Uncertainty Estimation for Molecules: Desiderata and Methods

Tom Wollschläger [1]   **Nicholas Gao** [1]   **Bertrand Charpentier** [1]   **Mohamed Amine Ketata** [1]   **Stephan Günnemann** [1]

Abstract

Graph Neural Networks (GNNs) are promising surrogates for quantum mechanical calculations as they establish unprecedented low errors on collections of molecular dynamics (MD) trajectories. Thanks to their fast inference times they promise to accelerate computational chemistry applications. Unfortunately, despite low in-distribution (ID) errors, such GNNs might be horribly wrong for out-of-distribution (OOD) samples. Uncertainty estimation (UE) may aid in such situations by communicating the model's certainty about its prediction. Here, we take a closer look at the problem and identify six key desiderata for UE in molecular force fields, three 'physics-informed' and three 'application-focused' ones. To overview the field, we survey existing methods from the field of UE and analyze how they fit to the set desiderata. By our analysis, we conclude that none of the previous works satisfies all criteria. To fill this gap, we propose Localized Neural Kernel (LNK) a Gaussian Process (GP)-based extension to existing GNNs satisfying the desiderata. In our extensive experimental evaluation, we test four different UE with three different backbones and two datasets. In out-of-equilibrium detection, we find LNK yielding up to 2.5 and 2.1 times lower errors in terms of AUC-ROC score than dropout or evidential regression-based methods while maintaining high predictive performance.

1. Introduction

In recent years, access to molecular forces has become an essential aspect in various applications such as geometry optimization, and molecular dynamics (MD) simulations (Jensen, 2010). However, the underlying quantum mechanical (QM) calculations required for these predictions are computationally demanding. In order to reduce this computational burden, graph neural networks (GNNs) fitted to QM data have been proposed as a means to accelerate MD simulations (Chmiela et al., 2017; Schütt et al., 2018). While such surrogates have recently achieved exceptional reproduction of the dataset they have been trained on, they tend to perform poorly on out-of-distribution (OOD) data (Li et al., 2022). In practical applications, having access to the full-dimensional potential energy surface of molecules is rare. For example, in MD simulations, one typically starts with an initial structure and iteratively applies molecular forces (Hoja et al., 2021). These simulations may cause the structure to step out of the training domain, rendering the network's predictions unreliable (Stocker et al., 2022). Uncertainty estimation (UE) is a promising direction for detecting such unforeseen events.

In the context of molecular force fields, UE comes with a unique and more stringent set of requirements compared to other fields. We categorize these requirements into 'physical-informed' and 'application-focused' desiderata. In a survey, we then analyze existing works in UE for molecular force fields on these desiderata. Our analysis reveals that none of the previous work completely satisfies all desiderata. To fill the gap, we present Localized Neural Kernel (LNK), a Gaussian Process (GP)-based extension to existing GNN-based force fields that reliably estimates uncertainty with a single forward pass while not harming the predictive performance fulfilling all desiderata. When testing out-of-equilibrium detection, we find LNK yielding up to 2.5 and 2.1 times lower errors in terms of AUC-ROC score than dropout-based or evidential-based UE (Gal & Ghahramani, 2016; Soleimany et al., 2021).

To summarize, our contributions are:[1]

- We derive *physics-informed* and *application-focused* desiderata for uncertainty-aware molecular force fields.
- We survey previous UE methods based on our desiderata and conclude that existing methods fail on at least one of our desiderata.
- We present Localized Neural Kernel (LNK), a GP-based extension to existing GNN-based force fields satisfying all desiderata.

---

[1]Department of Computer Science & Munich Data Science Institute, Technical University of Munich, Germany. Correspondence to: Tom Wollschläger <tom.wollschlaeger@tum.de>.

---

[1]Find our code at cs.cit.tum.de/daml/uncertainty-for-molecules

2. Uncertainty Estimation Criteria for Molecular Predictions

We consider the task of energy and force predictions on molecules. One molecule is represented as a point cloud of n points (atoms), each associated with a position and a set of rotationally invariant features (e.g., atom types), defined as $\boldsymbol{X} \in \mathbb{R}^{n \times 3}$ and $\boldsymbol{H} \in \mathbb{R}^{n \times h}$, respectively. In addition to learning the molecular energy $E \in \mathbb{R}$ and the atom forces $\boldsymbol{F} \in \mathbb{R}^{n \times 3}$, we investigate the ability of our approach, LNK, to quantify energy uncertainty u_E and demonstrate that it can further be used in a variation-based way to quantify force uncertainty $u_{\boldsymbol{F}}$. In particular, we aim at learning a function f_θ such that $f_\theta(\boldsymbol{X}, \boldsymbol{H}) \approx E^{\text{gt}}$ and $\frac{\partial f_\theta}{\partial \boldsymbol{X}}(\boldsymbol{X}, \boldsymbol{H}) \approx \boldsymbol{F}^{\text{gt}}$ where E^{gt} and $\boldsymbol{F}^{\text{gt}}$ denote the ground-truth molecular energy and forces, respectively. The predicted values are denoted with star subscript, i.e., E_* and $\boldsymbol{F}_*$. We denote the training set consisting of all molecule positions and features as $\mathcal{M} = \{(\boldsymbol{X}_1, \boldsymbol{H}_1), \ldots, (\boldsymbol{X}_N, \boldsymbol{H}_N)\}$. Stacking the tuples of the dataset into a matrix, we refer to it as $\boldsymbol{M}$ with corresponding target vector of energies $\mathbf{y}$.

2.1. Desiderata

Dealing with energy and forces in molecular structures, which are influenced by physical symmetries, poses unique challenges for UE in the context of molecular force fields. To effectively perform UE in this area, it is essential to take into account both *physics-informed* and *application-focused* desiderata. Physics-informed desiderata capture the physical constraints and behavior characteristic of molecular systems. In this work, we identify three main physics-informed desiderata:

- *Symmetry*: Symmetries play an essential role in physics. They describe the fundamental behavior of quantities such as energies or forces under euclidean transformations. For instance, the energy of a molecule E is invariant to the euclidean group $\delta \in \mathcal{E}(3)$ and permutation group $\pi \in S_n$, i.e., $E(\delta \circ \pi \circ \boldsymbol{X}, \pi \circ \boldsymbol{H}) = E(\boldsymbol{X}, \boldsymbol{H})$, while the associated atomic forces behave equivariantly, $\boldsymbol{F}(\delta \circ \pi \circ \boldsymbol{X}, \pi \circ \boldsymbol{H}) = \delta \circ \pi \circ \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{H})$.
- *Energy conservation*: Molecular forces, being the gradient of an energy surface, must form a conservative vector field, i.e., all paths from point a to b have the same integral and the force field must be curl-free. One may obtain all these properties by defining the forces via differentiation of a scalar field $\boldsymbol{F} := \frac{\partial E}{\partial \boldsymbol{X}}$.
- *Locality*: As most of the interaction between atoms happens within a short distance, locality plays an important role in molecular force fields (Leach, 2001). While there are long-range interactions in molecules, these only account for a small fraction of the total energy and are negligible for molecular forces (Lan et al., 2023). As a consequence of locality, a molecule's en-

ergy behaves size-consistently with increased system sizes. Formally, $E \approx \sum_{i=1}^n E_i$, where E is the total energy of n molecules and E_i is the energy of the ith molecule. Similarly, the energy's uncertainty should behave additive under extension.

With MD simulations as an application in mind, we identify three key application-focused desiderata:

- *Accuracy*: MD simulations require high precision and are often performed for millions of steps where each step simulates femto seconds (10^{-15}s) (Stocker et al., 2022). To keep simulations stable and not accumulate errors, high predictive performance and, thus, accurate reproduction of the QM calculations is essential.
- *Confidence-aware*: During an MD simulation, a structure may move outside of the training domain. In such cases, the surrogate must effectively communicate its uncertainty regarding its predictions. If this aspect is not accounted for, divergence has been observed (Stocker et al., 2022).
- *Speed*: Lastly, as MD simulations typically involve millions of steps (Narumi et al., 1999), it is important to preserve the efficient runtime of surrogate methods.

As we discuss in Section 3, current UE approaches for molecules lack at least one of these desiderata. We address this by introducing LNK, a GP-based extension to existing GNNs that satisfies all desiderata.

3. Survey of UE for molecular force fields

While there is a variety of UE methods for deep learning[2], here we only discuss (twice) differentiable solutions that fit the application of predicting molecular energies and forces. UE methods can be broadly categorized into two families of methods: *sampling-based* and *sampling-free* methods. For each method, we outline its advantages and disadvantages and analyze it based on our desiderata set in Section 2.1. An overview of the methods can be found in Table 1.

Ensembles are touted as the 'gold-standard' solution when it comes to UE. An ensemble is a collection of different methods trained on the same dataset (Ovadia et al., 2019). By varying between architecture or initialization between the individual models, one obtains several estimates about the desired property. By computing statistical metrics of these samples, e.g., their standard deviation, one obtains an uncertainty quantification (Lakshminarayanan et al., 2017). In mathematical notation this can be expressed as:

$$u_E = \sqrt{\text{Var}\left(f_1(\boldsymbol{M}), \ldots, f_N(\boldsymbol{M})\right)}$$
$$u_{\boldsymbol{F}} = \text{Tr}\left[\text{Cov}\left(\frac{\partial f_1(\boldsymbol{M})}{\partial \boldsymbol{X}}, \ldots \frac{\partial f_N(\boldsymbol{M})}{\partial \boldsymbol{X}}\right)\right]. \quad (1)$$

---

[2]we refer the interested reader to Gawlikowski et al. (2021)

	Ensemble	MC-Dropout	Evidential	GP	SVGP	SVGP-DKL	LNK
Symmetry	✓	✓	✓	✓	✗	✓	✓
Energy conservation	✓	✓	✓	✓	✗	✓	✓
Locality	✓	∼	✗	✗	✗	✗	✓
Accuracy	✓	∼	✓	✗	✗	∼	✓
Confidence-aware	✓	✓	∼	✓	✓	✓	✓
Speed	✗	✗	✓	✗	✓	✓	✓

Table 1. Desiderata fulfillment of different uncertainty estimation methods.

Here $u_E, u_{\boldsymbol{F}}$ are the measures of the predictive uncertainty for the energy and forces. f_i denotes the i-th component of the ensemble, i.e., the i-th trained network. We use the trace as a metric to capture the uncertainty of the forces covariance matrix in a single number. In terms of our desiderata, ensembles generally preserve the properties of the underlying methods. Thus, they satisfy the same desiderata as their components with *Speed* being a notable exception. As one must train several models *and* perform inference multiple times, an ensemble's runtime scales linearly with its size. In case one uses ensembles of GNNs (Gasteiger et al., 2019) one directly satisfies the physics-informed. As we show in Section 6, ensembles also empirically perform well on *Accuracy* and *Confidence-aware* but crucially fail at *Speed*.

Monte Carlo Dropout (MCD) is proposed as a comparatively efficient alternative to ensembles with the idea of merging the ensemble into a single model with probabilistic output. Instead of having different models as in ensembles, we leverage the dropout during inference and get multiple predictions for the same molecule. Hence, the formulation for $u_E, u_{\boldsymbol{F}}$ is the same as in Equation 1 while the only difference is that f_i denotes a single evaluation of the same model but with different dropped connections (Gal & Ghahramani, 2016). As one simply drops neurons within one model, only one model has to be trained. Like ensembles, MCD generally inherits the desiderata from its base model but only in expectation as it randomly drops connections. As we show in Section 6, MCD deteriorates the predictive performance. Further, we observe that while in expectation locality preserving, MCD's uncertainty is not behaving size-consistent for practical sample sizes and is thus not fulfilling *Locality*. Lastly, while one only has to train a single model compared to ensembles, one still must perform multiple passes during inference violating the *Speed* desiderata.

Evidential Regression: In evidential regression, one directly parametrizes an evidential distribution over the target rather than the target (Soleimany et al., 2021). In practice, this means that instead of directly estimating a normal distribution on the energy, the model estimates the parameters of an evidential Normal-Inverse-Gamma distribution (Amini

et al., 2020; Charpentier et al., 2022). Given this distribution, one can perform statistical tests to quantify uncertainties. While only requiring a single forward pass, this approach does not necessarily yield trustworthy uncertainty estimates, as we will see in Section 6. Similar to the unreliability of the energy prediction itself, the outputted distribution parameters are similarly subject to unpredictable behavior outside the training regime (Charpentier et al., 2020; Kopetzki et al., 2021). Further, by predicting a single distribution for the entire molecular graph, evidential regression requires global embeddings violating *Locality*.

Gaussian Processes (GPs) are a type of non-parametric models for distributions over functions that are known for their flexibility and expressiveness (Bishop & Nasrabadi, 2006; Jakkala, 2021). They are particularly frequently used in UE. One of the key advantages of using GPs is that they allow for the modeling of complex relationships between variables, while also providing a way to quantify the uncertainty associated with the predictions made by the model. In the following, we adapt the notation from Rasmussen & Williams (2009) to our setting. Given the data $\boldsymbol{M}$, we assume a GP prior: $f(\boldsymbol{M}) \sim \mathcal{GP}(\mu(\boldsymbol{M}), k_\gamma(\boldsymbol{M}, \boldsymbol{M}^{\mathrm{T}}))$, where $\mu(\cdot)$ is the mean function and $k_\gamma(\cdot, \cdot)$ is the covariance function with parameters γ. Any collection of points $\mathbf{f}$ lying on $f(\boldsymbol{M})$ follows a normal distribution as:

$$p(\mathbf{f}) = p(f(\boldsymbol{M}))$$
$$= p([f(\boldsymbol{M}_1), \ldots, f(\boldsymbol{M}_n)]^{\top}) \sim \mathcal{N}(\mathbf{f} \,|\, \boldsymbol{\mu}, \boldsymbol{K}_{MM}), \tag{2}$$

where $\boldsymbol{\mu}_i = \mu(\boldsymbol{M}_i)$ is the mean vector and $\boldsymbol{K}_{ij}^{M,M} = k_\gamma(\boldsymbol{M}_i, \boldsymbol{M}_j)$ the Gram matrix. Assuming that our labels are given as noisy observations of the latent variable, we model the prediction as: $\mathbf{y}_i = f(\boldsymbol{M}_i) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma)$. We can then predict the values for a new collection of inputs, $\boldsymbol{M}_\star$, via the predictive distribution:

$$p(\mathbf{f}_\star \,|\, \boldsymbol{M}_\star, \boldsymbol{M}, \mathbf{y}, \sigma, \gamma) \sim \mathcal{N}(\mathbf{f}_\star \,|\, \mathbb{E}[\mathbf{f}_\star], \mathrm{Cov}(\mathbf{f}_\star)) \tag{3}$$

$$\mathbb{E}[\mathbf{f}_\star] = \mu_{M_\star} + \boldsymbol{K}_{M_\star M} \left[K_{MM} + \sigma^2 \boldsymbol{I} \right]^{-1} \mathbf{y}, \tag{4}$$

$$\mathrm{Cov}(\mathbf{f}_\star) = \boldsymbol{K}_{M_\star M_\star} - \boldsymbol{K}_{M_\star M} \left[K_{MM} + \sigma^2 \boldsymbol{I} \right]^{-1} \boldsymbol{K}_{MM_\star}. \tag{5}$$
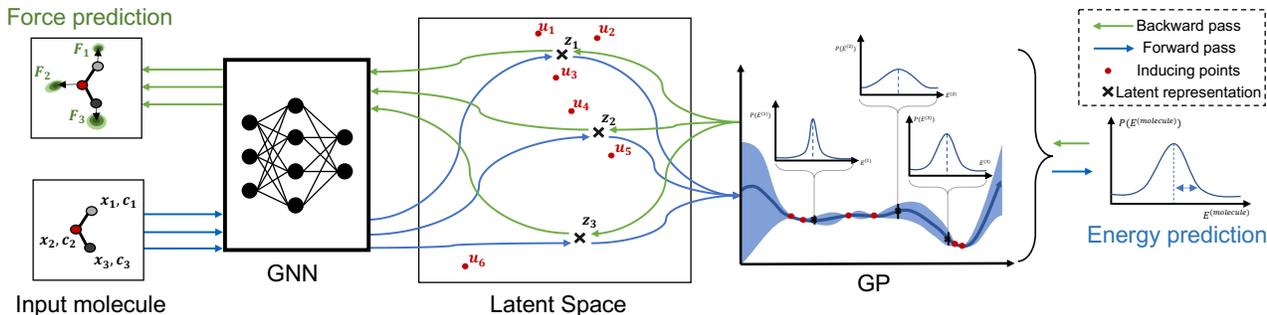
Figure 1. Overview of Localized Neural Kernel

As shown by Vandermause et al. (2020), we can define a GP acting on the input and hand-crafted features to predict the forces of a molecule with an energy-conserving model while being rotationally covariant. However, due to the inversion of the Gram matrix, the exact GP inference scales with $\mathcal{O}(N^3)$ where N is the number of columns or rows of M. Furthermore, compared to recently proposed GNNs the GPs only achieve low accuracies (Gasteiger et al., 2021a; Batzner et al., 2022).

The desiderata *Accuracy* and *Speed* are thus not fulfilled. Further, because a GP compares the graphs as a whole, it does not scale additively with system size, thereby violating *Locality*.

Stochastic Variational Gaussian Process (SVGP): As the exact GP requires the inversion of the covariance matrix, it comes with a significant computational burden. To mitigate this issue, sparse approximations and variational inference have been proposed, known as Sparse and Variational GP (SVGP) (Titsias, 2009; Hensman et al., 2013; 2014).

The SVGP approach entails the selection of a subset of data points, referred to as inducing points $\mathbf{\Phi}$ and variables $\mathbf{u}$ to approximate the underlying distribution through the use of variational inference. A prior can be imposed on the joint distribution:[3]

$$p(\mathbf{f}, \mathbf{u}) \sim \mathcal{N}\left(\left[\begin{array}{c} \mathbf{f} \\ \mathbf{u} \end{array}\right] \Big| \mathbf{0}, \left[\begin{array}{cc} K_{MM} & K_{\Phi M} \\ K_{\Phi M} & K_{\Phi\Phi} \end{array}\right]\right). \quad (6)$$

Note, that this equation describing the relationship between latent variables $\mathbf{f}$ and inducing variables $\mathbf{u}$ is equivalent to the relationship between the training data and the new test points for an exact GP.

To learn the inducing points and variables, we need to include these into the optimization formulation. Titsias (2009) introduced variational inference to approximate the posterior $p(f, u|\mathbf{y})$ by $q(\mathbf{f}, \mathbf{u})$. With the introduction of additional variational parameters, Hensman et al. (2013) extended this approach and enabled stochastic gradient optimization. As-

suming $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u})$ and $q(\mathbf{u})$ being gaussian, we can lower bound the marginal likelihood by the Evidence Lower Bound (ELBO):

$$\begin{aligned} \log &p(\mathbf{f}_\star) \\ &\geq \mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})}\left[\log p(\mathbf{f}_\star|\mathbf{f})\right] - D_{\mathrm{KL}}\left(q(\mathbf{u})||p(\mathbf{u})\right). \end{aligned} \quad (7)$$

This ELBO can be used as loss function for stochastic gradient optimization. The computational complexity of an SVGP is in $\mathcal{O}(m^2 N)$ where m is the number of inducing points. Important to note, is that the inducing points and variables are optimized to best fit the distribution and are not instances of the training data.

Compared to the exact GP, the SVGP fulfills the *Speed* desideratum. However, as the inducing points are free variables to be optimized, their location information has no connection to the atom positions breaking rotational invariance (Vandermause et al., 2020)). Thus, compared to GPs, *Symmetry* is not fulfilled anymore.

4. Localized Neural Kernel

In the following, we introduce LNK in two steps. First, we introduce a non-localized version, termed SVGP-DKL, by combining the SVGP with Deep Kernel Learning (DKL) (Wilson et al., 2016). In DKL one uses deep neural networks to encode the data in a learned, potentially lower-dimensional, space before applying the GP. Both are then trained end-to-end. Second, we discuss how we localize the SVGP-DKL to obtain LNK fulfilling all desiderata.

SVGP-DKL: To leverage the predictive performance of GNN-based molecular force fields, we use such a GNN as encoder for DKL. Further, we avoid the expensive runtimes of the GP by relying on the SVGP approximation. Let $h(\boldsymbol{X}, \boldsymbol{H}) = (h_{\mathrm{pred}} \circ h_{\mathrm{rep}})(\boldsymbol{X}, \boldsymbol{H})$ be a GNN with $h_{\mathrm{rep}} : \mathbb{R}^{n \times 3} \times \mathbb{R}^{n \times h} \to \mathbb{R}^{n \times d}$ being its learned invariant atom representations and $h_{\mathrm{pred}} : \mathbb{R}^{n \times d} \to \mathbb{R}$ its property predictor. By replacing the predictor h_{pred} with an SVGP, we combine the benefits of GNNs with approximate GPs. Note that GPs operate on fixed-dimensional spaces and we,

---

[3]We adapt Hensman et al. (2014)'s notation for consistency.

thus, have to introduce a summation over all atoms to obtain a representation for the whole molecule:

$$p(E_\star|\boldsymbol{X}, \boldsymbol{H}) = \mathcal{GP}_\phi \circ \sum_{i=1}^n h_{\text{rep}}(\boldsymbol{X}, \boldsymbol{H})_i \tag{8}$$
$$\sim \mathcal{N}\left(E_\star | \mathbb{E}[E_\star], \text{Cov}(E_\star)\right)$$

We can view this as a learned kernel (Wilson et al., 2016):

$$k_{\text{SVGP-DKL}}((\boldsymbol{X}^{(k)}, \boldsymbol{H}^{(k)}), (\boldsymbol{X}^{(j)}, \boldsymbol{H}^{(j)})) =$$
$$k\left(\sum_i h_{\text{rep}}(\boldsymbol{X}^{(k)}, \boldsymbol{H}^{(k)})_i, \sum_i h_{\text{rep}}(\boldsymbol{X}^{(j)}, \boldsymbol{H}^{(j)})_i\right) \tag{9}$$

Hence, the prediction of the model is a normal distribution. We can use the variance of the predictive distribution as energy uncertainty estimate. Given that the embeddings of the studied GNNs (SchNet, DimeNet++, NequIP) are invariant to the euclidean group $\mathcal{E}(3)$ and permutations S_n, this GP extension fulfills *Symmetry*.

To fulfill the physical desideratum of *Energy conservation*, we need to calculate the forces as the derivative of the energy. Hence, our method needs to be twice differentiable to be trained. Taking the derivative of the energy with respect to the position thus yields the force prediction:

$$\boldsymbol{F} = -\frac{\partial E}{\partial \boldsymbol{X}} = -\frac{\partial \text{GP}_\phi}{\partial h_{\text{rep}}} \frac{\partial h_{\text{rep}}}{\partial \boldsymbol{X}}. \tag{10}$$

As a GP with RBF kernel is C^∞ smooth, we only have to pick a twice differentiable representation function h_{rep} to fulfill *Energy conservation*. To train the GP, we use the predictive log-likelihood as objective to estimate the energy error (Jankowiak et al., 2020). This objective, similar to the variational ELBO shown in Equation 7, can be formally written as:

$$\mathcal{L} = \log \mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})}[p(\mathbf{f}_\star|\mathbf{f})] - D_{\text{KL}(q(\mathbf{u})||p(\mathbf{u}))}. \tag{11}$$

This formulation has been reported to yield improved estimates of predictive variances when compared to alternative approaches (Jankowiak et al., 2020; Strohbeck et al., 2022). We use the RMSE loss to estimate the force error. However, as the GP operates on the singular, fixed-size, molecule embedding, it fails to satisfy *Locality*.

Localizing SVGP-DKL: To attain *Locality*, we use a localized version of the just defined GNN-based GP to obtain Localized Neural Kernel (LNK). We achieve this by reversing the order of the sum and the GP in Equation 8. Instead of using the global graph embedding, we fit the GP on the atomic embeddings directly and our final output is the sum of these atomic GPs. Hence, the predictive distribution of

the predicted energy value $E_\star$ can be written as[4]:

$$p(E_\star|\boldsymbol{X}, \boldsymbol{H}) = \sum_i^n \mathcal{GP}_\phi \circ h_{\text{rep}}(\boldsymbol{X}, \boldsymbol{H})_i \tag{12}$$
$$\sim \mathcal{N}\left(E_\star | \sum_i^n \mathbb{E}[E_\star]_i, \sum_{ij}^n \text{Cov}(E_\star)_{ij}\right).$$

Since we sum the individual atom energy contributions and uncertainties, we fulfill *Locality*. Compared to the global embedding version, adding an independent molecule will increase the uncertainty proportionally. We will show this comparison also empirically in Section 6. In regards to the 'application-focused' desiderata, LNK only needs a single forward pass fulfilling *Speed*, and as we will show in Section 6 also fulfills *Accuracy* and *Confidence-aware*. Thus, it is the only approach fulfilling all of our set desiderata.

Avoiding pitfalls. As the use of large encoder networks can lead to unstable uncertainty estimation for deep kernel learning models (Ober et al., 2021), proper regularization has to be deployed. Ober et al. suggest using a full-bayesian treatment to avoid this pitfall. We use two different approaches: (1) approximate a full bayesian treatment by using dropout (Gal & Ghahramani, 2016; Srivastava et al., 2014) and (2) a fixed-encoder which is motivated by the strong empirical performance in their work (see Table 3 Ober et al. (2021)).

In order to apply dropout to equivariant networks like NequIP, one cannot simply drop any features. In Section A.2, we discuss how we translate dropout to such group-equivariant neural networks. Note that during inference, we do not perform dropout with LNK as we use the variance of the GP's predictive distribution as an uncertainty estimate rather than an empirical Monte Carlo estimate.

Instead of applying dropout, one might also avoid overfitting by fixing the representation function h_{rep}. This could be done, e.g., by first training the GNN with its original predictor on the objective. After pretraining, the representation function will be frozen similar to common fine-tuning on computer vision models. By limiting the embedding space in such a way we can avoid DKL pitfalls (Ober et al., 2021).

In the following, we refer to the dropout training approach as LNK while we call the fixed encoder approach fixed-LNK.

5. Related work

Machine learning potentials: Machine learning has a long history in learning potentials and force fields for molecular simulations. Starting with Halgren (1996) where empirical force fields were introduced with the goal of fitting func-

---

[4]Assuming independence of the atom random variables and, thus, summing only the diagonal elements leads to better performance.

tions to QM calculations to avoid performing the expensive QM calculations many times. Later, kernel methods (Behler, 2011; Bartók et al., 2013; Christensen et al., 2020) based on sophisticated feature constructions of molecular neighborhoods took over the field. These kernel methods were fast to evaluate and capable of accurately capturing molecular interactions. By focusing on locality and freely learnable featurizations GNNs (Schütt et al., 2018; Unke & Meuwly, 2019) achieved accurate reproductions of molecular force fields. Due to the implementation of physical symmetries to the euclidean group $\mathcal{E}(3)$ and the permutational group S_n, GNNs presented sample efficient ways of learning from molecular data. Recent advances provided universal models, i.e., models that can model any function on geometry point clouds, accomplished by including dihedral angles (Gasteiger et al., 2021b; Liu et al., 2021) or by the introduction of SO(3) equivariant models (Thomas et al., 2018; Batzner et al., 2022). Other developments focus on closing the gap between QM calculations and ML potentials, e.g., by including prior physical knowledge (Unke et al., 2021), by providing efficient QM calculations as additional input (Qiao et al., 2020; 2021), by implementing DFT functionals with GNNs (Snyder et al., 2012; Kirkpatrick et al., 2021), or by learning potentials directly from first principle (Gao & Günnemann, 2022; 2023a;b).

Uncertainty in machine learning. There are different approaches for uncertainty estimation (UE) in deep learning. The interested reader is referred to Gawlikowski et al. (2021) for a detailed survey. UE methods can be broadly categorized into two families of methods: *sampling-based* and *sampling-free* methods.

Sampling-based methods generally evaluate the uncertainty by estimating statistics over multiple different predictions. Generally, when the prediction is very similar of over the different samples, this indicates the low-uncertainty of the model. Ensembles (Kim & Ghahramani, 2012; Lakshminarayanan et al., 2017; Simpson et al., 2012; Wen et al., 2020; Wenzel et al., 2020), often refered to as the gold-standard, are a collection of individual models. They estimate the uncertainty by the variation in the prediction of the different ensemble members. Dropout methods (Gal & Ghahramani, 2016), approximate this technique by following the argument that dropout let the model learn different subnetworks during training. Thus each random initialization might be an approximation of an ensemble member. Other notable techniques are bayesian neural networks (Blundell et al., 2015; Ritter et al., 2018; Maddox et al., 2019), where the model parameters are a distribution instead of a single value. Hence, one can sample parameters of the model multiple times and use these to infer a data sample resulting in multiple predictions from which we can calculate the variance again.

Sampling-free generally requires to evaluate the uncertainty in a single forward-pass methods. Evidential methods propose to parametrize conjugate prior distributions (Ulmer, 2021; Kopetzki et al., 2021; Nandy et al., 2020; Sensoy et al., 2020; Shi et al., 2020; Stadler et al., 2021; Gast & Roth, 2018). Deep Kernel Learning aim at learning Gaussian processes in latent space with random feature projections or learned inducing points (Lakshminarayanan et al., 2020; van Amersfoort et al., 2021; 2020; Biloš et al., 2019). Calibration models are methods which aim at predicting confidence which are good approximations of the true probability of correctness (Kuleshov et al., 2018; Moon et al., 2020; Zhao et al., 2020; Song et al., 2019; Rahimi et al., 2020). Other models propose to propagate uncertainty accross layers and model uncertainty at the weight or activation levels (Wang et al., 2016; Postels et al., 2019; Shekhovtsov & Flach, 2019; Gast & Roth, 2018; Hernandez-Lobato & Adams, 2015). Most of these models do not predict uncertainty on molecular properties. Only Soleimany et al. (2021) leverage uncertainty estimation of evidential deep learning methods for guided molecular property prediction and show its effectiveness. We use their work as a further baseline, to which we compare our results. Another interesting approach is the Graph Mixture Density Network (Errica et al., 2021), which outputs a mixture of distributions on the predicted target but does not focus on molecules.

6. Experiments

In this section, we focus on the empirical evaluation of our desiderata. We compare Ensembles, Monte Carlo dropout (MCD), evidential regression, SVGP-DKL[5], and LNK. We pair each of the UE methods with SchNet (Schütt et al., 2021), DimeNet++(Gasteiger et al., 2019; 2020a) and NequIP (Batzner et al., 2022). As traditional GPs and SVGP either do not allow for varying input sizes or result in poor predictive performance, we do not discuss these here. Further, as *Symmetry* and *Energy conservation* are fixed desiderata, discussed in earlier Section 3 and Section 4, we do not perform experiments for those.

In our first experiment, we analyze *Locality* by testing whether the UE is size consistent. Next, we analyze the *Accuracy* on two common QM datasets, MD17 (Chmiela et al., 2017) and QM7X (Hoja et al., 2021). To measure *Confidence-aware*, we look at the calibration score and out-of-distribution (OOD) detection. For calibration, we use the calibration regression score from Charpentier et al. (2022). For OOD detection we use the area under the receiver operating characteristic curve (AUC-ROC). For each backbone, we use the respective hyperparameters from their papers, listed in Section A.2. If not further specified, we train each model on the standard combined energy and force loss (Gasteiger

---

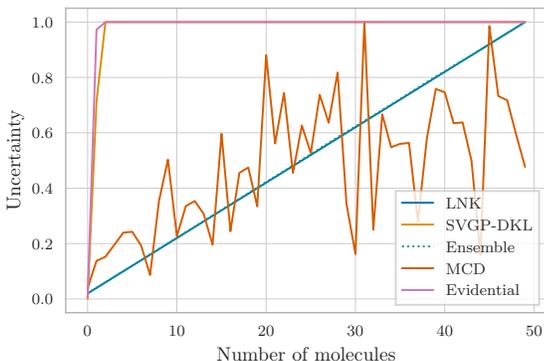[5]For SVGP-DKL versus LNK, see Section C.1

Figure 2. u_E for increasing number of identical molecules with sufficient distance to one another.

Table 2. MAE of SchNet on MD17 with different UE methods (energies in $\mathrm{kcal\,mol^{-1}}$, forces in $\mathrm{kcal\,mol^{-1}\,Å^{-1}}$)

		MCD	Evidential	fixed-LNK	Backbone	Ensemble
aspirin	Energy	2.336	1.007	**0.325**	0.321	0.272
	Forces	2.055	1.088	**0.865**	0.852	0.701
	Calibration	1.256	**1.077**	1.370	-	1.354
ethanol	Energy	0.646	0.081	**0.065**	0.064	0.06
	Forces	0.925	0.328	**0.271**	0.112	0.191
	Calibration	**1.288**	1.643	1.566	-	1.469
MDA	Energy	1.493	0.130	**0.111**	0.111	0.098
	Forces	1.467	0.535	**0.481**	0.461	0.368
	Calibration	**1.257**	1.580	1.475	-	1.442
Nap	Energy	2.739	0.346	**0.129**	0.131	0.124
	Forces	1.385	0.357	**0.266**	0.255	0.211
	Calibration	1.280	**1.258**	1.427	-	1.510
SAC	Energy	2.301	0.165	**0.142**	0.141	0.132
	Forces	1.853	0.607	**0.481**	0.465	0.378
	Calibration	**1.266**	1.527	1.504	-	1.483
toluene	Energy	1.662	0.202	**0.106**	0.108	0.099
	Forces	1.404	0.402	**0.310**	0.296	0.230
	Calibration	**1.269**	1.472	1.501	-	1.526
uracil	Energy	1.759	0.168	**0.121**	0.119	0.115
	Forces	1.945	0.480	**0.338**	0.326	0.265
	Calibration	**1.262**	1.524	1.440	-	1.543

et al., 2019):

$$\mathcal{L}_{\text{total}} = (1 - \rho)\mathcal{L}_{\text{energy}} + \rho\mathcal{L}_{\text{force}}. \quad (13)$$

Here ρ is a hyperparameter to set the importance of energy and force loss. For E, force, we specify the loss in Appendix A.2.

Methods. *Ensembles:* For Ensembles we use a sample size of 5 differently initialized models trained on the same dataset. *Monte-Carlo Dropout:* Similar to Ensembles, we choose a sample size of 5 for evaluating uncertainties. As dropout rate, we use a rate of 20%. We study different dropout rates in Section C.5 but found 20% to be good tradeoff between *Accuracy* amd *Confidence-aware*. *Evidential regression:* We adapt the energy loss according to (Soleimany et al., 2021) instead of the MAE. Note that we use a smaller coefficient for the forces loss of $\rho = 0.9$ (instead of $\rho = 0.99$ used with other models). In early experiments, we found larger force coefficients leading to poor uncertainty estimates. *SVGP-DKL and LNK:* As described in Section 4, we use the Predictive Log-Likelihood as energy loss instead of the MAE. We compare LNK against SVGP-DKL in Appendix C.1 to highlight the importance of localization. Graph Mixture Density Networks: we investigate the performance of this approach in Appendix C.2.

Datasets. *QM7-X:* (Hoja et al., 2021) This dataset covers both equilibrium and non-equilibrium structures. We train on equilibrium structures and non-equilibrium structures are considered OOD data. *MD17:* (Chmiela et al., 2017) MD17 contains energies and forces for molecular dynamics trajectories of different organic molecules.

6.1. *Locality*

In this experiment, we analyze the *Locality* property for all UE models. For this, we duplicate the same molecule

translated by $15\,Å$[6]. We measure the total uncertainty of the increasing number of molecules. For a model fulfilling *Locality*, the total uncertainty should be proportional to the number of molecules, i.e., $u_E \propto i$ where i is the number of copies.

Figure 2 depicts the dependency of the uncertainty by the number of molecules. We observe that both SVGP-DKL and Evidential do not exhibit the desired behavior. The increase of uncertainty is too steep and after three molecules, stops increasing. Ensembles and LNK meet the requirements of increasing linearly with slope 1. We can observe that MCD exhibits an increase in uncertainty with the growing number of molecules but due to its randomness, *Locality* will only be satisfied in expectation.

6.2. *Accuracy*

For this task, we use the fixed-LNK as it enables us to reuse one ensemble backbone while showing the comparison of how well the pure GNN can perform on that dataset. Hence, we also list the backbone performance as a reference. We compare the average predictive performance across all seven MD17 molecules for each combination of backbone and UE method in Table 3. One may see that while MCD and evidential regression sacrifice accuracy, LNK maintains backbone-like predictive performance. For SchNet, we provide per-molecule results in Table 2. We show additional results for other backbones in Section C.

In Table 4, we can see the predictive performance on QM7X. In energy prediction, we again observe that Ensembles, Evidential, and LNK perform well in terms of accuracy. Remarkably, MCD's energy accuracy significantly deteriorates

---

[6]The distance larger than the cutoff of all GNNs.

Table 3. Average metrics over seven molecules of MD17 for different backbones

		Dropout	Evidential	fixed-LNK	Backbone	Ensenmble
DimeNet++	Energy	1.126	0.104	**0.106**	0.109	0.144
	Forces	0.624	0.2	**0.196**	0.217	0.219
SchNet	Energy	1.848	0.3	**0.13**	0.142	0.129
	Forces	1.576	0.542	0.423	0.395	0.335
NequIP	Energy	1.557	**0.098**	0.102	0.101	0.098
	Forces	0.655	0.405	**0.342**	0.147	0.106

Table 4. MAE of Dimenet++ trained on QM7X-Equilibrium with different UE methods (energies in eV, forces in eV Å^{-1})

	MCD 20%	MCD 1%	Evidential	fixed-LNK	LNK	Ensemble
Energy	0.24	0.09	**0.02**	0.046	**0.0208**	0.0523
Forces	0.0087	**0.0043**	0.0225	0.047	0.0046	0.0034
Calibration	**1.279**	1.285	1.629	1.293	1.339	2.597

with increasing dropout rates. For forces, Ensembles, and LNK perform well. But, unlike energies, the Evidential approach significantly worsens predictive results. Despite its worse energy error, MCD achieves low force errors.

6.3. *Confidence-aware*

Now, we measure the uncertainty estimates for different approaches. For SchNet on MD17, Table 2 lists the calibration scores for each UE method on each molecule. Here, most approaches perform similarly while MCD surprisingly yields the the best results. We observe a similar pattern for QM7X in Table 4. However, the energy prediction of MCD with 0.24 is more than a magnitude worse than others and as a uniform distribution minimizes the calibration score, it only has significance if the energy prediction is good. All other methods have similar calibration scores.

In the following we test the performance of the methods on left-out class detection: we train a model on one of seven molecules and see whether it can distinguish the others based on its uncertainty estimate. An ideal estimator will have small values for the trained molecule and higher values for the other ones. This would enable a potential user to set a threshold to trust the model prediction whenever its uncertainty is lower than that threshold. To measure performance, we compute the area under the receiver operator curve of the uncertainty scores for ID and OOD data. A value closer to 1 indicates a better separation of the uncertainty scores, which means the model is better at detecting when it is operating on uncertain data.

For Evidential and LNK we use a single forward pass uncertainty estimator and for MCD and Ensembles we can use the variance of the energy prediction. In Figure 3, we can see heat maps depicting the AUC-ROC score for each pairwise combination with SchNet as backbone. The rows show the molecule that the model is trained on and the off-diagonal
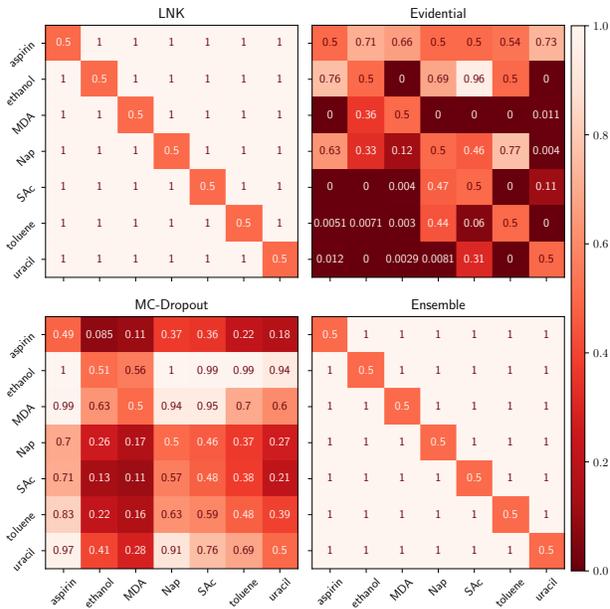

Figure 3. Heatmap of AUC-ROC values from SchNet on MD17. Each row corresponds to a separate model trained on the molecule written on the left and tested on all other molecules.

columns are the respective OOD sample. On the diagonal, we expect a score of 0.5 while a score of 1 is optimal on the off-diagonals.

We observe that MCD's and Evidential's energy uncertainties are insufficient for such a task as both results exhibit no clear diagonal structure. Meanwhile, Ensembles exhibit perfect separation between in and out-of-distribution samples. With significantly fewer computational resources, LNK achieves identical results to Ensembles in a single forward pass, see Section C for further results. This shows that both approaches can distinguish the molecule distribtions perfectly.

We further evaluate the uncertainty estimation on QM7X, where we use the equilibrium structures to train our model and try to distinguish the out-of-equilibrium (OOE) ones. This task is more challenging as the distribution of the OOD samples is closer to the ID ones as OOE structures are generated by perturbing ID molecules.

In Table 5, we can see the AUC-ROC scores with DimeNet++ as backbone. For MCD, we tested two dropout rates and neither could distinguish OOD samples. We observe similar behavior for the Evidential model. In contrast, Ensembles again achieve close to perfect separation. LNK is located in between these two extremes with an AUC-ROC score of 0.803, i.e., 2.1 times lower error than Evidential. Note, that this is obtained with a forward pass at inference, compared to five from the Ensemble.

Table 5. OOD detection on QM7X based on the energy prediction with DimeNet++ backbone.

	MCD 20%	MCD 1%	Evidential	fixed-LNK	LNK	Ensemble
AUC-ROC	0.511	0.504	0.587	0.742	**0.803**	0.947

Table 6. Inference runtime in ms on QM7X data for each UE method.

	LNK	Evidential	MCD	Ensemble
Runtime [ms]	59.3 ± 2.68	58.8 ± 5.35	229 ± 0.25	274 ± 8.25
#Trainings	1	1	1	5

Table 7. Comparison of Ensembles, MCD, and LNK on QM7X with DimeNet++ as backbone for different dropout rates.

Method	Metric	0%	1%	5%	10%	20%
MCD	Energy MAE	–	0.09	0.134	0.259	0.24
	Forces MAE	–	0.004	0.005	0.012	0.009
	AUC-ROC energy	–	0.504	0.511	0.55	0.511
	AUC-ROC forces	–	0.86	0.87	0.89	0.91
MCD-LNK	Energy MAE	–	0.059	0.024	0.039	0.036
	Forces MAE	–	0.007	0.004	0.006	0.007
	AUC-ROC energy	–	0.6	0.69	0.72	0.72
	AUC-ROC forces	–	0.79	0.82	0.84	0.9
Ensemble	Energy MAE	0.052	–	–	–	–
	Forces MAE	0.003	–	–	–	–
	AUC-ROC energy	0.947	–	–	–	–
	AUC-ROC forces	0.996	–	–	–	–

6.4. *Speed*

Evidential and LNK both only require one single forward pass[7] while MCD and Ensembles use 5 in our experiments. Thus it is natural that these techniques offer a higher *Speed*. We evaluate the average time per sample for DimeNet++ on QM7X in Table 6. We can see that both Evidential and LNK are indeed more than $4.6\times$ faster than the variance based approaches. We further observe that LNK and Evidential are having very similar runtime which is due to similar number of parameters, as we in Appendix B. Lastly, Ensembles are not only the having the computational demand at inference, but we also have to train multiple models which is additional overhead, especially in an active learning setting where one has to retrain the model many times.

6.5. Force Uncertainty

As one cannot obtain force uncertainties with a single forward pass (Gasteiger et al., 2020a), force uncertainties directly violate our *Speed* desiderata. Nontheless, to provide a broad overview we here look at force uncertainties for OOE detection. As single-forward pass methods cannot model $u_{\boldsymbol{F}}$, we use the MCD approach on LNK by leveraging the dropout at inference time, denoted as MCD-LNK. We compare against Ensembles and MCD. For all methods we obtain the force uncertainties by computing the trace of the empirical covariance matrix.

Table 7 lists the predictive performances and AUC-ROC scores for different dropout rates. We observe that MCD-LNK consistently outperforms MCD both in predictive performance as well as in OOE detection. Further, as the error increases significantly for MCD with increasing dropout rates, we find MCD-LNK to scale stably with clear improvements in OOE detection. While Ensembles report close to perfect separation, it is also the computationally most intensive method due to its multiple trainings.

---

[7]With the backpropagation to calculate the forces

Limitations

Despite LNK meeting our desiderata, some aspects remain unexplored. Firstly, while we cover several common techniques for UE for molecules this is not an exhaustive list, e.g., Bayesian neural networks (Lamb & Paige, 2020) or normalizing flows (Köhler et al., 2021). Secondly, recent research (Fu et al., 2022) has shown a gap between force prediction and simulation performance, emphasizing the importance of stability for backbones. However, due to noise in active learning (Mittal et al., 2019; Ren et al., 2021) and challenges in bridging this gap for many core architectures (Fu et al., 2022), we leave a simulation benchmark for UE as future work.

7. Conclusion

In this work, we approached the field of uncertainty estimation (UE) for molecular force fields by defining six key desiderata, three 'physics-informed' and three 'application-focused', that UE for force fields should satisfy. In a survey, we analyzed previous works based on these desiderata and found that none of them satisfies all of them. To fill the gap, we proposed LNK, a localized GP-based extension to existing GNNs. In our experimental evaluation, we found LNK to fulfill all desiderata and provide significant empirical improvements over previous single-forward methods for UE such as Monte Carlo dropout or evidential regression.

Acknowledgements

References

Amini, A., Schwarting, W., Soleimany, A., and Rus, D. Deep evidential regression. *NeurIPS*, 2020.

Bartók, A. P., Kondor, R., and Csányi, G. On representing chemical environments. *Physical Review B*, 2013.

Batatia, I., Batzner, S., Kovács, D. P., Musaelian, A., Simm, G. N. C., Drautz, R., Ortner, C., Kozinsky, B., and Csányi, G. The design space of e(3)-equivariant atom-centered interatomic potentials, 2022.

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, May 2022.

Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of Chemical Physics*, February 2011.

Biloš, M., Charpentier, B., and Günnemann, S. Uncertainty on asynchronous time event prediction. *NeurIPS*, 2019.

Bishop, C. M. and Nasrabadi, N. M. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. *ICML*, 2015.

Charpentier, B., Zügner, D., and Günnemann, S. Posterior Network: Uncertainty Estimation without OOD Samples via Density-Based Pseudo-Counts. *NeurIPS*, 2020.

Charpentier, B., Borchert, O., Zügner, D., Geisler, S., and Günnemann, S. Natural Posterior Network: Deep Bayesian Uncertainty for Exponential Family Distributions. *ICLR*, 2022.

Chmiela, S., Tkatchenko, A., Sauceda, H. E., Poltavsky, I., Schütt, K. T., and Müller, K.-R. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 2017.

Christensen, A. S., Bratholm, L. A., Faber, F. A., and Anatole von Lilienfeld, O. FCHL revisited: Faster and more accurate quantum machine learning. *The Journal of chemical physics*, 2020.

Errica, F., Bacciu, D., and Micheli, A. Graph mixture density networks, 2021.

Fu, X., Wu, Z., Wang, W., Xie, T., Keten, S., Gomez-Bombarelli, R., and Jaakkola, T. Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations, 2022.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *ICML*, 2016.

Gao, N. and Günnemann, S. Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. *ICLR*, 2022.

Gao, N. and Günnemann, S. Generalizing neural wave functions. *ICML*, 2023a.

Gao, N. and Günnemann, S. Sampling-free Inference for Ab-Initio Potential Energy Surface Networks. *ICLR*, 2023b.

Gast, J. and Roth, S. Lightweight probabilistic deep networks. *CVPR*, 2018.

Gasteiger, J., Groß, J., and Günnemann, S. Directional Message Passing for Molecular Graphs. *ICLR*, 2019.

Gasteiger, J., Giri, S., Margraf, J. T., and Günnemann, S. Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules. In *NeurIPS-W*, December 2020a.

Gasteiger, J., Giri, S., Margraf, J. T., and Günnemann, S. Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules. *Machine Learning for Molecules Workshop at NeurIPS*, 2020b.

Gasteiger, J., Becker, F., and Günnemann, S. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34, 2021a.

Gasteiger, J., Becker, F., and Günnemann, S. GemNet: Universal Directional Graph Neural Networks for Molecules. *NeurIPS*, 2021b.

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., et al. A survey of uncertainty in deep neural networks. *arxiv*, 2021.

Halgren, T. A. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *Journal of Computational Chemistry*, 17, 1996.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian Processes for Big Data. *UAI*, 2013.

Hensman, J., Matthews, A., and Ghahramani, Z. Scalable variational gaussian process classification, 2014.

Hernandez-Lobato, J. M. and Adams, R. Probabilistic backpropagation for scalable learning of bayesian neural networks. *ICML*, 2015.

Hoja, J., Medrano Sandonas, L., Ernst, B. G., Vazquez-Mayagoitia, A., DiStasio Jr., R. A., and Tkatchenko, A. QM7-X, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules. *Scientific Data*, 8, February 2021.

Jakkala, K. Deep gaussian processes: A survey. *ArXiv*, abs/2106.12135, 2021.

Jankowiak, M., Pleiss, G., and Gardner, J. R. Parametric Gaussian Process Regressors, December 2020.

Jensen, J. H. *Molecular modeling basics*. CRC Press, 2010.

Kim, H.-C. and Ghahramani, Z. Bayesian classifier combination. *AISTATS*, 2012.

Kirkpatrick, J., McMorrow, B., Turban, D. H. P., Gaunt, A. L., Spencer, J. S., Matthews, A. G. D. G., Obika, A., Thiry, L., Fortunato, M., Pfau, D., Castellanos, L. R., Petersen, S., Nelson, A. W. R., Kohli, P., Mori-Sánchez, P., Hassabis, D., and Cohen, A. J. Pushing the frontiers of density functionals by solving the fractional electron problem. *Science*, 374, December 2021.

Köhler, J., Krämer, A., and Noé, F. Smooth Normalizing Flows. *arXiv:2110.00351*, October 2021.

Kopetzki, A.-K., Charpentier, B., Zügner, D., Giri, S., and Günnemann, S. Evaluating robustness of predictive uncertainty estimation: Are dirichlet-based models reliable ? *ICML*, 2021.

Kuleshov, V., Fenner, N., and Ermon, S. Accurate uncertainties for deep learning using calibrated regression. *ICML*, 2018.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Lakshminarayanan, B., Tran, D., Liu, J., Padhy, S., Bedrax-Weiss, T., and Lin, Z. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *NeurIPS*, 2020.

Lamb, G. and Paige, B. Bayesian Graph Neural Networks for Molecular Property Prediction. *arXiv:2012.02089*, November 2020.

Lan, J., Palizhati, A., Shuaibi, M., Wood, B. M., Wander, B., Das, A., Uyttendaele, M., Zitnick, C. L., and Ulissi, Z. W. AdsorbML: Accelerating Adsorption Energy Calculations with Machine Learning, January 2023.

Leach, A. R. *Molecular modelling: principles and applications*. Pearson education, 2001.

Li, H., Wang, X., Zhang, Z., and Zhu, W. Out-Of-Distribution Generalization on Graphs: A Survey, December 2022.

Liu, Y., Wang, L., Liu, M., Lin, Y., Zhang, X., Oztekin, B., and Ji, S. Spherical Message Passing for 3D Molecular Graphs. In *ICLR*, 2021.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. *NeurIPS*, 2019.

Mittal, S., Tatarchenko, M., Özgün Çiçek, and Brox, T. Parting with illusions about deep active learning, 2019.

Moon, J., Kim, J., Shin, Y., and Hwang, S. Confidence-aware learning for deep neural networks. *ICML*, 2020.

Nandy, J., Hsu, W., and Lee, M. Towards maximizing the representation gap between in-domain & out-of-distribution examples. *NeurIPS*, 2020.

Narumi, T., Susukita, R., Ebisuzaki, T., McNiven, G., and Elmegreen, B. Molecular dynamics machine: Special-purpose computer for molecular dynamics simulations. *Molecular Simulation*, 21, 1999.

Ober, S. W., Rasmussen, C. E., and van der Wilk, M. The Promises and Pitfalls of Deep Kernel Learning, July 2021.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *NeurIPS*, 2019.

Postels, J., Ferroni, F., Coskun, H., Navab, N., and Tombari, F. Sampling-free epistemic uncertainty estimation using approximated variance propagation. *arxiv*, 2019.

Qiao, Z., Welborn, M., Anandkumar, A., Manby, F. R., and Miller III, T. F. OrbNet: Deep Learning for Quantum Chemistry Using Symmetry-Adapted Atomic-Orbital Features. *The Journal of Chemical Physics*, 153, 2020.

Qiao, Z., Christensen, A. S., Manby, F. R., Welborn, M., Anandkumar, A., and Miller III, T. F. UNiTE: Unitary N-body Tensor Equivariant Network with Applications to Quantum Chemistry. *arXiv:2105.14655*, May 2021.

Rahimi, A., Shaban, A., Cheng, C.-A., Hartley, R., and Boots, B. Intra order-preserving functions for calibration of multi-class neural networks. *Advances in Neural Information Processing Systems*, 2020.

Rasmussen, C. E. and Williams, C. K. I. Gaussian processes for machine learning. In *Adaptive computation and machine learning*, 2009.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X., and Wang, X. A survey of deep active learning, 2021.

Ritter, H., Botev, A., and Barber, D. A scalable laplace approximation for neural networks. *ICLR*, 2018.

Schütt, K., Unke, O., and Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *ICML*, 2021.

Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148, June 2018.

Sensoy, M., Kaplan, L., Cerutti, F., and Saleki, M. Uncertainty-aware deep classifiers using generative models. *AAAI*, 2020.

Shekhovtsov, A. and Flach, B. Feed-forward propagation in probabilistic neural networks with categorical and max layers. *ICLR*, 2019.

Shi, W., Zhao, X., Chen, F., and Yu, Q. Multifaceted uncertainty estimation for label-efficient deep learning. *NeurIPS*, 2020.

Simpson, E., Roberts, S., Psorakis, I., and Smith, A. Dynamic bayesian combination of multiple imperfect classifiers. *arXiv*, 2012.

Snyder, J. C., Rupp, M., Hansen, K., Müller, K.-R., and Burke, K. Finding Density Functionals with Machine Learning. *Physical Review Letters*, 108, June 2012.

Soleimany, A. P., Amini, A., Goldman, S., Rus, D., Bhatia, S. N., and Coley, C. W. Evidential deep learning for guided molecular property prediction and discovery. *ACS central science*, 7, 2021.

Song, H., Diethe, T., Kull, M., and Flach, P. Distribution calibration for regression. *arxiv*, 2019.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15, 2014.

Stadler, M., Charpentier, B., Geisler, S., Zügner, D., and Günnemann, S. Graph posterior network: Bayesian predictive uncertainty for node classification. In *NeurIPS*, 2021.

Stocker, S., Gasteiger, J., Becker, F., Günnemann, S., and Margraf, J. T. How Robust are Modern Graph Neural Network Potentials in Long and Hot Molecular Dynamics Simulations? *Machine Learning: Science and Technology*, 2022.

Strohbeck, J., Müller, J., Herrmann, M., and Buchholz, M. Deep kernel learning for uncertainty estimation in multiple trajectory prediction networks. In *IROS*, 2022.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. *arXiv:1802.08219*, May 2018.

Titsias, M. K. Variational Learning of Inducing Variables in Sparse Gaussian Processes. *AISTATS*, 2009.

Ulmer, D. A survey on evidential deep learning for single-pass uncertainty estimation. *CoRR*, 2021.

Unke, O. T. and Meuwly, M. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *Journal of Chemical Theory and Computation*, 15, June 2019.

Unke, O. T., Chmiela, S., Gastegger, M., Schütt, K. T., Sauceda, H. E., and Müller, K.-R. SpookyNet: Learning force fields with electronic degrees of freedom and nonlocal effects. *Nature Communications*, December 2021.

van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. Uncertainty estimation using a single deep deterministic neural network. *ICML*, 2020.

van Amersfoort, J., Smith, L., Jesson, A., Key, O., and Gal, Y. On feature collapse and deep kernel learning for single forward pass uncertainty. *arxiv*, 2021.

Vandermause, J., Torrisi, S. B., Batzner, S., Xie, Y., Sun, L., Kolpak, A. M., and Kozinsky, B. On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events. *Nature Computational Materials*, 6, December 2020.

Wang, H., SHI, X., and Yeung, D.-Y. Natural-parameter networks: A class of probabilistic neural networks. *NeurIPS*, 2016.

Wen, Y., Tran, D., and Ba, J. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *ICLR*, 2020.

Wenzel, F., Snoek, J., Tran, D., and Jenatton, R. Hyperparameter ensembles for robustness and uncertainty quantification. *NeurIPS*, 2020.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. Deep kernel learning, 2016.

Zhao, S., Ma, T., and Ermon, S. Individual calibration with randomized forecasting. *ICML*, 2020.

A. Implementation details

A.1. Localized Neural Kernel

To avoid constant cross-referencing, we recap essential parts of Section 4 here. Our model predicts a distribution for the energy value E_* as:

$$p(E_\star|\boldsymbol{X}, \boldsymbol{H}) = \sum_i^n \mathcal{GP}_\phi \circ h_{\mathrm{rep}}(\boldsymbol{X}, \boldsymbol{H})_i \tag{14}$$
$$\sim \mathcal{N}\left(E_\star \mid \sum_i^n \mathbb{E}[E_\star]_i, \sum_{ij}^n \mathrm{Cov}(E_\star)_{ij}\right).$$

Here h_{rep} is a GNN backbone while $\mathcal{GP}_\phi$ is the gaussian process with parameters ϕ and n is the number of atoms of the molecule. Any GNN that implements atom-wise embeddings qualifies as backbone. However, to fulfill the desiderata we use recent approaches for ML force fields that implement *Symmetry* and *Energy conservation*. We show the schematic depiction of our model in Figure 1. For one molecule one obtains n different embeddings. For each we use the same GP to predict an energy distribution. The final predictive distribution is the sum of the individual distributions.

As we use an approximate GP with inducing points we have the following additional hyperparameters we consider: number of inducing points, inducing point initialization and kernel lengthscale initialization. For the mean value of the GP prior we used a standard learn-able constant. For the number of inducing points we tested multiple different values in $\{10, , 25, 50, 100, 256, 512\}$. We choose 100 for MD17 as the training set only consists of 1000 data points and more inducing points lead to overfitting. For QM7X we used 256 as we have more data points.

For the inducing point initialization we tested three different options: first, we use randomly generated points. Secondly, we use the k-means algorithm to find multiple cluster centers within the training embeddings and choose these. Lastly, we take the first $\lfloor N_{\mathrm{ind\,pnts}}/N_{\mathrm{atoms}} \rfloor$ training embeddings for the atom embeddings. As we noticed no significant difference between the second and the third we use the last one as it is faster to compute.

For the fixed-LNK a crucial hyperparameter is the lengthscale initialization. As the inducing points might be far apart in a high-dimensional embedding space, such as for DimeNet++ with 256 dimensions, a too small initial lengthscale leads to sparse gradients and poor to no optimization of the GP posterior. Thus, we use the mean distance within all clusters of inducing points which are N_{atoms} many. If we have varying number of atoms it will N_{atoms} refers to the maximum number of atoms of a molecule in the training dataset.

Model	Hyperparameter	QM7-X	MD17
Fixed-LNK	learning rate	0.01	0.01
	patience (epochs)	50	50
	force weighting factor ρ	0.99	0.99
	# inducing points	256	100
	inducing point initialization	first k	first k
LNK	learning rate	0.001	-
	patience (epochs)	50	-
	force weighting factor ρ	0.99	-
	# inducing points	256	-
	inducing point initialization	first k	-

Table 8. Caption

A.2. Baseline details

Backbones. We use three different model architectures: DimeNet++ (Gasteiger et al., 2020b), SchNet (Schütt et al., 2018), and NequIP (Batzner et al., 2022). Unless otherwise stated, we use the default hyperparameters of each model architecture and training as stated in the original works. For DimeNet++ and SchNet, specific hyperparameters used for training are listed in Table 9. For NequIP we used exactly the same hyperparameters as the original implementation and we omit the

details here. The hyperparameters specific to the datasets and that are used with all models are listed in Table 10. Unless otherwise specified, we train the backbones the standard combined energy and force loss (Gasteiger et al., 2019):

$$\mathcal{L}_{\text{total}} = (1 - \rho)\mathcal{L}_{\text{energy}} + \rho\mathcal{L}_{\text{force}}, \tag{15}$$

$$\mathcal{L}_{\text{energy}} = \frac{1}{N} \sum_{i=1}^{N} |f_\theta(\boldsymbol{X}, \boldsymbol{H}) - E^*|, \tag{16}$$

$$\mathcal{L}_{\text{force}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{n} \sqrt{\sum_{k=1}^{3} \left(\frac{\partial}{\partial \boldsymbol{X}_i} f_\theta(\boldsymbol{X}_i, \boldsymbol{H}_i) - \boldsymbol{F}_i^* \right)_{jk}^2}. \tag{17}$$

A.3. Monte Carlo Dropout: implementation details

DimeNet++. This architecture only uses representations of dimension 0, so we use standard dropout. We apply it across different blocks after every fully-connected layer that uses an activation function. One exception is the directional message passing block, where we apply dropout after the two Hadamard product operations and once after the final fully-connected layer. Note that applying dropout separately on the two operands of the Hadamard product would result in effectively more dropped neurons than applying it after.

SchNet. Similar to DimeNet++, we here also apply dropout after each linear layer that is followed by an activation function. Specifically, we use dropout inside the interaction, convolution, and output layers.

NequIP. This architecture uses higher representations of dimension 1. Therefore, in order to preserve the equivariance property, we apply dropout at every layer of the radial network inside the interaction block which uses only representations of dimension 0. We also drop representations of dimensions 1 by applying the MCD layer described above after each convolution layer.

Table 9. Hyperparameters of DimNet++ and SchNet

Model	Hyperparameter	QM7-X	MD17
DimeNet++ Dropout	learning rate	0.002	0.001
	warmup steps	550	10000
	decay steps	6600000	1200000
	decay rate	0.01	0.01
	EMA decay	0.999	0.999
	patience (epochs)	50	50
	force weighting factor ρ	0.99	0.99
	dropout locations	embedding block, interaction block, output block	
DimeNet++ Evidential	learning rate	0.001	0.001
	warmup steps	550	10000
	decay steps	660000	1200000
	decay rate	0.01	0.01
	EMA decay	0.99	0.999
	patience (epochs)	50	50
	force weighting factor ρ	0.9	0.9
	evidential loss weighting factor λ	0.2	0.2
SchNet Dropout	learning rate	0.001	0.001
	warmup steps	1000	10000
	decay steps	4000000	4000000
	decay rate	0.01	0.01
	EMA decay	0.999	0.999
	patience (epochs)	50	50
	force weighting factor ρ	0.99	0.99
	dropout locations	interaction block, convolution layers, output block	
SchNet Evidential	learning rate	0.001	0.001
	warmup steps	1000	1000
	decay steps	4000000	4000000
	decay rate	0.01	0.01
	EMA decay	0.999	0.999
	patience (epochs)	50	50
	force weighting factor ρ	0.9	0.9
	evidential loss weighting factor λ	0.2	0.2

Table 10. Hyperparameters of the datasets used with all models

	QM7-X	MD17
batch size	32	1
train set size	33213	1000
val set size	4151	1000
standardize data	False	True

B. Complexity details

In Table 11 we show the number of parameters for each model. We used the default parameter settings of the respective paper. Notably, our model LNK, introduces only a marginal increase in additional parameters compared to existing models. As seen in Table 6, the slight increase in parameters translates to a minimal decrease in computational speed compared to the Evidential model.

We observe that Ensembles have five times more parameters than the other models. An interesting future work may be investigating whether the Ensembles can be reduced in their number of parameters to have the same computational demand as the single model. However, according to research conducted by Batatia et al. (2022), a reduction in the number of channels on NequIP is reduced from 32 to 16 (which is only a reduction of about 2.6), resulted in 8.4% decrease in force prediction performance and a 19.35% decline in energy prediction performance.

Approach	DimeNet++	SchNet	NequIP
MCD	1,885,830	455,809	189,080
Evidential	1,889,670	456,004	189,128
LNK	1,920,150	468,263	197,567
Ensemble	9,429,150	2,279,045	1,119,800

Table 11. Number of parameters for each UE approach and backbone. SchNet is the version of Pytorch-Geometric with 6 interaction blocks.

C. Additional Experiments

In the following we list additional experiments, i.e., evaluating different backbones or approaches.

C.1. LNK vs. SVGP-DKL

To analyze the improvement from the localization of LNK, we compare the performance of LNK against SVGP-DKL on MD17 and QM7X. The results are listed in Table 12. For MD17 We observe that LNK decreases the error by 5% on average for the energy prediction and by 28% for forces. For QM7X we decrease the error by 99.3% for energy but increase the forces slightly by 6.5%.

C.2. Graph Mixture Density Network

In the following we investigate the performance of the Graph Mixture Density Network (GMDN) (Errica et al., 2021). This network outputs a mixture of distributions as resulting predictive distribution. Each individual distribution is learned by a neural network head. Here we investigate *Accuracy* and *Confidence-aware*. In Appendix D, we show that the network is not necessarily fulfilling *Locality*.

In Table 13, we observe that LNK is outperforming GMDN on every single target. For aspirin and napthalene the model did not learn to predict the energy properly. As GMDN was not tested for molecular property prediction by the authors, trained for different hyperparamters for the number of hidden units, the number of mixture components and the number of output layers, see Appendix D for a detailed model description. For each target we used the best performing setting, i.e., not a fix hyperparameter setting for all molecules on MD17 as done for all other baselines but the best set for each target. However, there might be hyperparameter sets where this network yields better results and we leave it for future work to adapt GMDN to this setting.

In Figure 4, we see the OOD detection performance we use to measure *Confidence-aware*. Compared to MCD or Evidential, the method has higher separation for most of the OOD molecules. However, it is not close to the results of Ensemble or LNK.

C.3. Additional experiments on MD17

In the following we list the experimental results for the other backbones. We use either 69 samples or 5 samples for MCD evaluation.

Dataset	Training Target		LNK	SVGP-DKL
MD17	aspirin	Energy	**0.145**	0.208
		Forces	**0.359**	0.504
	ethanol	Energy	**0.054**	0.065
		Forces	**0.201**	0.296
	malonaldehyde	Energy	**0.096**	0.107
		Forces	**0.299**	0.478
	napthalene	Energy	0.133	**0.107**
		Forces	**0.121**	0.166
	salicylic acid	Energy	**0.119**	**0.12**
		Forces	**0.257**	0.324
	toluene	Energy	0.094	**0.091**
		Forces	**0.122**	0.191
	uracil	Energy	**0.102**	0.108
		Forces	**0.222**	0.265
QM7X	Equilibrium	Energy	**0.05**	7.047
		Forces	0.049	**0.046**

Table 12. Comparison of LNK vs SVGP-DKL with DimeNet++ as backbone.

Dataset	Training Target		LNK	GMDN
MD17	aspirin	Energy	**0.325**	174.051
		Forces	**0.865**	0.907
	ethanol	Energy	**0.065**	0.071
		Forces	**0.271**	0.356
	malonaldehyde	Energy	**0.111**	0.124
		Forces	**0.481**	0.585
	napthalene	Energy	**0.129**	85.265
		Forces	**0.266**	0.316
	salicylic acid	Energy	**0.142**	0.160
		Forces	**0.481**	0.605
	toluene	Energy	**0.106**	0.112
		Forces	**0.310**	0.381
	uracil	Energy	**0.121**	0.129
		Forces	**0.338**	0.476

Table 13. Comparison of LNK vs GMDN with SchNet as backbone. For MD17 We observe that LNK outperforms GMDN on every target.

C.4. Additional experiments on QM7X

In the following we show the experimental results for QM7X for other backbones and for dropout with 5 and 69 samples respectively.

C.5. Dropout Rate Experiment

We compare different dropout probabilities for MCD on QM7X for DimeNet++ and SchNet.
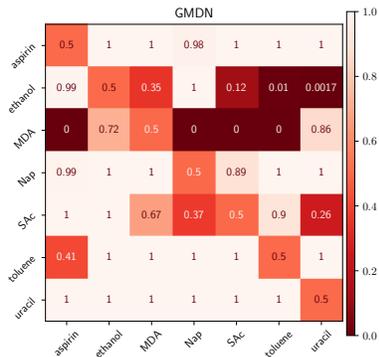
Figure 4. OOD detection with GMDN and SchNet as backbone model.

Table 14. MAE of Dimenet++ on MD17 with different uncertainty estimation methods. MCD stands for MC-Dropout (energies in $\text{kcal}\,\text{mol}^{-1}$, forces in $\text{kcal}\,\text{mol}^{-1}\,\text{Å}^{-1}$)

		MCD ($N = 5$)	MCD ($N = 69$)	Evidential	LNK	Backbone	Ensemble
aspirin	Energy	1.243	0.445	**0.165**	0.175	0.174	0.155
	Forces	0.793	0.633	0.388	**0.376**	0.356	0.281
	Calibration	1.263	1.517	**0.480**	1.686	-	1.454
ethanol	Energy	0.892	0.280	**0.054**	**0.055**	0.055	0.054
	Forces	0.360	0.260	**0.154**	0.186	0.173	0.137
	Calibration	**1.282**	1.572	1.487	1.688	-	1.528
malonaldehyde	Energy	1.147	0.373	**0.080**	0.093	0.095	0.09
	Forces	0.632	0.463	**0.254**	0.336	0.326	0.269
	Calibration	**1.287**	1.563	1.500	1.680	-	1.514
napthalene	Energy	1.078	0.338	**0.117**	**0.117**	0.114	0.117
	Forces	0.536	0.412	**0.098**	0.114	0.101	0.075
	Calibration	**1.283**	1.555	1.512	1.664	-	1.613
salicylic acid	Energy	1.215	0.392	**0.110**	0.121	0.12	0.113
	Forces	0.762	0.585	**0.220**	0.274	0.243	0.19
	Calibration	1.299	1.561	**0.699**	1.687	-	1.562
toluene	Energy	1.09	0.331	0.092	**0.091**	0.09	0.09
	Forces	0.52	0.378	**0.104**	0.136	0.121	0.089
	Calibration	**1.287**	1.571	1.582	1.683	-	1.582
uracil	Energy	1.219	0.399	**0.110**	0.112	0.116	0.387
	Forces	0.762	0.577	**0.158**	0.223	0.196	0.49
	Calibration	1.273	1.556	**0.816**	1.688	-	1.11

Table 15. MAE of NequIP on MD17 with different uncertainty estimation methods (energies in $\mathrm{kcal\,mol^{-1}}$, forces in $\mathrm{kcal\,mol^{-1}\,\mathring{A}^{-1}}$)

		MCD ($N=5$)	MCD ($N=69$)	Evidential	LNK	Backbone	Ensemble
aspirin	Energy	2.466	0.744	0.194	**0.142**	0.153	0.144
	Forces	0.943	0.754	0.54	**0.304**	0.287	0.210
	Calibration	**1.278**	1.553	1.688	-	-	1.472
ethanol	Energy	0.932	0.272	0.073	**0.056**	0.053	0.050
	Forces	0.588	0.410	0.401	**0.15**	0.137	0.096
	Calibration	**1.283**	1.576	1.692	-	-	1.518
malonaldehyde	Energy	1.131	0.432	0.103	**0.086**	0.079	0.075
	Forces	0.733	0.535	0.496	**0.307**	0.234	0.166
	Calibration	**1.271**	1.538	1.675	-	-	1.494
napthalene	Energy	1.615	0.456	0.066	-	0.114	0.115
	Forces	0.445	0.296	0.3	-	0.054	0.042
	Calibration	**1.273**	1.584	1.683	-	-	1.619
salicylic acid	Energy	1.843	0.561	**0.11**	0.127	0.109	0.107
	Forces	0.750	0.560	0.433	**0.27**	0.146	0.103
	Calibration	**1.275**	1.580	1.667	-	-	1.558
toluene	Energy	1.428	0.437	**0.071**	0.093	0.089	0.089
	Forces	0.513	0.348	0.299	**0.068**	0.069	0.051
	Calibration	**1.272**	1.594	1.689	-	-	1.611
uracil	Energy	1.485	0.449	**0.069**	0.109	0.107	0.104
	Forces	0.611	0.407	**0.363**	0.955	0.101	0.073
	Calibration	**1.293**	1.583	1.701	-	-	1.597

Figure 5. Heatmap of AUC-ROC on MD17 values based on different UE methods with DimeNet++ as backbone. Each row corresponds to a separate model trained on the molecule written on the left and tested on all other molecules.

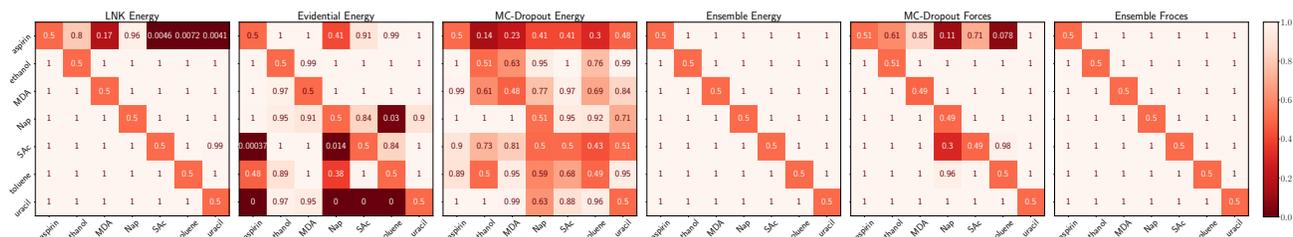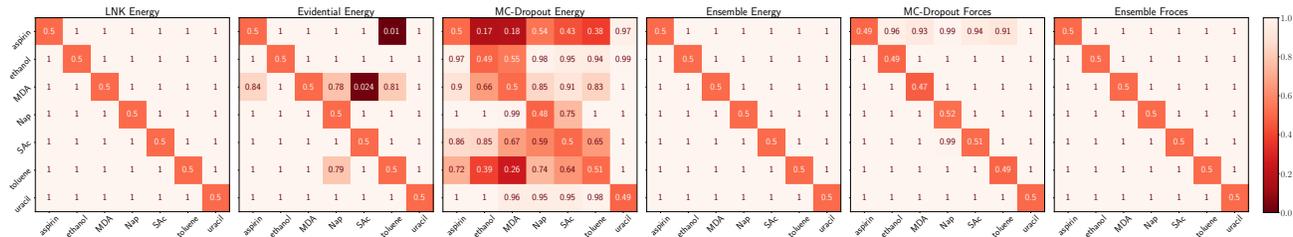*Figure 6.* Heatmap of AUC-ROC on MD17 values based on different UE methods with NequIP as backbone. Each row corresponds to a separate model trained on the molecule written on the left and tested on all other molecules.

Table 16. MAE of SchNet on MD17 with different uncertainty estimation methods (energies in $\mathrm{kcal\,mol^{-1}}$, forces in $\mathrm{kcal\,mol^{-1}\,\mathring{A}^{-1}}$)

		MCD ($N = 5$)	MCD ($N = 69$)	Evidential	LNK	Backbone	Ensemble
aspirin	Energy	2.336	1.302	1.007	**0.271**	0.321	0.272
	Forces	2.055	1.773	1.088	**0.857**	0.852	0.701
	Calibration	1.256	1.419	1.077	**0.78**	-	2.275
ethanol	Energy	0.646	0.284	0.081	**0.066**	0.064	0.06
	Forces	0.925	0.697	0.328	**0.298**	0.112	0.191
	Calibration	1.288	1.479	1.643	**0.84**	-	2.538
malonaldehyde	Energy	1.493	0.942	0.130	**0.106**	0.111	0.098
	Forces	1.467	1.185	0.535	**0.465**	0.461	0.368
	Calibration	1.257	1.319	1.580	**1.208**	-	2.401
napthalene	Energy	2.739	1.645	0.346	**0.132**	0.131	0.124
	Forces	1.385	1.119	0.357	**0.263**	0.255	0.211
	Calibration	1.280	1.330	**1.258**	1.297	-	2.601
salicylic acid	Energy	2.301	1.310	0.165	**0.132**	0.141	0.132
	Forces	1.853	1.535	0.607	**0.457**	0.465	0.378
	Calibration	1.266	1.389	1.527	**0.801**	-	2.519
toluene	Energy	1.662	0.621	0.202	**0.106**	0.108	0.099
	Forces	1.404	1.155	0.402	**0.282**	0.296	0.230
	Calibration	1.269	1.535	1.472	**1.042**	-	2.649
uracil	Energy	1.759	0.959	0.168	**0.123**	0.119	0.115
	Forces	1.945	1.656	0.480	**0.341**	0.326	0.265
	Calibration	**1.262**	1.396	1.524	1.370	-	2.725

Table 17. MAE of Dimenet++ trained on QM7X-Equilibrium with different uncertainty estimation methods (energies in eV, forces in $\mathrm{eV\,\mathring{A}^{-1}}$)

	MC-Dropout 20%	MC-Dropout 1%	Evidential	LNK	LNK + Drop 5%
Energy ID	0.240	0.090	**0.020**	0.049	0.0209
Forces ID	0.009	**0.004**	0.023	0.047	0.0045
Energy OOD	5.242	5.277	5.172	5.530	5.230
Forces OOD	1.266	1.282	1.367	1.361	1.293
Calibration	**1.279**	1.285	1.629	2.596	2.596
AUC-ROC energy	0.511	0.504	0.587	**0.795**	0.7913
AUC-ROC forces (trace)	**0.905**	0.858	——	0.729	0.8278

Table 18. MAE of Dimenet++ ($N = 69$ samples) trained on QM7X-Equilibrium with different uncertainty estimation methods (energies in eV, forces in eV Å^{-1})

	MC-Dropout 20%	MC-Dropout 1%	Evidential	LNK	LNK + Drop 1%
Energy ID	0.165	0.077	**0.020**	0.049	0.0265
Forces ID	0.007	**0.004**	0.023	0.047	0.0061
Energy OOD	5.022	5.063	5.172	5.53	5.23
Forces OOD	1.27	1.271	1.367	1.361	1.293
Calibration	1.367	**1.296**	1.629	2.596	2.596
AUC-ROC energy	0.513	0.521	0.587	**0.795**	0.74
AUC-ROC forces (trace)	**0.911**	0.865	——	0.729	0.854

Table 19. MAE of SchNet, and NequIP trained on QM7X-Equilibrium with different uncertainty estimation methods (energies in eV, forces in eV Å^{-1}). The numbers marked with a star denote that they were obtained after removing 7 out of the 1537 test molecules that were causing numerical instability of the NequIP model.

	MC-Dropout		Evidential	
	SchNet	NequIP	SchNet	NequIP
Energy ID	0.44	0.158	0.027	0.019
Forces ID	0.025	0.019	0.027	0.009
Energy OOD	5.292	37.751*	5.322	21.24
Forces OOD	1.271	26.139*	1.319	84.441
Calibration	1.346	1.272	1.670	1.549
AUC-ROC energy	0.501	0.627	0.741	0.746
AUC-ROC forces (trace)	0.8	0.988	——	——

Table 20. MAE of SchNet, and NequIP ($N = 69$ samples) trained on QM7X-Equilibrium with different uncertainty estimation methods (energies in eV, forces in eV Å^{-1}). The numbers marked with a star denote that they were obtained after removing 7 out of the 1537 test molecules that were causing numerical instability of the NequIP model.

	MC-Dropout		Evidential	
	SchNet	NequIP	SchNet	NequIP
Energy ID	0.423	0.109	0.027	0.019
Forces ID	0.023	0.017	0.027	0.009
Energy OOD	5.291	36.253*	5.322	21.24
Forces OOD	1.271	25.937*	1.319	84.441
Calibration	1.274	1.329	1.670	1.549
AUC-ROC energy	0.478	0.68	0.741	0.746
AUC-ROC forces (trace)	0.81	0.992	——	——

Table 21. MAE and uncertainty metrics of MC-Dropout Dimenet++ ($N = 5$ samples) trained on QM7X-Equilibrium with different dropout rates (energies in eV, forces in eV Å^{-1})

Dropout Rate	1%	5%	10%	15%	20%
Energy ID	0.09	0.134	0.259	0.206	0.24
Forces ID	0.004	0.005	0.012	0.008	0.009
Energy OOD	5.277	5.319	4.978	5.245	5.242
Forces OOD	1.282	1.288	1.675	1.273	1.266
Calibration	1.285	1.275	1.273	1.271	1.279
AUC-ROC energy	0.504	0.511	0.55	0.517	0.511
AUC-ROC forces (trace)	0.858	0.872	0.886	0.904	0.905

Table 22. MAE and uncertainty metrics of MC-Dropout SchNet ($N = 5$ samples) trained on QM7X-Equilibrium with different dropout rates (energies in eV, forces in eV Å^{-1})

Dropout Rate	1%	5%	10%	15%	20%
Energy ID	0.177	0.256	0.359	0.430	0.440
Forces ID	0.012	0.015	0.019	0.022	0.025
Energy OOD	5.126	5.135	5.036	5.235	5.292
Forces OOD	1.255	1.262	1.260	1.260	1.271
Calibration	1.318	1.328	1.347	1.332	1.346
AUC-ROC energy	0.492	0.493	0.496	0.484	0.501
AUC-ROC forces (trace)	0.816	0.845	0.826	0.811	0.800

D. *Locality* consideration of baseline methods

Ensemble As written above, we estimate the uncertainty of a molecule for an ensemble $f_1, \ldots, f_5$ of five models f_i as:

$$\mathbf{u}_E(\boldsymbol{M}) = \sqrt{\text{Var}\left(f_1(\boldsymbol{M}), \ldots, f_N(\boldsymbol{M})\right)}. \tag{18}$$

If we duplicate the molecule and have it sufficiently far apart, then we get:

$$f_i(\{\boldsymbol{M}, \boldsymbol{M}\}) = 2 * f_i(\boldsymbol{M}), \tag{19}$$

as the embeddings of the respective molecules do not influence each other and the final result is the sum over the atom predictions. Hence, we can derive that the uncertainty estimate of the ensemble is satisfies *Locality*:

$$\begin{aligned}
u_E(\{\boldsymbol{M}, \boldsymbol{M}\}) &= \sqrt{\text{Var}\left(f_1(\{\boldsymbol{M}, \boldsymbol{M}\}), \ldots, f_N(\{\boldsymbol{M}, \boldsymbol{M}\}))\right)} \\
&= \sqrt{\text{Var}\left(2f_1(\boldsymbol{M}), \ldots, 2f_N(\boldsymbol{M})\right)} \\
&= \sqrt{4 \cdot \text{Var}\left(f_1(\boldsymbol{M}), \ldots, f_N(\boldsymbol{M})\right)} \\
&= 2u_E(\boldsymbol{M}).
\end{aligned} \tag{20}$$

Graph Mixture Density Networks (Errica et al., 2021) can be formulated as:

$$\sum_{i=1}^{C} P(y_g | Q_g^i, g) P(Q_g^i | g) \tag{21}$$

where C is the number of mixing components (experts), g is the graph, Q_g^i is the mixing weight i for graph g. It is modelled by a network Φ_i. y_g is the target we want to model, i.e. $P(y_g | g)$. Given the node embeddings

$$\mathbf{h}_v^{l+1} = \phi^{l+1}\left(\mathbf{h}_v^l, \Psi(\{\psi^{l+1}(\mathbf{h}_u^l) | u \in \mathcal{N}_v\})\right), \forall l = 1 \ldots L \tag{22}$$

The final node representation is given by $\mathbf{h}_v := \mathbf{h}_v^L$. $P(Q_g^i | g)$ models a mixing weight by a neural network. The interesting part is the emission $P(y_g | Q_g^i, g)$, i.e. the distribution parameters. They are compute as:

$$\mu_i, \Sigma_i = \Phi_i(\mathbf{h}_g) = f_i(r_g^i(\mathbf{h}_{\mathcal{V}_g})). \tag{23}$$

Here, $\mathbf{h}_g$ is the aggregated embedding for the full graph, Φ_i is a sub-network to compute the individual distributional parameter, f_i is a neural network, r_g^i is the readout network and $\mathcal{V}_g$ is the set of nodes of graph g.

If we now construct a graph $\hat{g} := \{g, g\}$ whithout edges between the individual graphs g, then, due to the non-linearity of f_i (or Φ_i), we would have:

$$\Phi_i(\mathbf{h}_{\hat{g}}) = \hat{\mu}_i, \hat{\Sigma}_i \neq 2\mu_i, 2\Sigma_i = \Phi_i(\mathbf{h}_g). \tag{24}$$

Hence, GMDN is not necessarily fulfilling *Locality*.

E. Metrics

E.1. Calibration

We use the calibration score introduced by Kuleshov et al. (2018). It computes the difference between the percentage of targets lying in a certain confidence region and the corresponding percentile p. In mathematical notation, the calibration metric is:

$$p_{pred} = \frac{1}{N} \sum_i \mathbb{1} \left[\mathbb{P} \left(y \leq y^{*,(i)} | \theta^{(i)} \right) \in I_p \right],$$ (25)

where $I_p = [0, \frac{p}{2}] \cup [1 - \frac{p}{2}, 1]$. To obtain a single score we compute the norm:

$$s_{\text{calib}} := \sqrt{\sum_p (p - p_{\text{pred}})^2} \quad , \quad p \in \{0.1, \ldots, 0.9\}$$ (26)

E.2. OOD Detection

The OOD detection task can be viewed as a binary classification task. We assign class 1 to ID data and class 0 to OOD data and use the predicted uncertainty values as scores, based on which we want to distinguish between ID and OOD data. This enables to compute UE metrics using, for instance, the area under the receiver operating characteristic curve (AUC-ROC). We obtain numbers in [0,1], where higher values indicate better performance. For the MC-Dropout and the ensemble methods, we use the variance of the energy samples as the uncertainty estimate corresponding to the energy prediction and the trace of the sample covariance matrix of the forces samples as the uncertainty estimate corresponding to the forces prediction. We found that the trace is more stable than taking the determinant and leads to good results. For the evidential models, we use the analytic variance of the energy prediction, which we can compute in closed form based on the parameters of the evidential distribution, as the uncertainty estimate corresponding to the energy prediction. We don't use any uncertainty estimates for the forces prediction, because there is no obvious way how to do that.