag{36}$$

$$= \lim_{\Delta x \to 0} -\frac{\left\langle \sigma_\theta(x+\Delta x)\sigma_\theta(x)\right\rangle (R_{\varepsilon\varepsilon}(-\Delta x) - 2R_{\varepsilon\varepsilon}(0) + R_{\varepsilon\varepsilon}(\Delta x))}{\Delta x^2} \tag{37}$$

$$= -\lim_{\Delta x \to 0} \left\langle \sigma_\theta(x+\Delta x)\sigma_\theta(x)\right\rangle \lim_{\Delta x \to 0} \frac{R_{\varepsilon\varepsilon}(-\Delta x) - 2R_{\varepsilon\varepsilon}(0) + R_{\varepsilon\varepsilon}(\Delta x)}{\Delta x^2} \tag{38}$$

$$= -\mathbb{E}\left[\sigma_\theta^2(x)\right] \lim_{\Delta x \to 0} \frac{R_{\varepsilon\varepsilon}(-\Delta x) - 2R_{\varepsilon\varepsilon}(0) + R_{\varepsilon\varepsilon}(\Delta x)}{\Delta x^2} \tag{39}$$

$$= -\mathbb{E}\left[\sigma_\theta^2(x)\right] \frac{\partial^2 R_{\varepsilon\varepsilon}(x)}{\partial x^2}\Big|_{x=0}\,. \tag{40}$$

Here we have used that the correlation function is even $R_{\varepsilon\varepsilon}(x) = R_{\varepsilon\varepsilon}(-x)$ and recognized the second symmetric derivative. These expressions are the same as found in equation 27.

B Epistemic Uncertainty

Using a Bayesian interpretation of deep ensemble models [16, 6, 5] we can interpret the model weights $\boldsymbol{\theta}^{(m)}$ of each ensemble member, m, as samples from an approximate posterior distribution $q(\boldsymbol{\theta}) \approx p(\boldsymbol{\theta}|\mathcal{D})$, where $\mathcal{D}$ is the training set. For a regression model with input x^* and output y^* trained on $\mathcal{D}$ we have:

$$p(y^*|x^*, \mathcal{D}) = \int p(y^*|x^*, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta}, \tag{41}$$

$$\approx \frac{1}{M}\sum_{m=1}^{M} p(y^*|x^*, \boldsymbol{\theta}), \quad \boldsymbol{\theta}^{(m)} \sim p(\boldsymbol{\theta}|\mathcal{D})\,, \tag{42}$$

$$\approx \frac{1}{M}\sum_{m=1}^{M} p(y^*|x^*, \boldsymbol{\theta}), \quad \boldsymbol{\theta}^{(m)} \sim q(\boldsymbol{\theta})\,. \tag{43}$$

The first approximation is to estimate the integral with M samples from the distribution $p(\boldsymbol{\theta}|\mathcal{D})$ and the second approximation comes from approximating the true posterior $p(\boldsymbol{\theta}|\mathcal{D})$ with the distribution $q(\boldsymbol{\theta})$. The uncertainty arising from $p(y^*|x^*, \boldsymbol{\theta})$ is the aleatoric uncertainty while the epistemic uncertainty is modeled as the uncertainty arising from the distribution of the model parameters $q(\boldsymbol{\theta})$. When we train an ensemble of models using the colored noise model presented in section 3 as the base model, we get the following expressions for the energy prediction mean and variance:

$$\mathbb{E}_{\boldsymbol{\theta}}\left[E_{\text{obs}}(\boldsymbol{z}, \boldsymbol{r})\right] = \mathbb{E}_{\boldsymbol{\theta}}\left[E_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{r})\right], \tag{44}$$

$$\text{Var}_{\boldsymbol{\theta}}\left(E_{\text{obs}}(\boldsymbol{z}, \boldsymbol{r})\right) = \underbrace{\mathbb{E}_{\boldsymbol{\theta}}\left[\rho^2_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{r})\right]}_{\text{aleatoric}} + \underbrace{\text{Var}_{\boldsymbol{\theta}}\left(E_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{r})\right)}_{\text{epistemic}}. \tag{45}$$

The expression for the variance can be derived by applying the rule of total variance $\text{Var}(Y) = \mathbb{E}\left[\text{Var}(Y|X)\right] + \text{Var}(\mathbb{E}[Y|X])$. Similarly for the force variance we get:

$$\mathrm{Var}\left(-\frac{\partial E_{\mathrm{obs}}}{\partial r_{i,d}}(\boldsymbol{z},\boldsymbol{r})\right) = \underbrace{\mathbb{E}_{\boldsymbol{\theta}}\left[\hat{\gamma}\rho^2_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{r}) + \left(\frac{\partial \rho_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{r})}{\partial r_{i,d}}\right)^2\right]}_{\text{aleatoric}} + \underbrace{\mathrm{Var}_{\boldsymbol{\theta}}\left(\frac{\partial E_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{r})}{\partial r_{i,d}}\right)}_{\text{epistemic}}. \tag{46}$$

C Datasets

The ANI-1x dataset [15] consists of Density Functional Theory (DFT) calculations for approximately 5 million diverse molecular conformations with an average of 8 heavy atoms (C, N, O) and an average of 15 total atoms (including H) computed at the ωB97x/6-31G(d) level of theory and the structures were generated by pertubing equilibrium configurations and using an active learning procedure to ensure diversity. The Transition1x dataset [11] contains DFT calculations of energy and forces for 9.6 million molecular conformations with up to 7 heavy atoms (C, N, O) and an average of 14 total atoms (including H), likewise computed at the ωB97x/6-31G(d) level of theory and the structures were generated by running a nudged elastic band (NEB) algorithm to find transition states between two equilibrium configurations. Intermediate images of the NEB algorithm were also included in the dataset, with the aim to improve ML potentials around transition states. The ANI-1x and the Transition1x datasets are split into training, test and validation using the splits from Schreiner et al. [11]. For Transition1x the training, validation and test splits contain 9091788, 269636 and 283316 molecules respectively. For ANI-x the training, validation and test splits contain 4449806, 244331 and 261868 molecules respectively.

D Model Hyperparameters and Training

The models are trained using the Adam optimizer [7] with an initial learning rate of 10^{-4} and a batch size of 64 molecules. The learning rate decays with the factor $0.96^{s/10^5}$ where s is the number of gradient steps. Simultaneous training of mean and variance networks with negative log-likelihood cost function is an ill-posed problem and requires some tricks to work reliably [14, 13]. When training a model with variance network, we use the following procedure. The first 2 million gradient update steps we train only the mean function with mean squared error cost function. For the next 1 million gradient update steps we include training of the variance using the β-negative-log-likelihood cost function [13]. During this phase the β parameter is decreased linearly from 1 (the gradient of the mean error corresponds to squared-error) to 0 (the cost function is identical to negative log-likelihood), which gives a smooth transition from mean squared error to negative log-likelihood and increases the stability of the training. Finally we train the model for another 3-4 million gradient steps with only the negative log-likelihood cost function. We use the validation set for model selection using an early stopping procedure and the model selection is always performed using the negative log-likelihood cost function, except for the vanilla model that uses the mean squared error cost function for all training and validation.

Training a single model takes up to 7 days on a single NVIDIA RTX 3090 GPU. Individual members of the ensemble models are trained in parallel using the same procedure as the single models but with different random seeds for the initialization of the weight matrices and different seeds for the minibatch sampling of the stochastic gradient descent to induce model diversity.

For the PaiNN [12] base model we use 3 layers of message passing and a hidden node size of 256. The cutoff distance for creating the message passing graph is 5 Ångström.

E Reliability Diagrams and Uncertainty Metrics

There are various techniques available for assessing the calibration of regression models [2]. The negative log-likelihood (NLL) can be used as a standard metric for evaluating the overall performance of probabilistic models. In that case it measures the (negative) likelihood of observing the hold out test data based on the predicted distribution. The NLL loss for energy, assuming a normally distributed error, is given for a single instance by the following expression where $x = \{(Z_i, \vec{r_i})\}$ represents the model input and the observed values of energy and forces are denoted by E^{obs} and

F^{obs}, respectively:

$$\text{NLL}_E(\theta) = -\log p(E^{\text{obs}}|x,\theta) \tag{47}$$

$$= \frac{1}{2}\left(\frac{\left(E^{\text{obs}} - E(x)\right)^2}{\sigma_E^2(x)} + \log\sigma_E^2(x) + \log 2\pi\right). \tag{48}$$

The instance-wise energy losses are then averaged over the number of instances.

Analogous to the MSE loss for forces, the NLL loss for forces is evaluated per atom i and component-wise over the spatial dimensions D (recall that the predicted atom-wise force uncertainty $\sigma_{F_i}^2$ is a single scalar applied over all spatial dimensions):

$$\text{NLL}_{F_i}(\theta) = \sum_{d=1}^{D} -\log p(F_{i,d}^{\text{obs}}|x,\theta) \tag{49}$$

$$= \sum_{d=1}^{D} \frac{1}{2}\left(\frac{\left(F_{i,d}^{\text{obs}} - F_{i,d}(x)\right)^2}{\sigma_{F_i}^2(x)} + \log\sigma_{F_i}^2(x) + \log 2\pi\right). \tag{50}$$

When using an ensemble of models we use the predicted total mean and total variance to parameterise a normal distribution [1] in order to obtain NLL scores of the test data.

As shown in equation 48 and equation 50, the NLL is dependent on both the predicted mean and uncertainty. However, in some cases, it is more informative to assess the quality of the uncertainty estimates separately. For instance, a common practice is to visually compare the predicted uncertainties with empirical errors through plotting. Recently Pernot [10] proposed analysis of the z-scores (standard scores). The z-scores are defined as the empirical error divided by the standard deviation of the predictive distribution [10]:

$$z = \frac{y^{\text{obs}} - y(x)}{\sigma(x)}. \tag{51}$$

To get a single number summarizing the calibration, we can compute the z-score variance (ZV) [10]: A z-score variance (ZV) close to 1 indicates that the predicted uncertainty, on average, corresponds well to the variance of the error, thus indicating good overall calibration. It is often useful to plot the histogram of z-scores, but to summarise the calibration we give the square root of the z-variance (RZV), i.e. the standard deviation of the z-scores.

Additionally, we can assess how well the uncertainty estimates correspond to the expected error locally by sorting the predictions in equal size bins by increasing uncertainty and plotting the root mean variance (RMV) of the uncertainty versus the empirical root mean squared error (RMSE), also known as an error-calibration plot or reliability diagram. Reliability diagrams of our experiments are shown for ANI-1X in Figure 2 and Transition1x in Figure 3 The error-calibration can be summarized by the expected normalized calibration error (ENCE) [9], which measures the mean difference between RMV and RMSE normalised by RMV:

$$\text{ENCE} = \frac{1}{K}\sum_{k=1}^{K}\frac{|\text{RMV}_k - \text{RMSE}_k|}{\text{RMV}_k}, \tag{52}$$

where $k = 1, \ldots, K$ iterates the bins.

Achieving good average or local calibration is not enough to ensure that individual uncertainty estimates are informative. If the uncertainty estimates are homoscedastic (lack variation), they may not be very useful. Therefore, it is generally desirable for uncertainty estimates to be as small as possible while still displaying some variation, which is referred to as sharpness. To measure sharpness, two metrics can be used: the root mean variance (RMV) of the uncertainty, which should be small and correspond to the RMSE, and the coefficient of variation (CV), which quantifies the ratio of the standard deviation of the uncertainties to the mean uncertainty providing a simple metric for the heteroscedasticity of the predicted uncertainty. The equation for CV is as follows:

$$\text{CV} = \frac{\sqrt{N^{-1}\sum_{n=1}^{N}(\sigma(x_n) - \overline{\sigma})^2}}{\overline{\sigma}}, \tag{53}$$

where $n = 1, \ldots, N$ in this case iterates the test dataset, $\sigma(x_n)$ is the predicted standard deviation (uncertainty) of instance n and $\overline{\sigma} = N^{-1}\sum_{n=1}^{N}\sigma(x_n)$ is the mean predicted standard deviation.

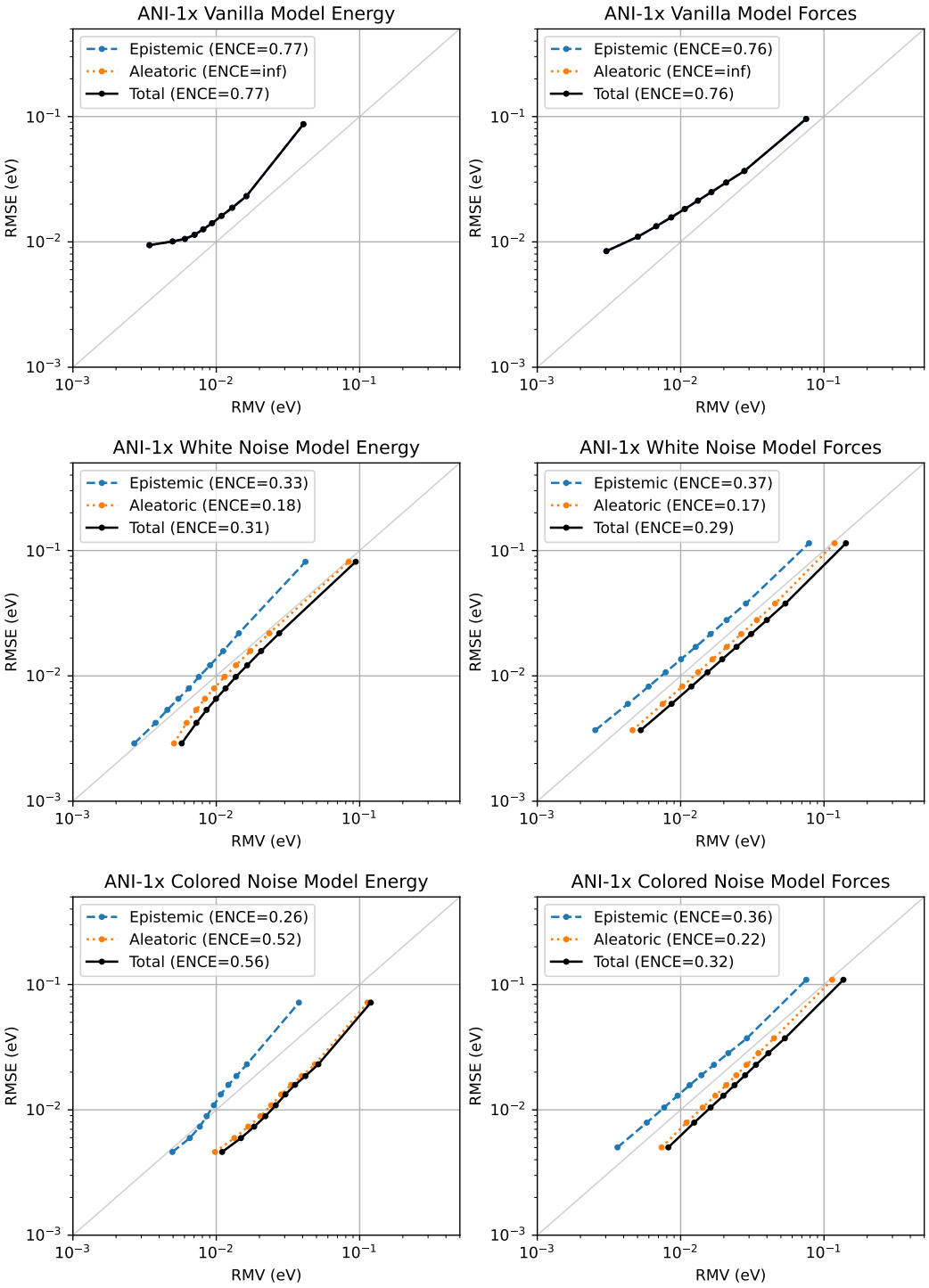

Figure 2: Reliability diagrams for models trained on the ANI-1x dataset.

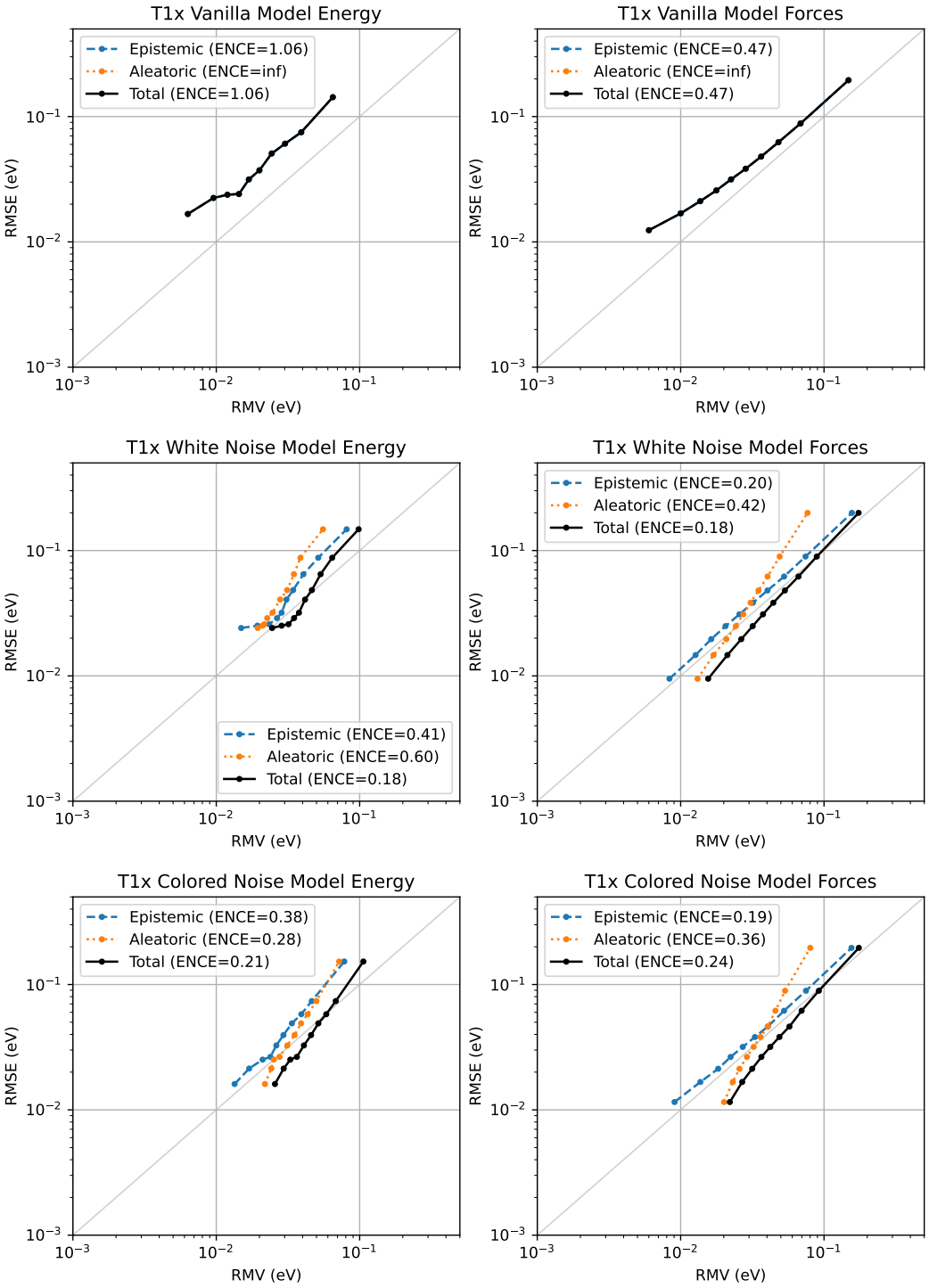

Figure 3: Reliability diagrams for models trained on the Transition1x dataset.