BAYESIAN LINEAR LAST LAYER FOR MACHINE LEARNING INTERATOMIC POTENTIALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliable uncertainty quantification is essential for deploying Machine Learning Interatomic Potentials (MLIPs), especially when molecular dynamics or materials simulations encounter configurations outside the training distribution. Deep ensembles are among the strongest practical baselines for MLIP uncertainty, but training and storing several copies of a modern pretrained model is often prohibitively expensive. We show that Bayesian Linear Last Layers (BLLs) provide a scalable alternative for MLIPs: a single pretrained backbone supplies atomic features, while exact Bayesian inference over the final force-prediction layer gives predictive uncertainties; the construction extends directly to equivariant force heads. However, BLLs are known to be overconfident, so we analyze the sources of this miscalibration and introduce a simple post-hoc recalibration to address the issue. We show that BLLs recalibrated on in-distribution examples are competitive with LoRA ensembles on several foundation backbones for molecules and materials.

1 INTRODUCTION

Computational chemistry and materials science are currently undergoing a rapid transformation driven by MLIPs. While ab initio methods like Density Functional Theory (DFT) provide high accuracy, their cubic scaling with system size makes them prohibitively expensive for large-scale systems or long-duration molecular dynamics (MD) simulations. MLIPs bridge this gap by learning to map atomic configurations directly onto the potential energy surface, offering DFT-level accuracy at a fraction of the cost. Recent architectural innovations and the emergence of massive, heterogeneous datasets have led to "universal" foundation models, such as the Orb-v3 (Rhodes et al., 2025) and eSEN (Fu et al., 2025) architectures trained on datasets such as OMat24 (Barroso-Luque et al., 2024) and OMol25 (Levine et al., 2025) making the neural approach increasingly attractive.However, the neural surrogate approach is highly susceptible to catastrophic failures when encountering out-of-distribution (OOD) atomic configurations. This vulnerability is particularly acute in MD simulations. A single unphysical prediction can push the system into a high-energy distorted state absent from the training data resulting in compounding errors.

Uncertainty Quantification (UQ) offers a path forward. Using UQ, one could identify inaccurate predictions at inference time, potentially triggering a more expensive, more accurate computation for a given configuration (Pilania et al., 2017). Moreover, Bayesian uncertainties provide a natural pathway within an active learning loop, in which accurate DFT calculations can be used to further improve a model. Unfortunately, UQ remains an effectively unresolved problem in deep learning (Papamarkou et al., 2024), where the practical gold standard, deep ensembles (Tan et al., 2023), are prohibitively costly at the scale of foundation models. This paper investigates the feasibility of Bayesian Linear Last Layer (BLL) for MLIPs, in which exact Bayesian posterior estimation is limited to the final linear head, and asks when, and how, a BLL on a frozen backbone gives useful uncertainty for MLIPs. While simple and elegant, BLL is known to underestimate the uncertainty. We find that:

- **BLL can be built at foundation scale for structured data of molecules and materials.** We fit BLLs on frozen pretrained backbones, including an equivariant variant for force heads built from $l = 1$ vector features, with an exact one-pass fit on almost one billion atoms.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- **BLL miscalibration has identifiable causes and we propose an effective post-hoc correction.** Empirical Bayes inflates the residual covariance to absorb misspecification, while the input-dependent term shrinks as $1/N$. Our analysis suggests a decoupled recalibration of the two terms.
- **Recalibrated BLL is competitive with LoRA ensembles.** On eSEN backbones and under distribution shift on MPtrj, BLL is surpassing or on par with LoRA ensembles, but falls behind on OrbMol.

2 Related Work

Uncertainty quantification in MLIPs. The need for reliable uncertainty estimates in MLIPs has been recognized since their early days (Bartók et al., 2010). The dominant approach remains deep ensembles (Lakshminarayanan et al., 2017), in which multiple independently trained models provide both a mean prediction and a measure of disagreement. A systematic comparison by Tan et al. (2023) across several atomic force and energy benchmarks found that ensembles consistently outperformed single-model alternatives such as Gaussian mixture models, deep evidential regression, and mean-variance estimation. This has cemented ensembles as the practical gold standard for UQ in MLIPs. Training and storing multiple copies of large foundation models such as UMA (Wood et al., 2025), Orb-v3 (Rhodes et al., 2025), and eSEN (Fu et al., 2025) is, however, prohibitive at scale, motivating the search for cheaper single-model alternatives that can match ensemble-level calibration. Bayesian alternatives designed for MLIPs include BLIP (Coscia et al., 2025), which applies variational dropout throughout the network and which we use as a baseline.

Bayesian neural networks. BNNs provide a principled framework for UQ by maintaining a posterior over model parameters (MacKay, 1992; Neal, 1996), with approximate inference via MCMC (Izmailov et al., 2021), variational inference (Blundell et al., 2015; Graves, 2011; Thaler et al., 2023), and Laplace approximations (Ritter et al., 2018; Daxberger et al., 2021). Several works draw formal connections between deep ensembles and Bayesian inference (Wilson & Izmailov, 2020; Pearce et al., 2020), suggesting that the empirical success of ensembles partly reflects implicit Bayesian properties.

Gaussian processes for interatomic potentials. A natural compromise between full BNNs and point-estimate networks is to apply Bayesian inference to features extracted by the deep network. The Gaussian Approximation Potential (GAP) framework (Bartók et al., 2010) pairs Gaussian processes (GPs) with handcrafted atomic descriptors such as SOAP (Bartók et al., 2013) to learn potential energy surfaces with native uncertainty estimates. GPs are attractive in this setting because exact posterior inference is available in closed form, but their non-parametric exact inference scales as $O(N^3)$ in the dataset size and limits their applicability to the large, heterogeneous datasets underpinning modern foundation MLIPs.

Bayesian last-layer methods. BLL retains the closed-form posterior of GPs but shifts the cost from N to the feature dimension D, making it tractable at MLIP scale; it is mathematically equivalent to a GP with a linear kernel in the learned feature space (Rasmussen & Williams, 2006). This neural-linear model has a long history: Snoek et al. (2015) used it for scalable Bayesian optimization, deep kernel learning (Wilson et al., 2016a;b) learns the feature map through the GP marginal likelihood, and Ober & Rasmussen (2019) benchmark it for regression and find that its performance depends strongly on how the features and hyperparameters are learned. Kristiadi et al. (2020) show that a Gaussian posterior over the last layer alone suffices to fix the asymptotic overconfidence of ReLU networks, and the last-layer Laplace approximation (LLLA; Daxberger et al., 2021) makes this a post-hoc tool for pretrained networks, which we use as a baseline. Watson et al. (2021) showed that BLLs can outperform full BNNs trained with Bayes-by-Backprop on standard regression tasks. The main limitation of last-layer methods is feature collapse: deterministic feature extractors tend to map dissimilar inputs to nearby representations, which can make the last-layer posterior overconfident on out-of-distribution inputs (van Amersfoort et al., 2021). Existing remedies modify the backbone, e.g., spectral normalization in SNGP (Liu et al., 2020) and DUQ (van Amersfoort et al., 2020). We instead operate in the foundation-MLIP regime in which the backbone is taken as fixed. We then address calibration post hoc, characterize the residual error this process leaves uncorrected (section 4.2), and measure the collapse through the effective rank of the features (section 5).

Post-hoc calibration. Post-hoc methods adjust pretrained model uncertainties using held-out data, with temperature scaling (Guo et al., 2017) the canonical approach in classification. Calibration in regression has received comparatively less attention (Kuleshov et al., 2018). The generalized posterior and SafeBayes literature (Grünwald & van Ommen, 2018; Bissiri et al., 2016) provides theoretical grounding for tempering the posterior under model misspecification, but applies a single temperature to the entire likelihood. Our analysis motivates a two-parameter rescaling that also works well in practice. When residuals are heavy-tailed, a Gaussian predictive can be calibrated on average but not at its quantiles; Student-t likelihoods are the standard robust alternative in regression (Lange et al., 1989). We use this family post hoc (section 4.3).

3 Bayesian Linear Last Layer for MLIPs

We estimate uncertainty by placing a Bayesian linear model on the final force-prediction layer of an MLIP, which is often a graph neural network (Schütt et al., 2017) or a transformer (Liao et al., 2024), while treating the rest of the network as a deterministic feature extractor. Let ϕ_θ denote the map from atomic configuration to penultimate force-head representation. For an atomic configuration with n_a atoms, the model produces per-atom features $\phi_i \in \mathbb{R}^D$, which are mapped to Cartesian forces in $\mathbb{R}^3$. Since n_a varies across structures such as molecules or unit cells, we take atoms rather than structures as the unit of Bayesian inference; therefore, a dataset of structures is flattened into N atoms, producing a fixed-dimensional regression problem with features $\Phi \in \mathbb{R}^{N\times D}$ and force targets $Y \in \mathbb{R}^{N\times K}$, where $K = 3$; we discuss mini-batching later. To save space, all equations are written inline; their numbered forms, which the equation references point to, are collected in section A. The final linear force head is modelled as $Y = \Phi B + E$ with $B \in \mathbb{R}^{D\times K}$. This per-atom formulation avoids padding or structure-level aggregation and gives a single posterior over B that can be used for structures of arbitrary size. The same last-layer weights are shared across all atoms; information about the parent structure enters through the node-level features produced by the backbone. For the Bayesian treatment, we need to specify the form of the likelihood and the prior.

Likelihood. We use a *matrix-normal* likelihood, $Y \mid B, \Phi, \Sigma_e \sim \mathcal{MN}(\Phi B,\, I_N,\, \Sigma_e)$, or, equivalently, $\mathrm{vec}(Y) \mid B, \Phi, \Sigma_e \sim \mathcal{N}\left(\mathrm{vec}(\Phi B),\; \Sigma_e \otimes I_N\right)$, where $\mathrm{vec}(\cdot)$ stacks the columns of its matrix argument. This is the same as assuming $\varepsilon_i \sim \mathcal{N}(0, \Sigma_e)$ independently across atomic environments. The identity row covariance is an important approximation: conditional on B and the atomic features Φ, the residual force errors of different atomic observations are modelled as independent, including atoms belonging to the same structure.[1]

Prior. We use the conjugate matrix-normal prior $B \mid \Lambda_0, \Sigma_e \sim \mathcal{MN}(0,\, \Lambda_0^{-1},\, \Sigma_e)$, where $\Lambda_0 = \mathrm{diag}(\lambda_1, \ldots, \lambda_D)$ is the prior precision over the features. Sharing the column covariance Σ_e between the likelihood and prior is the standard conjugate construction that results in a closed-form posterior. The diagonal precision provides a scalable feature-wise shrinkage prior. A dense $D \times D$ prior covariance would introduce substantially more parameters and more fragile inversions, while the diagonal form is sufficient to regularize poorly constrained feature directions and to enter the marginal likelihood through the Bayesian complexity penalty. More importantly, at this data scale the likelihood dominates, so the prior's role is mainly to regularize poorly constrained feature directions.

Posterior. Combining equation 2 and equation 4, the posterior over the last-layer weights is $B \mid Y, \Phi, \Lambda_0, \Sigma_e \sim \mathcal{MN}(B_n,\, \Lambda_n^{-1},\, \Sigma_e)$, with $\Lambda_n = \Phi^\top \Phi + \Lambda_0$ and $B_n = \Lambda_n^{-1} \Phi^\top Y$.

Marginal likelihood. We estimate Λ_0 and Σ_e by empirical Bayes. The marginal likelihood is the density of the observed force targets under the likelihood-prior model after removing the unobserved last-layer weights, $p(Y \mid \Phi, \Lambda_0, \Sigma_e) = \int p(Y \mid B, \Phi, \Sigma_e)\, p(B \mid \Lambda_0, \Sigma_e)\, \mathrm{d}B$. Here $p(\cdot)$ denotes the densities associated with the likelihood and prior in eqs. (2) and (4). Thus, B is integrated out of the joint density $p(Y, B \mid \Phi, \Lambda_0, \Sigma_e)$. Evaluating this Gaussian integral and taking the negative logarithm gives the negative log marginal likelihood (NLML) $\mathcal{L}_{\mathrm{NLML}} = \frac{NK}{2}\log(2\pi) + \frac{N}{2}\log|\Sigma_e| + \frac{K}{2}(\log|\Lambda_n| - \log|\Lambda_0|) + \frac{1}{2}\,\mathrm{tr}[\Sigma_e^{-1}(R^\top R + B_n^\top \Lambda_0 B_n)]$, with residuals $R = Y - \Phi B_n$. Equation (7) is differentiable in the hyperparameters and, through Φ, in the parameters of the feature extractor.

[1] This does not imply that the model's predicted forces are marginally independent after Bayesian inference. Once B is integrated out, predictions at two atoms with features ϕ_i and ϕ_j become correlated through the shared posterior uncertainty in B: $\mathrm{Cov}(y_i, y_j \mid Y, \Phi) = \delta_{ij}\Sigma_e + (\phi_i^\top \Lambda_n^{-1} \phi_j)\Sigma_e$.

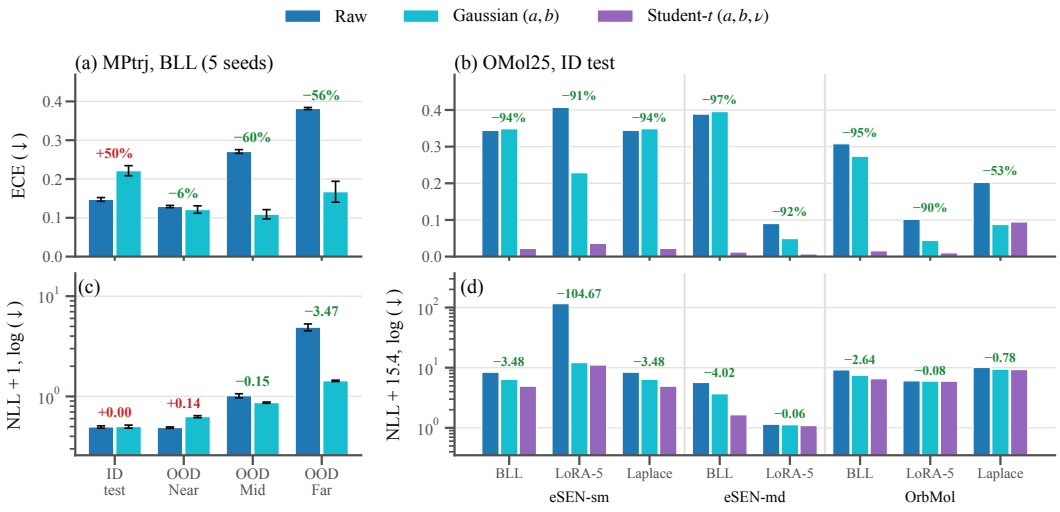

Figure 1: **Recalibration improves calibration and likelihood on MPtrj and OMol25.** Top: expected calibration error (ECE). Bottom: negative log-likelihood, shifted on a log scale to ensure positivity. (a, c) MPtrj BLL, mean ± 1 std over 5 seeds, with the Gaussian (a, b) fit on the in-distribution validation split. Recalibration cuts ECE by 60% on OOD-Mid and 56% on OOD-Far and lowers the OOD-Far NLL from 3.90 to 0.43, at the cost of over-correction on the already well-calibrated ID test. (b, d) OMol25 ID test, every method and backbone, with the Gaussian (a, b) and the Student-t (a, b, ν) fit on Neutral Val by the same procedure. The Gaussian rescaling leaves the BLL ECE almost unchanged ($0.34 \rightarrow 0.35$ on eSEN-sm, $0.39 \rightarrow 0.40$ on eSEN-md), while the Student-t brings it to 0.01–0.02 on all three backbones; MAE and Spearman are unchanged by recalibration. The OMol25 OOD splits are in fig. 5 and the fitted scalars in table 4.

We therefore use it to learn the last-layer prior and noise model, and optionally to fine-tune the representation.

Posterior Predictive. For a test atom with feature vector $\phi_* \in \mathbb{R}^D$, the posterior predictive distribution is $y_* \mid \phi_*, Y, \Phi \sim \mathcal{N}(\phi_*^\top B_n,\ \Sigma_e + u(\phi_*)\Sigma_e)$, with $u(\phi_*) = \phi_*^\top \Lambda_n^{-1} \phi_*$.

When fitting a BLL, we can simultaneously optimize the backbone (either part of all of it) as well as the BLL on the same NLML objective, or we can apply the BLL entirely post-hoc on frozen features. In the former case, which we use for our MPtrj experiments described in section 5, the Φ matrix is changing throughout training, so we optimize with minibatch estimates of the NLML until convergence of the backbone and BLL parameters and then do a final pass over the dataset to recompute $\Phi^\top \Phi$, $\Phi^\top Y$ and $Y^\top Y$. In the latter case, only a single inference pass of the backbone over the dataset is required, then we fit the exact posterior, which is the approach used for the Omol experiments in section 5.

Generalization to Equivariant Architectures. In eSEN (Fu et al., 2025), direct force prediction is done by taking a sum over the D $l = 1$ channels of each atom, $F_i = \Phi_i w$ with $\Phi_i \in \mathbb{R}^{3 \times D}$ and $w \in \mathbb{R}^D$. The BLL fits w and since each weight scales x, y and z components of its channel, three scalar regressions share w. As such, we must split every $l = 1$ vector into its Cartesian components and stack them to create the design matrix $\Phi \in \mathrm{R}^{3N \times D}$ with targets $Y \in \mathrm{R}^{3N \times 1}$. The BLL applies directly to this stacked design matrix. However, two things change: the noise becomes a scalar σ^2, and because an atom's three rows share w, its predictive covariance is the full matrix $\sigma^2 \left(I_3 + \Phi_* \Lambda_n^{-1} \Phi_*^\top\right)$, which is anisotropic and rotates with the structure.

4 Post-hoc Covariance Rescaling

The predictive covariance in eq. (8) decomposes into a constant term Σ_e and an input-dependent term $u(\phi_*)\Sigma_e$. This decomposition is exact under two assumptions of the Bayesian linear model:

(A1) *Linearity.* Given the encoder ϕ_θ, there exist weights B^* such that the noiseless target satisfies $y = \phi_\theta(x)^\top B^*$.

(A2) *Gaussian noise.* Observed targets are corrupted by additive noise $\varepsilon \sim \mathcal{N}(0, \Sigma_e^*)$.

Both assumptions are violated in the MLIP setting. (A2) fails because, conditional on an atomic configuration and a fixed DFT protocol, the target force is deterministic, so $\Sigma_e^* = 0$. (A1) fails because the linear span of ϕ_θ does not contain the exact target force field. Two distinct mechanisms produce this failure:

(M1) *In-distribution misspecification.* A single linear head over ϕ_θ does not fit the targets on the training set; the training residuals are nonzero.

(M2) *Feature collapse on out-of-distribution inputs.* Distinct OOD configurations with different targets are mapped onto similar training features by ϕ_θ, so no linear head over the fixed encoder can fit them simultaneously.

The two mechanisms have different implications for what post-hoc recalibration can correct; sections 4.1 to 4.3 develop this distinction. Throughout, we avoid the labels "aleatoric" and "epistemic" for the two covariance components, since neither retains its standard meaning under (A1)–(A2) violations. We refer to Σ_e as the *residual floor* and to $u(\phi_*)\Sigma_e$ as the *last-layer posterior covariance*.

4.1 Misspecification of the Residual Covariance

We first characterize Σ_e under empirical Bayes. The part of the NLML in eq. (7) depending on Σ_e is $\mathcal{L}(\Sigma_e) = \frac{N}{2}\log|\Sigma_e| + \frac{1}{2}\operatorname{tr}[\Sigma_e^{-1}(R^\top R + B_n^\top \Lambda_0 B_n)]$, with $R = Y - \Phi B_n$, where each row of R is a per-atom force residual after the linear fit. Setting $\partial\mathcal{L}/\partial\Sigma_e^{-1} = 0$ yields the stationary point $\widehat{\Sigma}_e = \frac{1}{N}R^\top R + \frac{1}{N}B_n^\top \Lambda_0 B_n$. The first term is the empirical second moment of the training residuals; the second is the prior shrinkage contribution, which is small once $N \gg D$. Empirical Bayes therefore sets Σ_e to (approximately) the residual covariance of the regularized fit.

Since DFT forces are deterministic given the configuration and protocol, Y contains no stochastic component. The fitted predictions are $\Phi B_n = HY$, where $H = \Phi(\Phi^\top\Phi + \Lambda_0)^{-1}\Phi^\top$ is the regularized hat matrix and the residuals are $R = (I - H)Y$; substituting in eq. (10), we get $\widehat{\Sigma}_e = \frac{1}{N}Y^\top(I-H)^\top(I-H)Y + \frac{1}{N}B_n^\top\Lambda_0 B_n$.

This means that Σ_e does not estimate noise; under (A2) failure, there is none to estimate, and $\widehat{\Sigma}_e$ instead absorbs the in-distribution misspecification (M1). A calibrated residual floor remains useful in the deterministic setting, since it improves predictive likelihood. However, the empirical-Bayes value of Σ_e is unlikely to be appropriately scaled for predictive scoring on held-out data, for two reasons. First, NLML evaluates Σ_e against training residuals and the marginal-likelihood penalty rather than against held-out predictive likelihood. Second, as established in section 4.2, the same Σ_e enters the input-dependent term through the conjugate prior, so its NLML value reflects a compromise between two distinct roles. The post-hoc scalar a introduced in section 4.3 corrects the floor; b corrects the input-dependent term independently.

4.2 The Last-Layer Posterior under a Fixed Encoder

The input-dependent term $u(\phi_*)\Sigma_e$ from eq. (8), with $u(\phi_*) = \phi_*^\top \Lambda_n^{-1}\phi_*$, is the predictive variance contributed by the posterior over B. This posterior places mass only on the function class $\{f_B(x) = \phi_\theta(x)^\top B : B \in \mathbb{R}^{D\times K}\}$ obtained by varying B over the fixed encoder.

Substituting $\Lambda_n = \Phi^\top\Phi + \Lambda_0$, we have $u(\phi_*) = \phi_*^\top(\Phi^\top\Phi + \Lambda_0)^{-1}\phi_*$, which is the squared Mahalanobis distance of ϕ_* from the training feature distribution up to the regularizer. It is small for ϕ_* in directions of high training mass and large for ϕ_* in directions weakly covered by training. The qualitative content of this quantity is geometric and does not depend on (A1) or (A2): a generic predictor is less reliable on inputs whose features lie far from the training support, since fewer data constrain the local fit. The ranking that $u(\phi_*)$ induces over test inputs is therefore robust to (M1) and to (A2) failure.

The magnitude of $u(\phi_*)\Sigma_e$, however, is not robust. The matrix scaling the input-dependent term is the same Σ_e that appears in the noise covariance, since the conjugate prior $B \sim \mathcal{MN}(0, \Lambda_0^{-1}, \Sigma_e)$

ties the two together to admit a closed-form posterior. As a result, Σ_e simultaneously sets the residual floor, where empirical Bayes inflates it to absorb the (M1) residuals (section 4.1), and scales the input-dependent term. There is no reason these two roles require the same magnitude. The post-hoc scalar b introduced in section 4.3 corrects the magnitude of the input-dependent term without altering its geometric ranking, and section 4.4 shows that this correction is equivalent to tempering the conditional posterior over B.

Mechanism (M2) therefore remains a residual source of error that we cannot address. When the encoder collapses OOD inputs onto well-supported feature regions, $u(\phi_*)$ is small on those inputs even though the predictions are unreliable, so the ranking itself is incorrect; no choice of b recovers it. Mechanism (M2) therefore remains a residual source of error after rescaling that we cannot address within our framework.

4.3 Decoupled Covariance Rescaling

The raw BLL covariance from eq. (8) couples the two components through Σ_e, $\mathrm{Cov}(y_* \mid \phi_*, Y, \Phi) = (1 + u(\phi_*))\Sigma_e$. A single scalar T would rescale this to $T(1 + u(\phi_*))\Sigma_e$, preserving the ratio between the floor and the input-dependent component. As established in sections 4.1 and 4.2, however, the empirical-Bayes value of Σ_e reflects a compromise between two roles whose appropriate predictive magnitudes need not coincide. We therefore consider the two-parameter rescaling $\mathrm{Cov}_{a,b}(y_* \mid \phi_*, Y, \Phi) = a\,\Sigma_e + b\,u(\phi_*)\Sigma_e$ with $a, b > 0$, in which a rescales the residual floor and b rescales the input-dependent term independently. Section 4.4 shows that b corresponds to a temperature on the conditional posterior over B. A single global scalar is restricted to the diagonal $a = b$ and cannot reach the optimum when it lies off the diagonal; fig. 4 shows the validation-NLL surface for a representative case.

Heavy-tailed predictive. When the model is misspecified, the residuals are not noise and need not be Gaussian. A few atoms carry large errors while most are small. A Gaussian can only match these errors by widening the predictive for every atom, which is calibrated on average but not across quantiles. We therefore replace it with a multivariate Student-t that has the same mean and the same decoupled scale, $t_\nu\big(\phi_*^\top B_n,\, a\,\Sigma_e + b\,u(\phi_*)\Sigma_e\big)$, and fit (a, b, ν) jointly on the validation set. Its heavier tails absorb the blowup errors without inflating the rest, and the Gaussian rescaling is recovered as $\nu \to \infty$. The choice follows from the conjugate model: an inverse-Wishart prior on Σ_e already gives a Student-t predictive, but with $\sim N$ degrees of freedom, so it is effectively Gaussian. Fitting a small ν post hoc is thus another form of tempering, like b. The scale still grows with $u(\phi_*)$, so the ranking of test inputs is unchanged.

4.4 Connection to Posterior Tempering

The scalar b admits a direct interpretation as a temperature on the conditional posterior over B. Consider tempering the unnormalized joint density by a factor $w > 0$, $p_w(B \mid Y, \Phi) \propto [p(Y \mid B, \Phi, \Sigma_e)\, p(B \mid \Lambda_0, \Sigma_e)]^w$, which parameterizes a controlled widening or narrowing of the posterior over B.

Proposition 1 (Tempering rescales the last-layer covariance)**.** *Tempering the joint density in eq.* (14) *by w changes the posterior precision from Λ_n to $\Lambda_{n,w} = w\Lambda_n$, and consequently $u_w(\phi_*) = \phi_*^\top \Lambda_{n,w}^{-1} \phi_* = w^{-1} u(\phi_*)$. Multiplying $u(\phi_*)\Sigma_e$ by b is therefore equivalent to a temperature $T = b = 1/w$ on the conditional posterior of B.*

Values $b > 1$ correspond to a warmer conditional posterior over B and inflate the input-dependent uncertainty; $b < 1$ corresponds to a cooler posterior. We determine the required temperature post hoc because the spread of B needed for calibrated prediction depends on how much the empirical-Bayes value of Σ_e was inflated to absorb the (M1) residuals.

5 Experiments

We evaluate BLL on two benchmarks chosen to span the deployment regimes of foundation MLIPs. MPtrj is the Materials Project's trajectory dataset (Deng, 2023). We fit UQ methods on this dataset on top of the orb-v3-direct backbone pretrained on OMat24 (Rhodes et al., 2025; Barroso-Luque

et al., 2024), and evaluate on a chemically heterogeneous in-distribution split and under distribution shift, obtained by stratifying held-out structures by their energy above the relaxed state. OMol25 (Levine et al., 2025) evaluates BLL at foundation scale. We use its Neutral subset and three backbones pretrained on the full OMol25 training set, OrbMol, and the equivariant eSEN-sm and eSEN-md (Fu et al., 2025), each used as a frozen feature extractor. All UQ methods are fit on a sub-split of the Neutral-train set , holding out a random ID test split and the top 5% largest molecules by size (OOD Size) and by maximum per-atom force (OOD force). On MPtrj, we compare BLL against the LoRA ensemble (K=3) (Hu et al., 2022), variational-dropout BLIP (Coscia et al., 2025), and SWAG (Maddox et al., 2019). On OMol25 we compare against LoRA ensembles (K=5; four members on eSEN-md) on all three backbones and a last-layer Laplace approximation (Daxberger et al., 2021) on eSEN-sm and OrbMol. We report force MAE, negative log-likelihood (NLL), expected calibration error (ECE), and the Spearman correlation between predicted uncertainty and absolute force error, see appendix section G for details. Additionally, we compare the BLL with an ensemble formed of the 3 pretrained backbone models and evaluate on the neutral validation set, which none of the backbones were trained on.

Recalibration protocol. The post-hoc recalibration is fit on an in-distribution validation set, with the Gaussian (a, b) of eq. (13) on MPtrj and the Student-t (a, b, ν) of section 4.3 on OMol25, with all methods getting the same recalibration protocol as the BLL. For all experiments except the backbone ensemble, we use the validation set for recalibration. For the backbone ensemble we use the ID Test set instead.

Recalibration Performance Figure 1(a, c) isolates the contribution of the post-hoc rescaling on MPtrj. The rescaling produces significant ECE reductions on OOD-Mid and OOD-Far with a $\sim$3.5 nat NLL improvement on OOD-Far, at the cost of mild over-correction on the already well-calibrated ID test split.

On OMol25, the Gaussian rescaling is not enough (fig. 1(b, d)). It improves the BLL NLL (from -7.01 to -9.02 on eSEN-sm) but leaves its ECE almost unchanged ($0.34 \rightarrow 0.35$ on eSEN-sm, $0.39 \rightarrow 0.40$ on eSEN-md), as expected from the heavy residual tails described in section 4.3. The Student-t brings the BLL ECE down to nearly 0 on all three backbones and improves NLL by a further 0.9–2.0 nats.

The fitted scalars (table 4) behave as our analysis predicts. For every BLL, b is of order 10^6 (5.1×10^5 to 1.3×10^6), and the degrees of freedom are $\nu \approx 1.7$–1.8. The eSEN-sm LoRA ensemble is similarly heavy-tailed ($\nu = 1.47$), whereas the OrbMol LoRA-5 ensemble and Laplace have $\nu = 11.8$ and 10.9. On OrbMol the floor multiplier collapses to $a \approx 10^{-19}$.

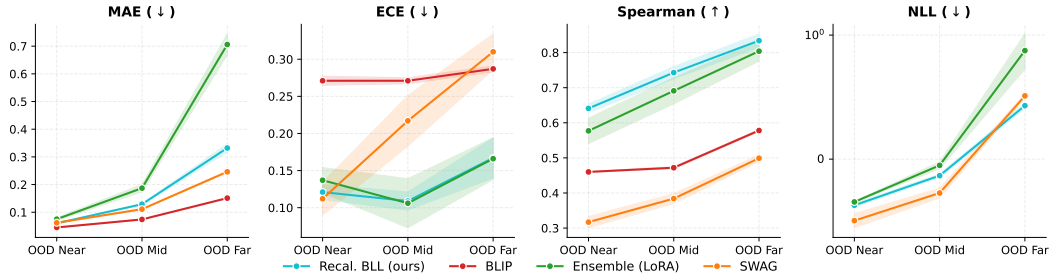

Figure 2: **Performance across increasing distribution shift (OOD Near $\rightarrow$ OOD Mid $\rightarrow$ OOD Far).** Recalibrated BLL maintains the strongest Spearman correlation across all OOD splits and remains competitive on ECE and NLL, while keeping MAE growth substantially below the ensemble. BLIP is omitted from the NLL panel because its values blow up on OOD Mid and OOD Far.

BLL against parameter-efficient ensembles and single-model baselines Figure 2 tracks BLL performance against the LoRA ensemble, BLIP, and SWAG across the three increasingly distorted MPtrj OOD splits. The recalibrated BLL maintains the strongest Spearman correlation throughout, with its ranking power growing as distortion increases. On ECE and NLL, BLL tracks the ensemble

within overlapping error bands, while MAE growth stays well below it (0.332 vs. 0.706 at OOD-Far).

Figure 3 reports the three OMol25 backbones (exact values in table 6). On eSEN-sm, BLL is competitive with or better than the LoRA-5 ensemble. It has lower MAE, ECE and NLL on all three splits, and a higher Spearman correlation on ID test (0.791 vs. 0.698) and OOD size (0.822 vs. 0.755). The ensemble has stronger error ranking only on OOD force (0.822 vs. 0.728). On eSEN-md, the BLL ties LoRA on MAE, but is behind on ECE, Spearman and NLL (fig. 3). On OrbMol the LoRA ensemble is best on every metric and split.

Table 1 compares BLL with the ensemble of the three pretrained backbones themselves on the Neutral Val set, with recalibration done on id test. A BLL on the single strongest backbone, eSEN-md, gives a better predictive distribution than the ensemble, with an NLL of -13.65 against -12.08 and roughly half its MAE (0.0041 vs. 0.0077). Furthermore, the eSEN-sm BLL is better calibrated (0.013 vs. 0.018), with only a single inference pass. The ensemble keeps the better error ranking (Spearman 0.90 against at most 0.80 for the BLLs).

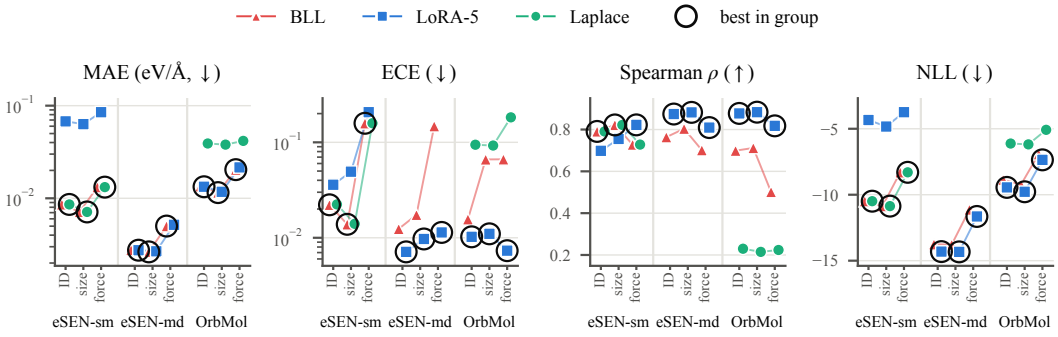

Figure 3: **Uncertainty quantification on OMol25.** Force MAE, ECE, Spearman correlation and NLL of BLL, the LoRA-5 ensemble and the last-layer Laplace approximation on three backbones, each on the ID test, OOD-size and OOD-force splits. MAE and Spearman are unaffected by recalibration. Exact values, including the uncalibrated NLL, are in table 6.

Table 1: BLL on each OMol25 backbone against the ensemble of pretrained backbones (eSEN-sm, eSEN-md and OrbMol)

					Student-t factors		
Model	**MAE (↓)**	**ECE (↓)**	**Spearman (↑)**	**NLL (↓)**	a	b	ν
BLL (eSEN-sm)	0.0101	**0.0131**	0.795	-10.68	5.2×10^{-6}	1.5×10^{6}	1.99
BLL (eSEN-md)	**0.0041**	0.0262	0.747	**-13.65**	6.3×10^{-5}	8.9×10^{5}	2.07
BLL (OrbMol)	0.0151	0.0312	0.661	-8.96	1.5×10^{-18}	4.8×10^{5}	1.99
Backbone ensemble	0.0077	0.0178	**0.901**	-12.08	5.4×10^{-7}	0.598	8.79

The effect of feature rank The ranking of BLL comes from $u(\phi_*)$, a Mahalanobis distance in feature space (section 4.2), and this distance can only separate inputs along directions in which the features actually vary. In table 2, we measure the effective rank and concentration of feature variance of the different backbones, as well as the relative performance of the BLL to the LoRA ensemble. As erank/D (or the number of principal components needed for 95% of the variance) decreases across the three backbones, BLL's Spearman and NLL relative to the ensemble worsen (Table 2), corresponding with a degradation in feature expressivity of the backbone.

6 Discussion

BLL is a capable and cheap post-hoc UQ method for MLIPs. Across a materials and a molecular benchmark, and with every method receiving the same recalibration, BLL is at or near the top

Table 2: Last-layer feature rank of the three foundation backbones, measured on mean-centered Neutral Val features (D is the feature dimension); "95%" is the number of principal components needed for 95% of the feature variance; dead dimensions have zero training variance. The last two columns compare BLL with the LoRA-5 ensemble on the ID test split (table 6)

							BLL − LoRA-5 (ID test)	
Backbone	D	erank	erank$/D$	95%	99%	dead	ΔSpearman	ΔNLL
eSEN-sm	128	74.6	0.583	50	87	13	+0.093	-6.15
eSEN-md	128	61.0	0.477	23	70	7	-0.110	+0.56
OrbMol	256	44.3	0.173	9	17	0	-0.178	+0.58

performing models across performance metrics. On MPtrj, it gives the best error ranking on every OOD split while matching or beating the LoRA ensemble on ECE and NLL (section 5). On Omol25, it beats the LoRA ensemble on the esen-sm backbone on MAE, ECE and NLL on all three splits, and on eSEN-md it yields a better predictive distribution than an ensemble of all three pretrained backbones. In the frozen-feature regime, this comes at very low cost: the fit is one pass over the training set to accumulate sufficient statistics followed by a closed-form posterior, and prediction needs a single backbone pass, versus K fine-tunes and K passes for an ensemble. Furthermore, the BLL mean is a ridge refit of the backbone's own last linear layer on the same features, so at $N \approx 10^9$ it retains almost all of the accuracy from the pretrained backbone (table 5), whereas the LoRA ensembles change the mean, matching BLL on OrbMol but degrading MAE on eSEN-sm.

Recalibration is essential. At the scale of data in our experiments, the BLL raw posterior is extremely overconfident, resulting in an ECE of 0.3–0.4 on every backbone (section 5). We outline an effective correction, but as we show, it is not specific to the BLL. The split into a residual floor and an input-dependent term exists in any UQ model with aleatoric and epistemic parts, and the same fit lowered the held-out NLL across all methods and data splits, however the BLL tends to gain most from it because it addresses the failures (A1) and (A2). Empirically we observe across all the recalibration fits that the a and b parameters behave as predicted for the BLL. Σ_e is always reduced, particularly on Omol25 ($a < 10^{-3}$ on eSEN, 10^{-19} on OrbMol). By contrast, b is very large ($b \sim 10^6$) which we interpret as a posterior temperature (proposition 1). Two factors are absorbed into b. First, atoms in the same structure have correlated errors so the BLL overcounts evidence. Second, the last layer posterior carries no uncertainty about the backbone itself, the cost of tractability.

Requirements of the features Our recalibration leaves the MAE and ranking unchanged. Ranking comes entirely from the leverage $u(\phi^*)$, which separates inputs only along directions in which the features vary. The effective rank of the features is therefore a natural candidate for what limits BLL, and the three OMol25 pretrained backbones are consistent with it. OrbMol uses the smallest fraction of its feature space (erank$/D = 0.17$), and BLL's ranking falls furthest behind the ensemble. eSEN-md has the lower effective rank than eSEN-sm and also a weaker spearman correlation (section 5). With three backbones, this is suggestive rather than established, but erank$/D$ is computed from the backbone alone and is a cheap pre-check before fitting anything. Raising it is a question of how the backbone is trained. Regularizers that encourage feature diversity, such as the bi-Lipschitz constraints of SNGP and DUQ (Liu et al., 2020; van Amersfoort et al., 2020), are compatible with BLL but we leave this to future work.

Limitations. Our formulation, including the equivariant setup, targets direct force heads; it does not address conservative heads with $F = -\nabla E$. The OMol25 OOD splits are OOD for the UQ heads but not for the backbones, which were pretrained on all of OMol25. We also do not directly compare to full deep ensembles, but the BLL is better situated as a post-hoc fitting method, and deep ensembles are prohibitively expensive to train for foundation models.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of proofs, help develop theoretical models or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses,

design or provide feedback on research methodology or experiments, implement methods, clean and reformat datasets and interpret results. We have not used generative AI tools for developing the core research topic of this work, and generating synthetic data, assisting with translation and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for creating scientific figures, suggesting experimental parameters, creating and editing software, drafting parts of the research paper, summarizing literature, identify gaps, brainstorm, editing and identifying relevant literature. We have reviewed all AI-assisted work. By verifying correctness of all mathematical claims and ensuring code and implementation correctness to the best of our abilities. We take respon- sibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Luis Barroso-Luque, Muhammed Shuaibi, Xiang Fu, Brandon M. Wood, Misko Dzamba, Meng Gao, Ammar Rizvi, C. Lawrence Zitnick, and Zachary W. Ulissi. Open materials 2024 (omat24) inorganic materials dataset and models, 2024. URL https://arxiv.org/abs/2410.12771.

Albert P. Bartók, Mike C. Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical Review Letters*, 104 (13):136403, 2010.

Albert P. Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18):184115, 2013.

P. G. Bissiri, C. C. Holmes, and S. G. Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130, 2016.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *International Conference on Machine Learning*, 2015.

Kamal Choudhary et al. Ai-driven expansion and application of the alexandria database. *arXiv preprint arXiv:2512.09169*, 2025/2026. URL https://doi.org/10.48550/arXiv.2512.09169.

Anders S. Christensen and Anatole Von lilienfeld. Revised MD17 dataset (rMD17). 7 2020. doi: 10.6084/m9.figshare.12672038.v4. URL https://figshare.com/articles/dataset/Revised_MD17_dataset_rMD17_/12672038.

Dario Coscia, Pim de Haan, and Max Welling. BLIPs: Bayesian learned interatomic potentials, 2025. URL https://arxiv.org/abs/2508.14022.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace Redux – effortless Bayesian deep learning. In *Advances in Neural Information Processing Systems*, 2021.

Bowen Deng. Materials Project Trajectory (MPtrj) Dataset. 7 2023. doi: 10.6084/m9.figshare.23713842.v2. URL https://figshare.com/articles/dataset/Materials_Project_Trjectory_MPtrj_Dataset/23713842.

Xiang Fu, Brandon M. Wood, Luis Barroso-Luque, Daniel S. Levine, Meng Gao, Misko Dzamba, and C. Lawrence Zitnick. Learning smooth and expressive interatomic potentials for physical property prediction, 2025. URL https://arxiv.org/abs/2502.12147.

Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, 2011.

Peter Grünwald and Thijs van Ommen. Inconsistency of bayesian inference for misspecified linear models, and a proposal for repairing it, 2018. URL https://arxiv.org/abs/1412.3730.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Pavel Izmailov, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. What are Bayesian neural network posteriors really like? In *International Conference on Machine Learning*, 2021.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being Bayesian, even just a bit, fixes overconfidence in ReLU networks. In *International Conference on Machine Learning*, 2020.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

Kenneth L. Lange, Roderick J. A. Little, and Jeremy M. G. Taylor. Robust statistical modeling using the t distribution. *Journal of the American Statistical Association*, 84(408):881–896, 1989.

Daniel S. Levine, Muhammed Shuaibi, Evan Walter Clark Spotte-Smith, Michael G. Taylor, et al. The open molecules 2025 (OMol25) dataset, evaluations, and models, 2025.

Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. EquiformerV2: Improved equivariant transformer for scaling to higher-degree representations. In *International Conference on Learning Representations*, 2024.

Jeremiah Z. Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *Advances in Neural Information Processing Systems*, 2020.

David J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning, 2019. URL https://arxiv.org/abs/1902.02476.

Radford M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, 1996.

Sebastian W. Ober and Carl Edward Rasmussen. Benchmarking the neural linear model for regression, 2019.

Cameron Owen, Steven Torrisi, Yu Xie, Simon Batzner, Kyle Bystrom, Jennifer Coulter, Albert Musaelian, Lixin Sun, and Boris Kozinsky. Complexity of many-body interactions in transition metals via machine-learned force fields from the tm23 data set. *npj Computational Materials*, 10 (1):105, May 2024. doi: 10.1038/s41524-024-01264-z. URL https://www.nature.com/articles/s41524-024-01264-z.

Theodore Papamarkou, Maria Skoularidou, Konstantina Palla, Laurence Aitchison, Julyan Arbel, David Dunson, Maurizio Filippone, Vincent Fortuin, Philipp Hennig, José Miguel Hernández-Lobato, et al. Position: Bayesian deep learning is needed in the age of large-scale AI. In *International Conference on Machine Learning*, 2024.

Tim Pearce, Felix Leibfried, and Alexandra Brintrup. Uncertainty in neural networks: Approximately Bayesian ensembling. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Ghanshyam Pilania, James E. Gubernatis, and Turab Lookman. Multi-fidelity machine learning models for accurate bandgap predictions of solids. *Computational Materials Science*, 129:156–163, 2017.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Benjamin Rhodes, Sander Vandenhaute, Vaidotas Šimkus, James Gin, Jonathan Godwin, Tim Duignan, and Mark Neumann. Orb-v3: atomistic simulation at scale, 2025. URL https://arxiv.org/abs/2504.06231.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable Laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *European Signal Processing Conference*, 2007.

Kristof T. Schütt, Pieter-Jan Kindermans, Huziel E. Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, 2017.

Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Prabhat, and Ryan P. Adams. Scalable Bayesian optimization using deep neural networks. In *International Conference on Machine Learning*, 2015.

Aik Rui Tan, Shingo Urata, Samuel Goldman, Johannes C B Dietschreit, and Rafael Gómez-Bombarelli. Single-model uncertainty quantification in neural network potentials does not consistently outperform model ensembles. *Npj Comput. Mater.*, 9(1), December 2023.

Stephan Thaler, Gregor Doehner, and Julija Zavadlav. Scalable Bayesian uncertainty quantification for neural network potentials: Promise and pitfalls. *Journal of Chemical Theory and Computation*, 2023.

Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, 2020.

Joost van Amersfoort, Lewis Smith, Andreas Jesson, Oscar Key, and Yarin Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. In *arXiv preprint arXiv:2102.11409*, 2021.

Joe Watson, Jihao Andreas Lin, Pascal Klink, Joni Pajarinen, and Jan Peters. Latent derivative Bayesian last layer networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *Advances in Neural Information Processing Systems*, 2020.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 370–378, 2016a.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, 2016b.

Brandon M. Wood, Misko Dzamba, Xiang Fu, Meng Gao, Muhammed Shuaibi, Luis Barroso-Luque, Kareem Abdelmaqsoud, Vahe Gharakhanyan, John R. Kitchin, Daniel S. Levine, Kyle Michel, Anuroop Sriram, Taco Cohen, Abhishek Das, Ammar Rizvi, Sushree Jagriti Sahoo, Zachary W. Ulissi, and C. Lawrence Zitnick. Uma: A family of universal models for atoms, 2025. URL https://arxiv.org/abs/2506.23971.

A Equations of the Main Text

For reference, this section collects the equations of the main text in numbered form, in the order in which they appear. References to equations in the main text point here.

Linear force head (section 3).

$$Y = \Phi B + E, \qquad B \in \mathbb{R}^{D\times K}. \tag{1}$$

Likelihood.

$$Y \mid B, \Phi, \Sigma_e \sim \mathcal{MN}(\Phi B,\ I_N,\ \Sigma_e), \tag{2}$$

Predictive covariance between two atoms (footnote in section 3).

$$\mathrm{Cov}(y_i, y_j \mid Y, \Phi) = \delta_{ij}\Sigma_e + \left(\phi_i^\top \Lambda_n^{-1} \phi_j\right) \Sigma_e. \tag{3}$$

Prior.

$$B \mid \Lambda_0, \Sigma_e \sim \mathcal{MN}(0,\ \Lambda_0^{-1},\ \Sigma_e), \tag{4}$$

Posterior.

$$B \mid Y, \Phi, \Lambda_0, \Sigma_e \sim \mathcal{MN}(B_n,\ \Lambda_n^{-1},\ \Sigma_e), \qquad \Lambda_n = \Phi^\top \Phi + \Lambda_0, \qquad B_n = \Lambda_n^{-1} \Phi^\top Y. \tag{5}$$

Marginal likelihood.

$$p(Y \mid \Phi, \Lambda_0, \Sigma_e) = \int p(Y \mid B, \Phi, \Sigma_e)\, p(B \mid \Lambda_0, \Sigma_e)\, \mathrm{d}B. \tag{6}$$

Negative log marginal likelihood (derived in section D).

$$\begin{aligned} \mathcal{L}_{\mathrm{NLML}} = \frac{NK}{2} \log(2\pi) + \frac{N}{2} \log |\Sigma_e| + \frac{K}{2} \left(\log |\Lambda_n| - \log |\Lambda_0| \right) \\ + \frac{1}{2} \operatorname{tr} \left[\Sigma_e^{-1} \left((Y - \Phi B_n)^\top (Y - \Phi B_n) + B_n^\top \Lambda_0 B_n \right) \right]. \end{aligned} \tag{7}$$

Posterior predictive.

$$y_* \mid \phi_*, Y, \Phi \sim \mathcal{N}\left(\phi_*^\top B_n,\ \Sigma_e + u(\phi_*) \Sigma_e \right), \qquad u(\phi_*) = \phi_*^\top \Lambda_n^{-1} \phi_*. \tag{8}$$

Part of the NLML that depends on Σ_e (section 4.1).

$$\mathcal{L}(\Sigma_e) = \frac{N}{2} \log |\Sigma_e| + \frac{1}{2} \operatorname{tr} \left[\Sigma_e^{-1} \left(R^\top R + B_n^\top \Lambda_0 B_n \right) \right], \qquad R = Y - \Phi B_n, \tag{9}$$

Empirical-Bayes stationary point of Σ_e.

$$\widehat{\Sigma}_e = \tfrac{1}{N} R^\top R + \tfrac{1}{N} B_n^\top \Lambda_0 B_n. \tag{10}$$

The same stationary point for deterministic targets.

$$\widehat{\Sigma}_e = \tfrac{1}{N} Y^\top (I - H)^\top (I - H) Y + \tfrac{1}{N} B_n^\top \Lambda_0 B_n. \tag{11}$$

Decoupled covariance rescaling (section 4.3).

$$\mathrm{Cov}_{a,b}(y_* \mid \phi_*, Y, \Phi) = a\, \Sigma_e + b\, u(\phi_*) \Sigma_e, \qquad a, b > 0, \tag{12}$$

Validation objective for (a, b).

$$(a^\star, b^\star) = \arg\min_{a,b>0} \sum_{(x_i, y_i) \in \mathcal{D}_{\mathrm{val}}} - \log \mathcal{N}\left(y_i;\ \phi_i^\top B_n,\ a \Sigma_e + b\, u(\phi_i) \Sigma_e \right), \tag{13}$$

Tempered joint density (section 4.4).

$$p_w(B \mid Y, \Phi) \propto \left[p(Y \mid B, \Phi, \Sigma_e)\, p(B \mid \Lambda_0, \Sigma_e) \right]^w, \tag{14}$$

Student-t predictive (section 4.3).

$$y_* \mid \phi_*, Y, \Phi \sim t_\nu \left(\phi_*^\top B_n,\ a\, \Sigma_e + b\, u(\phi_*) \Sigma_e \right), \tag{15}$$

Effective rank (section 5).

$$\mathrm{erank}(\Phi) = \exp\Big(- \sum_{k=1}^{D} p_k \log p_k \Big), \tag{16}$$

B Proof of proposition 1

Proof. of The terms in the unnormalized log joint quadratic in B are, up to constants, $-\frac{1}{2} \operatorname{tr}\left[\Sigma_e^{-1} B^\top (\Phi^\top \Phi + \Lambda_0) B \right]$. Multiplying the log joint by w scales $(\Phi^\top \Phi + \Lambda_0)$ by w, so the tempered posterior precision is $\Lambda_{n,w} = w(\Phi^\top \Phi + \Lambda_0) = w \Lambda_n$. The linear-in-$B$ terms also scale by w, so the posterior mean is unchanged: $B_{n,w} = (w \Lambda_n)^{-1} (w \Phi^\top Y) = B_n$. Substituting yields $u_w(\phi_*) = w^{-1} u(\phi_*)$. □

Algorithm 1 Bayesian Linear Last Layer for MLIPs: training, calibration, and prediction.

Require: Pretrained backbone ϕ_θ; train/val/test sets $\mathcal{D}_{\mathrm{tr}}, \mathcal{D}_{\mathrm{val}}, \mathcal{D}_{\mathrm{te}}$

procedure TRAIN($\phi_\theta, \mathcal{D}_{\mathrm{tr}}$)
  Initialize hyperparameters Λ_0, Σ_e (and optionally θ)
  for each minibatch $\mathcal{B} \subset \mathcal{D}_{\mathrm{tr}}$ **do**
    Compute per-atom features $\Phi_\mathcal{B}$ via ϕ_θ and stack force targets $Y_\mathcal{B}$
    Form $\Phi_\mathcal{B}^\top \Phi_\mathcal{B}$ and $\Phi_\mathcal{B}^\top Y_\mathcal{B}$; compute Λ_n, B_n via equation 5
    Evaluate $\mathcal{L}_{\mathrm{NLML}}$ on $\mathcal{B}$ via equation 7
    Update $\{\Lambda_0, \Sigma_e\}$ (and optionally θ) by gradient descent on $\mathcal{L}_{\mathrm{NLML}}$
  end for
  Recompute $\Phi^\top \Phi$, $\Phi^\top Y$ on the full $\mathcal{D}_{\mathrm{tr}}$
  Compute final posterior (Λ_n, B_n) via equation 5
  return $\theta, \Lambda_0, \Sigma_e, \Lambda_n, B_n$
end procedure

procedure CALIBRATE($\theta, \Lambda_n, B_n, \Sigma_e, \mathcal{D}_{\mathrm{val}}$)
  for each $(x_i, y_i) \in \mathcal{D}_{\mathrm{val}}$ **do**
    Compute $\phi_i = \phi_\theta(x_i)$ and $u(\phi_i) = \phi_i^\top \Lambda_n^{-1} \phi_i$
  end for
  Solve equation 13 for $(a^\star, b^\star)$ on $(\alpha, \beta) = (\log a, \log b)$ (Student-t: also $\nu^\star$, section 4.3; optimizers per dataset in section E)
  return $(a^\star, b^\star)$
end procedure

procedure PREDICT($x_*; \theta, B_n, \Lambda_n, \Sigma_e, a^\star, b^\star$)
  $\phi_* \leftarrow \phi_\theta(x_*)$; $u(\phi_*) \leftarrow \phi_*^\top \Lambda_n^{-1} \phi_*$
  Mean: $\hat{y}_* \leftarrow \phi_*^\top B_n$ ▷ from equation 8
  Covariance: $\widehat{\mathrm{Cov}}(y_*) \leftarrow a^\star \, \Sigma_e + b^\star \, u(\phi_*) \, \Sigma_e$ ▷ from equation 12
  return $(\hat{y}_*, \widehat{\mathrm{Cov}}(y_*))$
end procedure

C REPRODUCIBILITY

Code, model checkpoints, and trained recalibration parameters will be released upon acceptance. All datasets used are public; access details and licenses are listed in Appendix F. Hyperparameters, optimizer settings, and seeds are documented in Appendix E.

D DERIVATION OF EMPIRICAL BAYES WITH LEARNED GLOBAL NOISE

D.1 INITIAL SETUP

We use a Bayesian Linear Last Layer to perform Bayesian regression on embeddings from Orb-v3 Rhodes et al. (2025) to atomic forces, allowing for direct force prediction as well as uncertainty estimations for these predictions.
Multivariate Regression Setup:

$$Y = \Phi B + E \tag{17}$$

$$E \sim \mathcal{N}(0, \Sigma_e) \tag{18}$$

Where $Y \in \mathbb{R}^{N \times K}$ are the targets, $\Phi \in \mathbb{R}^{N \times D}$ are the features from the network, $B \in \mathbb{R}^{D \times K}$ are the weights of the final layer and $E \in \mathbb{R}^{N \times K}$ is the noise. N is the number of data points, K is the output dimension (for forces in XYZ K is 3) and D is the dimension of the feature embeddings.
We assume each row $y_n \in \mathbb{R}^{1 \times K}$ is independently distributed as

$$y_n \mid B, \Sigma_e \sim \mathcal{N}(\phi_n B, \Sigma_e) \tag{19}$$

So the joint likelihood over all rows follows a Matrix Normal Distribution.

$$p(Y \mid B, \Phi, \Sigma_e) = \mathcal{MN}(Y \mid \Phi B, \; I_N, \; \Sigma_e) \tag{20}$$

Note on the Matrix Normal: If $X \sim \mathcal{MN}(M, U, V)$ with $X \in \mathbb{R}^{n\times p}$, $U \in \mathbb{R}^{n\times n}$ (row covariance), $V \in \mathbb{R}^{p\times p}$ (column covariance), the density is:

$$(2\pi)^{-np/2}|V|^{-n/2}|U|^{-p/2}\exp\Big(-\frac{1}{2}\mathrm{tr}[V^{-1}(X-M)^T U^{-1}(X-M)]\Big) \tag{21}$$

And so the density can be written as:

$$p(Y \mid B, \Phi, \Sigma_e) = \frac{1}{(2\pi)^{NK/2}\,|\Sigma_e|^{N/2}}\exp\Big(-\tfrac{1}{2}\ \mathrm{tr}\big[\Sigma_e^{-1}(Y-\Phi B)^T(Y-\Phi B)\big]\Big) \tag{22}$$

D.2 CONJUGATE MATRIX NORMAL PRIOR ON B

We place a Matrix Normal prior on B with zero mean, a diagonal row precision Λ_0 and column covariance Σ_e (same as noise for conjugacy)

$$B \mid \Lambda_0, \Sigma_e \sim \mathcal{MN}(0, \Lambda_0^{-1}, \Sigma_e) \tag{23}$$

So

$$p(B \mid \Lambda_0, \Sigma_e) = \frac{|\Lambda_0|^{K/2}}{(2\pi)^{DK/2}\,|\Sigma_e|^{D/2}}\exp\Big(-\tfrac{1}{2}\ \mathrm{tr}\big[\Sigma_e^{-1}B^T\Lambda_0\,B\big]\Big) \tag{24}$$

D.3 MARGINAL LIKELIHOOD AND INTEGRATING OUT B

The Marginal Likelihood is given by:

$$p(Y \mid \Phi, \Lambda_0, \Sigma_e) = \int p(Y \mid B, \Phi, \Sigma_e)\, p(B \mid \Lambda_0, \Sigma_e)\ \mathrm{d}B \tag{25}$$

$$= \int \frac{1}{(2\pi)^{NK/2}\,|\Sigma_e|^{N/2}}\exp\Big(-\tfrac{1}{2}\ \mathrm{tr}\big[\Sigma_e^{-1}(Y-\Phi B)^T(Y-\Phi B)\big]\Big) * \frac{|\Lambda_0|^{K/2}}{(2\pi)^{DK/2}\,|\Sigma_e|^{D/2}}\exp\Big(-\tfrac{1}{2}\ \mathrm{tr}\big[\Sigma_e^{-1}B^T\Lambda_0\,B\big]\Big)$$

$$= \int \frac{|\Lambda_0|^{K/2}}{(2\pi)^{(N+D)K/2}\,|\Sigma_e|^{(N+D)/2}}\exp\Big(-\tfrac{1}{2}\ \mathrm{tr}\Big[\Sigma_e^{-1}((Y-\Phi B)^T(Y-\Phi B) + B^T\Lambda_0\,B)\Big]\Big)$$

The argument of the exponential is:

$$-\frac{1}{2}\,\mathrm{tr}\Big[\Sigma_e^{-1}(\underbrace{(Y-\Phi B)^T(Y-\Phi B)}_{\text{likelihood}} + \underbrace{B^T\Lambda_0\,B}_{\text{prior}})\Big]$$

We need to integrate out B, so first we expand and collect the terms in B. First we expand the likelihood term:

$$(Y-\Phi B)^T(Y-\Phi B) = Y^TY - Y^T\Phi\,B - B^T\Phi^TY + B^T\Phi^T\Phi\,B$$

Add the prior term back in:

$$Q(B) = Y^TY - Y^T\Phi\,B - B^T\Phi^TY + B^T\big(\Phi^T\Phi + \Lambda_0\big)B$$

Define the posterior precision

$$\Lambda_n = \Phi^T\Phi + \Lambda_0 \tag{26}$$

and the posterior mean given by the standard MLE solution

$$B_n = \Lambda_n^{-1}\Phi^T Y \tag{27}$$

$$\implies \Lambda_n B_n = \Phi^T Y \tag{28}$$

Substituting into the terms of $Q(B)$:

$$Y^T Y - B_n^T \Lambda_n^T B - B^T \Lambda_n B_n + B^T \Lambda_n B$$

$$= Y^T Y + (B - B_n)^T \Lambda_n (B - B_n) - B_n^T \Lambda_n B_n$$

$$Q(B) = \underbrace{Y^T Y - B_n^T \Lambda_n B_n}_{\text{constant w.r.t. } B} + \underbrace{(B - B_n)^T \Lambda_n (B - B_n)}_{\text{integrates out}} \tag{29}$$

We only integrate over the $(B - B_n)^T \Lambda_n (B - B_n)$ term since other terms are constant w.r.t. B

$$\int \exp\Big(-\tfrac{1}{2}\operatorname{tr}\big[\Sigma_e^{-1}(B - B_n)^T \Lambda_n (B - B_n)\big]\Big)\,\mathrm{d}B$$

This is the unnormalized integral of a $\mathcal{MN}(B_n, \Lambda_n^{-1}, \Sigma_e)$ density which evaluates to:

$$(2\pi)^{DK/2}\,|\Sigma_e|^{D/2}\,|\Lambda_n^{-1}|^{K/2} = (2\pi)^{DK/2}\,|\Sigma_e|^{D/2}\,|\Lambda_n|^{-K/2}$$

D.4 ASSEMBLING MARGINAL LIKELIHOOD

$$p(Y \mid \Phi, \Lambda_0, \Sigma_e) = \underbrace{\frac{1}{(2\pi)^{NK/2}|\Sigma_e|^{N/2}}}_{\text{from likelihood}} \times \underbrace{\frac{|\Lambda_0|^{K/2}}{(2\pi)^{DK/2}|\Sigma_e|^{D/2}}}_{\text{from prior}} \times \underbrace{(2\pi)^{DK/2}|\Sigma_e|^{D/2}|\Lambda_n|^{-K/2}}_{\text{from integral}}$$

$$\times\ \exp\Big(-\tfrac{1}{2}\operatorname{tr}\big[\Sigma_e^{-1}(Y^T Y - B_n^T \Lambda_n B_n)\big]\Big)$$

After canceling out, we are left with:

$$\boxed{p(Y \mid \Phi, \Lambda_0, \Sigma_e) = \frac{|\Lambda_0|^{K/2}}{(2\pi)^{NK/2}\,|\Sigma_e|^{N/2}\,|\Lambda_n|^{K/2}}\ \exp\Big(-\tfrac{1}{2}\operatorname{tr}\big[\Sigma_e^{-1}(Y^T Y - B_n^T \Lambda_n B_n)\big]\Big)} \tag{30}$$

Taking the $-\log$ we get our Negative Log Marginal Likelihood which can be used for our loss function:

$$\mathcal{L} = \frac{NK}{2}\log(2\pi) + \frac{N}{2}\log|\Sigma_e| + \frac{K}{2}\big(\log|\Lambda_n| - \log|\Lambda_0|\big) + \frac{1}{2}\operatorname{tr}\big[\Sigma_e^{-1}(Y^T Y - B_n^T \Lambda_n B_n)\big] \tag{31}$$

The derivation could end here but we further rewrite the loss for better interpretability.

D.5 DECOMPOSING LOSS TO IMPROVE INTERPRETABILITY

We can rewrite the data-dependent term in two forms,
Form A:

$$\operatorname{tr}\big[\Sigma_e^{-1}(Y^T Y - B_n^T \Lambda_n B_n)\big]$$

Form B (residuals + prior penalty):

$$(Y - \Phi B_n)^T (Y - \Phi B_n) = Y^T Y - Y^T \Phi B_n - B_n^T \Phi^T Y + B_n^T \Phi^T \Phi B_n$$

$$= Y^T Y - B_n^T \Lambda_n B_n - B_n^T \Lambda_n B_n + B_n^T (\Lambda_n - \Lambda_0) B_n$$

$$= \underbrace{Y^T Y - B_n^T \Lambda_n B_n}_{\text{Form A}} - B_n^T \Lambda_0 B_n$$

D.6 FINAL NLML

For gradient descent, the constant $\frac{NK}{2} \log 2\pi$ can be dropped, however, we keep it so we can report the NLML.

$$\boxed{\mathcal{L} = \frac{NK}{2} \log 2\pi + \frac{N}{2} \log |\Sigma_e| + \frac{K}{2} \big(\log |\Lambda_n| - \log |\Lambda_0| \big) + \frac{1}{2} \operatorname{tr} \big[\Sigma_e^{-1} \big((Y - \Phi B_n)^T (Y - \Phi B_n) + B_n^T \Lambda_0 B_n \big) \big]} \tag{32}$$

The final loss only depends on Σ_e and Λ_0, so these are the two parameters we learn using gradient descent during training. Additionally, you can also learn the parameters of the feature model simultaneously.

E EXPERIMENTS

All MPtrj and rMD17 models share the orb-direct-20 backbone (`https://github.com/orbital-materials/orb-models`) pretrained on OMat24, released under the Apache 2.0 license (`https://www.apache.org/licenses/LICENSE-2.0`). Deep methods use a batch size of 64; the exact GP baselines on rMD17 use closed-form inference and do not require batched training.

For rMD17, we train only on ethanol, malonaldehyde, toluene, uracil, and benzene (the five smallest molecules) since the remaining trajectories are too large for tractable exact GP regression. OOD evaluation is performed across all non-training molecules in the rMD17 dataset.

Training BLL. This paragraph describes the MPtrj and rMD17 runs, which train both the BLL and the layer just before the BLL jointly using the NLL objective. We remove the final projection layer from the orb-v3 force head and use the resulting penultimate embeddings as the BLL features ϕ. The backbone is frozen during training; only the penultimate force projection and the priors Λ_0, Σ_e are updated by gradient descent on the NLML objective (eq. 7). We initialize Σ_e and Λ_0 as identity matrices and optimize with Adam at learning rate 1×10^{-3}, with early-stopping patience 10 epochs. ZBL forces (as computed by the Orb model) are subtracted from the targets during training; on rMD17 we additionally convert units to eV/Å. We report results averaged over 5 seeds on MPtrj and 3 seeds per training molecule on rMD17. The recalibration parameters $(a^\star, b^\star)$ are fit via Nelder–Mead on the same validation split used for early stopping, optimizing $(\log a, \log b)$. MPtrj training typically converges in $\sim$70 epochs at $\sim$130 seconds per epoch on 4 A100s; rMD17 epochs run in a few seconds on an RTX 8000 but typically require 200–300 epochs to converge — its total compute is a small fraction of MPtrj's.

Training Ensemble LoRA. Each ensemble member reinitializes the last projection layer of the force head and adds a variance head that predicts log-variances. The variance head copies the force-head architecture with all weights and biases set to zero, so that the initial predicted variance is unity. We inject LoRA adapters (rank 8, $\alpha = 16$) into the message-passing MLPs. Training minimizes Gaussian NLL using Adam with learning rate 1×10^{-3} for the LoRA matrices and 5×10^{-4} for the force and variance heads, with early-stopping patience 10 epochs. On MPtrj we train 9 members (seeds 0–8) and form three ensembles of 3 members each ($\{0,1,2\}, \{3,4,5\}, \{6,7,8\}$); on rMD17 we train one ensemble of 3 members per training molecule (seeds 0, 1, 2). MPtrj members converged in $\sim$73 epochs at $\sim$184 seconds per epoch on 4 A100s; rMD17 training, as with BLL, accounts for a small fraction of total compute.

Training BLIP. We use the training script provided by the authors (`https://github.com/dario-coscia/blip`) with default hyperparameters and an added early-stopping patience of 10 epochs. MPtrj results are averaged over seeds 1, 2, 3. BLIP was tested only on MPtrj.

Training SWAG. We first train the orb-v3 force head on MSE using Adam at learning rate 1×10^{-3}, with the backbone frozen. After 10 epochs without improvement we reload the best checkpoint and continue training for 40 additional epochs at the same learning rate. SWAG retains the last 20 weight deltas to construct the low-rank covariance and averages over all 40 SWAG-collection epochs to estimate the mean. Reported numbers are averaged over seeds 0, 1, 2, with 10 posterior samples drawn at inference. SWAG was tested only on MPtrj.

OMol25 models. The OMol25 runs use the exact posterior fit of the BLL. For eSEN-sm and eSEN-md we take the $l = 1$ vector features that enter the direct-force head ($D = 128$, three rows per atom, section I); for OrbMol we take the output of the force-head MLP ($D = 256$, one row per atom, $K = 3$), with a full learned 3×3 Σ_e as in section 3. $\Phi^\top\Phi$, $\Phi^\top Y$ and $Y^\top Y$ are accumulated in float64 over all $54{,}980{,}568$ atoms of the training sub-split, and (Λ_0, Σ_e) are then fit by empirical Bayes on these statistics: we minimize the exact NLML equation 7 with L-BFGS (strong-Wolfe line search, up to 500 outer iterations, patience 20 on Neutral Val NLML), starting from $\lambda_d = 1$ and $\Sigma_e = 10^{-4}I$ in normalized units, with Σ_e parameterized by its Cholesky factor. The prior precisions stay at their initialization (median $\lambda_d = 1.0000$), and the median of $\lambda_d/(\Phi^\top\Phi)_{dd}$ is 1.8×10^{-8} for OrbMol, 6.9×10^{-8} for eSEN-sm and 5.2×10^{-8} for eSEN-md.

The LoRA-5 ensembles have $K{=}5$ members (seeds 0–4, which set both the adapter initialization and the data order) with rank-8 adapters ($\alpha = 16$). On OrbMol the adapters are placed on every linear layer of the node and edge MLPs of the message-passing stacks, and a per-component log-variance head cloned from the force-head MLP predicts a diagonal covariance; on eSEN-sm they are placed on the SO(2)-convolution and atom-wise feed-forward linear layers, and an equivariant head predicts a full 3×3 covariance. The force head is trained together with the adapters. We use Adam with learning rate 10^{-3} for the adapters and 5×10^{-4} for the heads, 500 warmup steps, gradient clipping at 10, batches of 64 structures, and at most 60k steps, validating every 2000 steps on a fixed subset of Neutral Val (at most 2M atoms) with patience 10 and keeping the checkpoint with the best validation NLL.

The last-layer Laplace approximation uses the same OrbMol features with an appended bias column ($D = 257$, flat prior on the bias). Its Hessian is the exact generalized Gauss–Newton matrix of the linear readout, $\Sigma_e^{-1} \otimes (\Phi^\top \Phi + \Lambda_0)$, with the mean fixed at the MAP weights and a scalar prior precision tuned by the Laplace marginal likelihood (Daxberger et al., 2021). All OMol25 methods are recalibrated on Neutral Val with the Student-t fit of section H.

Computational resources. Experiments were distributed across NVIDIA A100 (80GB), RTX 8000 (48GB), and L40S (48GB) GPUs, using 4 GPUs per training run with 8–16 CPUs per GPU and 256GB of system memory. Per-method training costs on a single 4-A100 instance are reported in Table 3; BLIP was trained on RTX 8000 hardware, and its A100-equivalent figure was obtained by scaling its measured wall-clock by a factor of 3.51, derived from 2 matched LoRA-ensemble runs across both architectures. No single seed exceeded 48 hours of wall-clock. Including failed runs, hyperparameter exploration, and the recalibration sweep, total project compute is approximately 375 A100-equivalent GPU-hours (9 LoRA members $\times$ 20 hr + 5 BLL seeds $\times$ 10 hr + 3 SWAG seeds $\times$ 10 hr + 3 BLIP seeds $\times$ 30 hr, plus rMD17 runs). All Omol experiments were run on NVidian H100 GPUs exclusively with 12 cpus per GPU.

Table 3: Training cost on MPtrj in A100 GPU-hours per trained instance (one BLL, one LoRA member, one BLIP or SWAG run), and inference cost in backbone forward passes per prediction. A $K{=}3$ LoRA ensemble costs three members. BLIP was trained on RTX 8000 GPUs and its training time (*) is hardware-adjusted.

Method	Train (A100-hr)	Inference (passes)
BLL	10.5	1
LoRA (per member)	19.63	1 per member
BLIP	26.5^*	10
SWAG	10.8	10

Effective rank background Following Roy & Vetterli (2007), let $s_1 \geq \cdots \geq s_D$ be the singular values of the mean-centred last-layer feature matrix, and $p_k = s_k / \sum_j s_j$. The effective rank is the exponential of the spectral entropy, $\mathrm{erank}(\Phi) = \exp(-\sum_{k=1}^{D} p_k \log p_k)$, which equals D when all directions carry equal weight and 1 when the features lie on a line. We report it as a fraction of the feature dimension, erank$/D$.

F DATASETS

MPtrj. The Materials Project trajectory dataset (Deng, 2023) (`https://doi.org/10.6084/m9.figshare.23713842`; MIT license, `https://opensource.org/license/MIT`). We partition MPtrj by the energy of each frame above its relaxed state, producing an in-distribution split (< 0.005 eV/Å) and three increasingly distorted out-of-distribution splits at $[0.005, 0.0175)$, $[0.0175, 0.070)$, and $[0.070, \infty)$ eV/Å, denoted OOD-Near, OOD-Mid, and OOD-Far. We split MP-IDs 80/10/10 (training/validation/test) using random seed 1.

rMD17. The revised MD17 dataset (Christensen & lilienfeld, 2020) (`https://doi.org/10.6084/m9.figshare.12672038`; CC0 1.0, `https://creativecommons.org/publicdomain/zero/1.0/`). We use the predefined index-1 train/test splits for all molecules.

TM23. The TM23 transition-metal benchmark (Owen et al., 2024) (CC BY-NC-ND 4.0, https://creativecommons.org/licenses/by-nc-nd/4.0/deed.en; dataset URL: https://archive.materialscloud.org/records/tcrks-ymp88/files/benchmarking_master_collection-20240316T202423Z-001.zip). We evaluate on the entire dataset (both train and test splits combined) without modification, consistent with the license's no-derivatives clause, and use it strictly for academic evaluation.

Alex-val-s. The Alexandria validation subsample distributed with OMat24 (Barroso-Luque et al., 2024; Choudhary et al., 2025/2026) (https://huggingface.co/datasets/facebook/OMAT24; CC BY 4.0, https://creativecommons.org/licenses/by/4.0/legalcode). We use the entire dataset for evaluation.

OMol25. The Open Molecules 2025 dataset (Levine et al., 2025)https://huggingface.co/facebook/OMol25; FAIR Chemistry License. We use the neutral subset. The UQ models are fit on a sub-split of the neutral training set, from which we remove the top 5% largest molecules (OOD size) and the top 5% of structures by maximum per-atom force magnitude (OOD force) , leaving $754{,}980{,}568$ training atoms, and draw a random ID test split of $30{,}000$ structures ($725{,}251$ atoms). Recalibration uses the OMol25 neutral validation set (Neutral Val, $27{,}697$ structures, 1.24×10^6 atoms), which consists of held-out compositions and is therefore already mildly shifted from the training data. All three pretrained backbones were trained on the full OMol25 training set, which contains every one of these splits.

G Metrics for Vector-valued Forces

Every method outputs, for each atom i, a predicted force $\hat{y}_i \in \mathbb{R}^3$ and a 3×3 covariance, or for the Student-t a 3×3 scale matrix, S_i. All metrics are computed per atom.

- **MAE** is the mean absolute force error in eV/Å over all Cartesian components of all atoms.
- **NLL** is the mean negative log density of y_i under the three-dimensional Gaussian $\mathcal{N}(\hat{y}_i, S_i)$ or Student-t $t_\nu(\hat{y}_i, S_i)$, in nats per atom.
- **ECE.** Let $d_i^2 = (y_i - \hat{y}_i)^\top S_i^{-1}(y_i - \hat{y}_i)$ be the squared Mahalanobis distance. For each nominal level $p \in \{0.05, 0.10, \ldots, 1.0\}$, the central predictive region is $\{d^2 \le q_p\}$, where q_p is the p-quantile of χ^2_3 for the Gaussian and $3\,F^{-1}_{3,\nu}(p)$ for the Student-t (since $d^2/3 \sim F_{3,\nu}$). ECE is the mean over the 20 levels of $|\hat{p} - p|$, where $\hat{p}$ is the observed fraction of atoms inside the region. It therefore measures calibration of the full 3×3 predictive, not of each coordinate separately.
- **Spearman** is the rank correlation between the total predicted standard deviation $\sqrt{\mathrm{tr}(S_i)}$ and the error norm $\|y_i - \hat{y}_i\|$, on a random sample of 2M atoms for splits larger than that. For BLL we compute it in float64. The raw leverage term $u(\phi_*)$ is of order 10^{-8} of the residual floor at $N \sim 10^9$, so in float32 the total variance of many atoms rounds to the same value; for eSEN-sm BLL these ties reduce the Spearman correlation to 0.14–0.21 in float32, against 0.73–0.82 in float64 across the four splits.

Because $\mathrm{tr}(S_i) = (a + b\,u)\,\mathrm{tr}(\Sigma_e)$ (for the equivariant head, $\sigma^2(3a + b\,\mathrm{tr}\ \Phi_i \Lambda_n^{-1} \Phi_i^\top)$) is increasing in the leverage for any $a, b > 0$, recalibration changes neither MAE nor Spearman.

H Recalibration Details

The Gaussian rescaling of eq. (13) and the Student-t rescaling of section 4.3 are fit on the recalibration split only: Neutral Val for OMol25. For the Student-t, we optimize the unconstrained parameters $(\log a, \log b, \log \nu)$ with L-BFGS-B and analytic gradients on the mean validation NLL, using the full 3×3 scale matrix of each atom. To avoid poor local optima we start from the Gaussian fit rescaled to six initial values of ν (from $\nu_0 \in \{1.5, 4, 30\}$), screen them on 5×10^5 atoms, and polish the best on up to 5M atoms. The median of d^2 ends close to its Student-t expectation (3.79 vs. 3.54 for eSEN-sm BLL), but this is a by-product of the fit, not a constraint. The Gaussian variant minimizes the validation NLL of eq. (13) with Nelder–Mead on $(\log a, \log b)$; at its optimum the

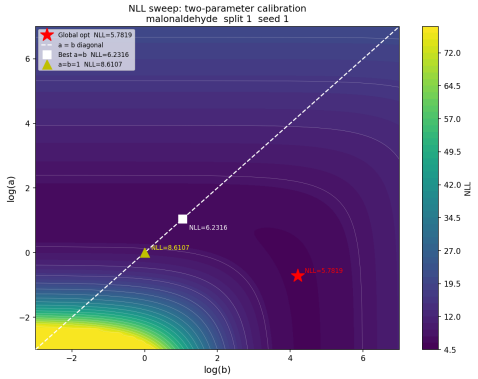

Figure 4: **Validation NLL surface for the decoupled rescaling on Malonaldehyde.** Validation NLL as a function of the rescaling parameters (a, b) from eq. (12); darker contours indicate better calibration. The dashed line marks the single-temperature subspace $a = b$ to which any scalar tempering of the BLL covariance is restricted. The unconstrained optimum (red star) sits well off the diagonal and improves substantially over both the raw BLL covariance (yellow triangle) and the best single-temperature solution (white square). This asymmetry matches our analysis: the last-layer posterior $u(\phi_*)\Sigma_e$ is too narrow and requires large upward rescaling (proposition 1), while the residual floor Σ_e is much closer to the correct scale. Absolute NLL values are elevated by a heavy right tail of high-error validation points.

validation mean of d^2 equals 3. The same code and split are used for every method. Table 4 lists the fitted scalars for every method and backbone, and table 7 reports all splits under the raw, Gaussian and Student-t predictives, together with the Student-t coverage at 0.9, 0.99 and 0.999.

H.1 Illustrative example

We estimate (a, b) on a held-out in-distribution validation set by minimizing the predictive negative log-likelihood, $(a^\star, b^\star) = \arg\min_{a,b>0} \sum_{(x_i,y_i)\in\mathcal{D}_{\text{val}}} -\log \mathcal{N}(y_i;\, \phi_i^\top B_n,\, a\Sigma_e + b\, u(\phi_i)\Sigma_e)$, optimizing the unconstrained parameterization $\alpha, \beta \in \mathbb{R}$ with $a = e^\alpha$, $b = e^\beta$ to enforce positivity. The posterior mean is unaffected; only the predictive covariance is recalibrated.

Table 4: Fitted recalibration scalars on OMol25 (fit on `neutral_val`). a multiplies the residual floor and b the input-dependent term (eq. (12)); ν is the Student-t degrees of freedom. The Gaussian fit matches the mean of d^2 to 3; the Student-t fit minimises validation NLL.

		Gaussian		**Student-t**		
Backbone	**Method**	a	b	a	b	ν
eSEN-sm	BLL	9.9×10^{-3}	1.8×10^{7}	1.4×10^{-4}	1.3×10^{6}	1.82
	LoRA-5	77.7	104	11.4	19.1	1.47
	Laplace	9.9×10^{-3}	1.8×10^{7}	1.4×10^{-4}	1.3×10^{6}	1.81
eSEN-md	BLL	0.0224	1.6×10^{7}	4.3×10^{-4}	8.5×10^{5}	1.73
	LoRA-5	0.846	1.23	0.689	0.63	10.40
OrbMol	BLL	7.6×10^{-19}	3.4×10^{6}	1.8×10^{-19}	5.1×10^{5}	1.80
	LoRA-5	0.873	0.331	0.731	0.173	11.82
	Laplace	0.0785	5.1×10^{6}	0.229	1.9×10^{6}	10.89

I Equivariant Head Details

For eSEN, the direct-force head produces, for each atom, $D = 128$ channels of $l = 1$ features, i.e. $\Phi_i \in \mathbb{R}^{3\times D}$, and the pretrained head maps them to a force with one scalar weight per channel (section section 3). We place the BLL on these weights. Each atom contributes three regression rows, the x, y and z components of its D channels, with targets the matching force components; over the 7.5×10^8 training atoms this gives 2.3×10^9 rows. With $K = 1$, the residual covariance is a single scalar σ^2, and the per-atom predictive covariance is $\sigma^2(I_3 + \Phi_i \Lambda_n^{-1} \Phi_i^\top)$. Its leverage part is anisotropic: on eSEN-sm validation, the median ratio between the largest and smallest of its three eigenvalues is 2.5. The recalibration multiplies the two parts by a and b as in eq. (12). Some channels of the eSEN features have zero variance on the training data (13 of 128 for eSEN-sm, 7 for eSEN-md, table 2); for these directions the posterior is set by the prior alone.

J ADDITIONAL OMOL25 RESULTS

Deterministic baseline. Table 5 gives the force MAE of each pretrained head without any UQ model, so that the MAE of each UQ method can be compared with the accuracy of its backbone. We report it on two splits with different meanings: Neutral Val, which none of the backbones was trained on, and our ID test split, which is part of the OMol25 training set and was therefore seen by the backbones during pretraining.

Table 5: Force MAE (eV/Å) of the pretrained heads, without a UQ model. Neutral Val was not used to train the backbones; the ID test split was part of their pretraining data.

Backbone	Neutral Val (not trained on)	ID test (trained on)
eSEN-sm	0.01008	0.00858
eSEN-md	0.00406	0.00276
OrbMol	0.01507	0.01337

Table 6: Uncertainty quantification on OMol25 across three foundation backbones. Every method is recalibrated in the same way: Student-t (a, b, ν) fit by validation NLL on `neutral_val` (section 4); ECE and NLL are after recalibration, and the last column gives the NLL of the uncalibrated predictive for reference. MAE and Spearman are unaffected by recalibration. Forces in eV/Å, NLL in nats per atom. Bold marks the best method within a backbone and split; single-method groups are not bolded. (↓ lower is better; ↑ higher is better.)

Backbone	Split	Method	MAE (↓)	ECE (↓)	Spearman (↑)	NLL (↓)	Raw NLL
eSEN-sm	Test (ID)	BLL	0.00859	**0.0222**	**0.791**	-10.49	-7.01
		LoRA-5	0.06772	0.0359	0.698	-4.34	100.33
		Laplace	**0.00858**	0.0224	0.791	**-10.49**	-7.01
	OOD size	BLL	0.00713	**0.0138**	**0.822**	-10.86	-7.93
		LoRA-5	0.06321	0.0492	0.755	-4.83	90.30
		Laplace	**0.00712**	0.0140	0.822	**-10.86**	-7.93
	OOD force	BLL	0.01320	**0.1583**	0.728	-8.30	-6.89
		LoRA-5	0.08507	0.2070	**0.822**	-3.74	13.66
		Laplace	**0.01320**	0.1587	0.728	**-8.30**	-6.89
eSEN-md	Test (ID)	BLL	0.00276	0.0124	0.764	-13.75	-9.73
		LoRA-5	**0.00275**	**0.0071**	**0.874**	**-14.31**	-14.25
	OOD size	BLL	**0.00264**	0.0174	0.804	-13.78	-10.59
		LoRA-5	0.00268	**0.0097**	**0.882**	**-14.33**	-14.23
	OOD force	BLL	**0.00501**	0.1477	0.701	-11.12	-9.77
		LoRA-5	0.00517	**0.0114**	**0.809**	**-11.65**	-11.59
OrbMol	Test (ID)	BLL	0.01348	0.0156	0.699	-8.86	-6.23
		LoRA-5	**0.01332**	**0.0102**	**0.877**	**-9.44**	-9.37
		Laplace	0.03905	0.0944	0.230	-6.12	-5.34
	OOD size	BLL	**0.01152**	0.0662	0.712	-9.24	-6.93
		LoRA-5	0.01175	**0.0110**	**0.883**	**-9.78**	-9.69
		Laplace	0.03797	0.0925	0.215	-6.18	-5.82
	OOD force	BLL	**0.02038**	0.0664	0.502	-6.92	-5.80
		LoRA-5	0.02156	**0.0073**	**0.818**	**-7.36**	-7.28
		Laplace	0.04166	0.1827	0.224	-5.08	-4.97

K ADDITIONAL MPTRJ RESULTS

Additional datasets. Beyond the MPtrj OOD splits, we evaluated on two further transfer sets (the transition-metal benchmark TM23 (Owen et al., 2024) and a non-WBM Alexandria subsample (Barroso-Luque et al., 2024), Alex-val-s). rMD17 dataset (Christensen & lilienfeld, 2020) evaluates

Table 7: Full OMol25 results under each recalibration: raw predictive, Gaussian (a, b) and Student-t (a, b, ν), all fit on `neutral_val` (the Val rows are therefore in-sample for the recalibration). Right: empirical coverage of the Student-t central regions at nominal 0.9, 0.99 and 0.999.

			ECE (↓)			**NLL (↓)**			**Student-t coverage**		
Backbone	**Split**	**Method**	Raw	Gauss.	Stud.-t	Raw	Gauss.	Stud.-t	0.9	0.99	0.999
eSEN-sm	Val	BLL	0.360	0.353	0.012	0.30	-8.25	-10.68	0.904	0.992	0.999
		LoRA-5	0.427	0.189	0.030	116.61	-3.31	-4.15	0.908	1.000	1.000
		Laplace	0.361	0.353	0.012	0.29	-8.25	-10.68	0.904	0.992	0.999
	Test (ID)	BLL	0.344	0.349	0.022	-7.01	-9.02	-10.49	0.903	0.998	1.000
		LoRA-5	0.407	0.229	0.036	100.33	-3.28	-4.34	0.920	1.000	1.000
		Laplace	0.344	0.349	0.022	-7.01	-9.02	-10.49	0.903	0.998	1.000
	OOD size	BLL	0.366	0.368	0.014	-7.93	-9.34	-10.86	0.925	1.000	1.000
		LoRA-5	0.418	0.247	0.049	90.30	-3.75	-4.83	0.927	1.000	1.000
		Laplace	0.366	0.369	0.014	-7.93	-9.34	-10.86	0.926	1.000	1.000
	OOD force	BLL	0.223	0.272	0.158	-6.89	-7.83	-8.30	0.840	0.999	1.000
		LoRA-5	0.370	0.401	0.207	13.66	-2.18	-3.74	0.995	1.000	1.000
		Laplace	0.223	0.272	0.159	-6.89	-7.83	-8.30	0.840	0.999	1.000
eSEN-md	Val	BLL	0.385	0.378	0.013	4.85	-10.77	-13.65	0.906	0.992	0.999
		LoRA-5	0.081	0.042	0.004	-14.15	-14.17	-14.24	0.901	0.992	0.999
	Test (ID)	BLL	0.389	0.396	0.012	-9.73	-11.72	-13.75	0.927	0.999	1.000
		LoRA-5	0.090	0.049	0.007	-14.25	-14.28	-14.31	0.910	0.994	1.000
	OOD size	BLL	0.387	0.390	0.017	-10.59	-11.95	-13.78	0.922	1.000	1.000
		LoRA-5	0.075	0.034	0.010	-14.23	-14.24	-14.33	0.904	0.994	1.000
	OOD force	BLL	0.257	0.312	0.148	-9.77	-10.35	-11.12	0.834	0.999	1.000
		LoRA-5	0.095	0.051	0.011	-11.59	-11.62	-11.65	0.919	0.996	1.000
OrbMol	Val	BLL	0.321	0.250	0.017	-1.41	-7.66	-8.95	0.901	0.995	1.000
		LoRA-5	0.090	0.034	0.002	-9.40	-9.43	-9.47	0.903	0.991	0.998
		Laplace	0.212	0.055	0.107	-1.96	-5.91	-6.12	0.946	0.982	0.991
	Test (ID)	BLL	0.308	0.274	0.016	-6.23	-7.94	-8.86	0.924	0.999	1.000
		LoRA-5	0.102	0.044	0.010	-9.37	-9.42	-9.44	0.916	0.994	1.000
		Laplace	0.202	0.087	0.094	-5.34	-5.96	-6.12	0.959	0.990	0.996
	OOD size	BLL	0.331	0.319	0.066	-6.93	-8.19	-9.24	0.954	1.000	1.000
		LoRA-5	0.107	0.046	0.011	-9.69	-9.76	-9.78	0.921	0.996	1.000
		Laplace	0.220	0.097	0.093	-5.82	-5.97	-6.18	0.980	0.997	0.999
	OOD force	BLL	0.174	0.318	0.066	-5.80	-5.87	-6.92	0.954	1.000	1.000
		LoRA-5	0.102	0.039	0.007	-7.28	-7.35	-7.36	0.918	0.996	1.000
		Laplace	0.098	0.322	0.183	-4.97	-4.42	-5.08	0.971	0.995	0.999

Table 8: OrbMol feature rank under different feature sites and adapters ($D = 256$, mean-centered). Adding a trainable MLP adapter, hooking an earlier layer, or using the features of a LoRA-r8 ensemble member moves the effective rank only slightly.

	Val			**Test (ID)**		
Features	erank	erank$/D$	95%	erank	erank$/D$	95%
Pretrained head (BLL features)	44.3	0.173	9	41.9	0.164	8
+ MLP adapter (256)	46.5	0.182	10	42.2	0.165	9
+ MLP adapter (256-256)	45.5	0.178	10	39.2	0.153	9
Penultimate hook + MLP adapter	50.7	0.198	9	46.6	0.182	9
LoRA-r8 member (seed 0)	41.7	0.163	9	39.4	0.154	8

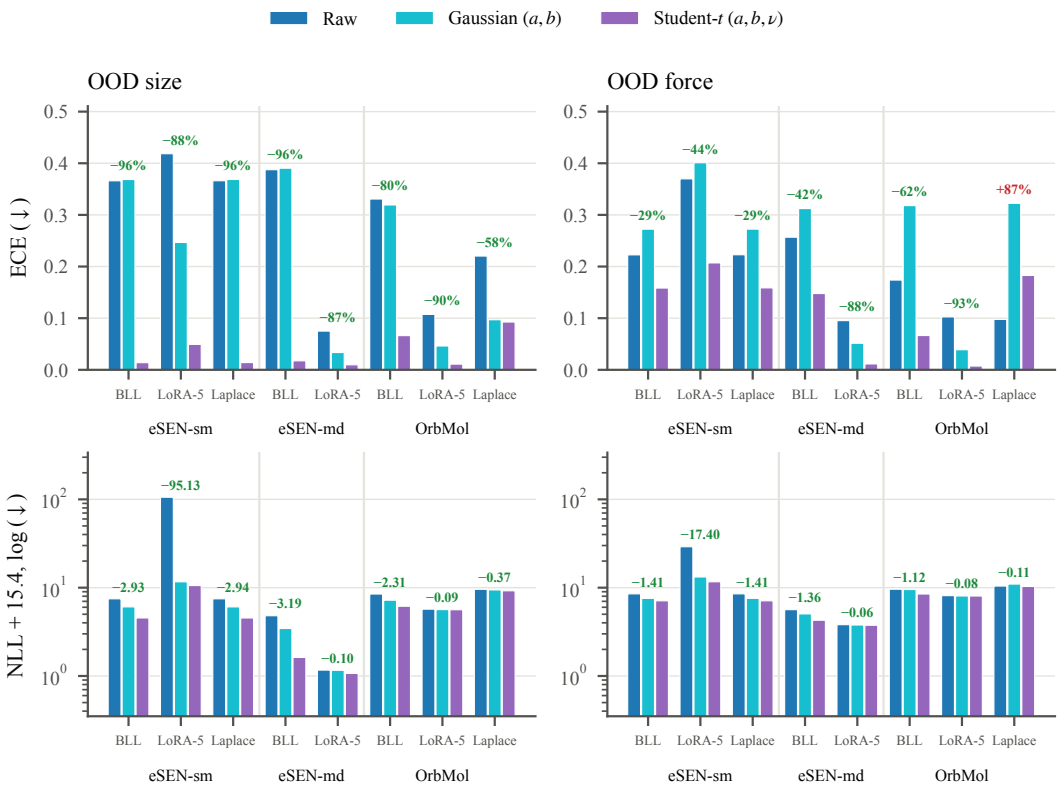

Figure 5: **Recalibration ablation on the OMol25 OOD splits.** Same layout as fig. 1(b, d), for OOD size (left) and OOD force (right). On OOD force the Gaussian rescaling increases ECE for every method except the OrbMol LoRA ensemble.

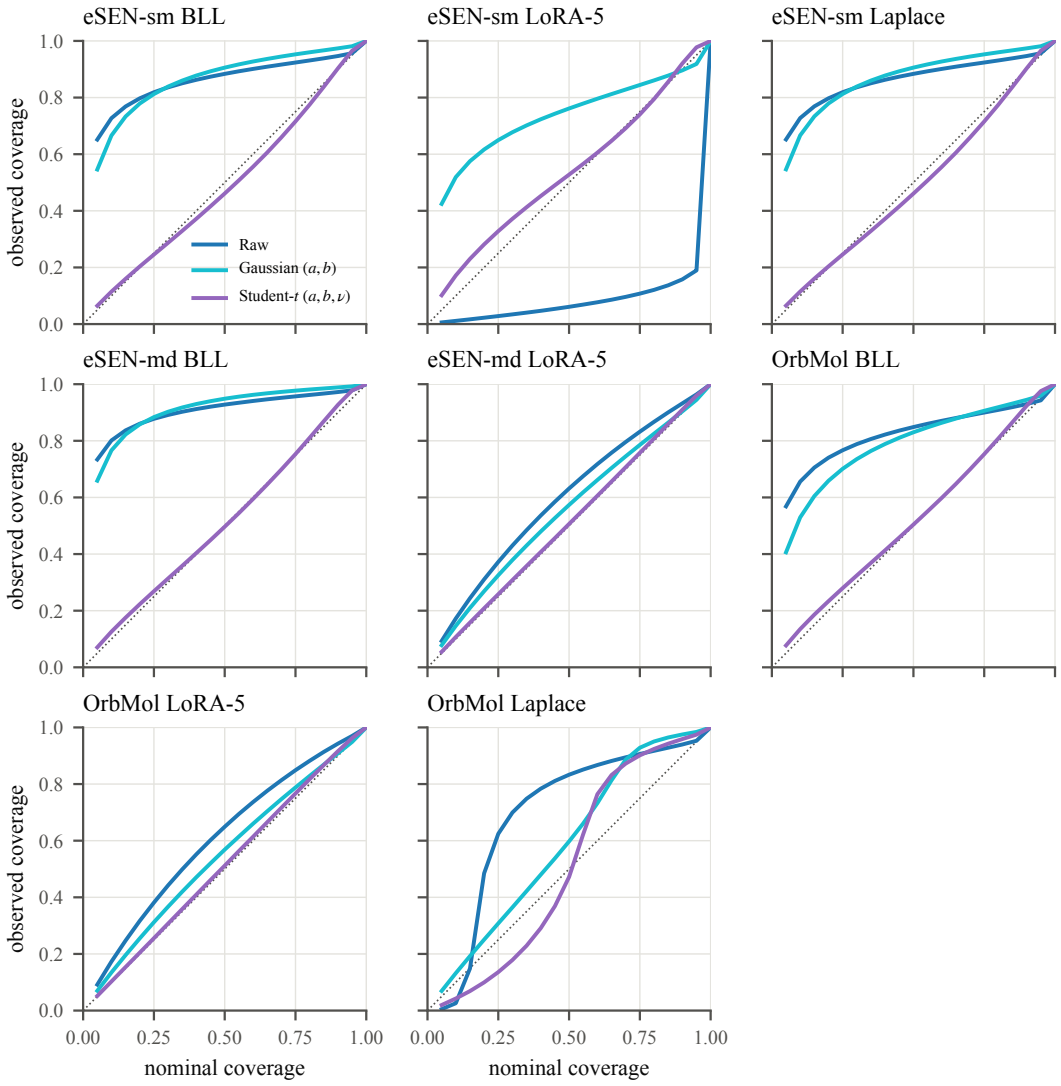

Figure 6: **Coverage curves on the OMol25 ID test split.** Observed versus nominal coverage of the central predictive regions for the raw predictive and the two recalibrations; the dotted diagonal is perfect calibration.

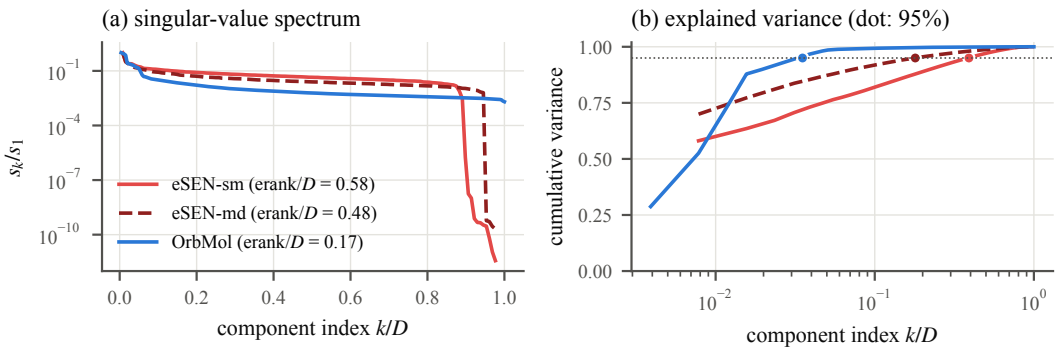

Figure 7: **Last-layer feature spectra of the OMol25 backbones** (mean-centered Neutral Val features). (a) Singular values relative to the largest; the sharp drop at the right end for eSEN marks the dead dimensions (13 for eSEN-sm, 7 for eSEN-md). (b) Cumulative explained variance; dots mark 95%.

a different regime; the model is trained on a single molecule's trajectory and asked to extrapolate to a held-out split of the same molecule (ID) and to all non-training molecules (OOD), a setting in which a linear head over a frozen encoder is most likely to overfit. On rMD17, exact Gaussian processes were added as baselines.

Results. The pattern carries through to the additional splits in Table 9: recalibrated BLL achieves the highest Spearman on four of the five (MPtrj ID, MPtrj TM23, rMD17 ID, rMD17 OOD), with Alex-val-s the lone exception (BLL 0.764 vs. BLIP 0.781). The rMD17 OOD split is the most pronounced instance of the pattern: although BLL has the worst MAE there (0.259), since a single-molecule training trajectory provides little extrapolation signal for the linear head, its Spearman of 0.821 is the second largest gap to the next-best method (0.702 for the ensemble) of any split in our comparison. Recalibration does not rescue OOD ECE in this regime ($0.483 \rightarrow 0.463$), a boundary case we examine below.

Discussion. The procedure fails on rMD17, where OOD ECE moves only from 0.483 to 0.463. The cause is that rMD17 trains and validates on a single molecule's trajectory, leaving too little variability to identify $(a^\star, b^\star)$. On MPtrj and rMD17, BLL achieves the highest Spearman correlation on nearly every split, including rMD17 OOD where its MAE and ECE are poor.

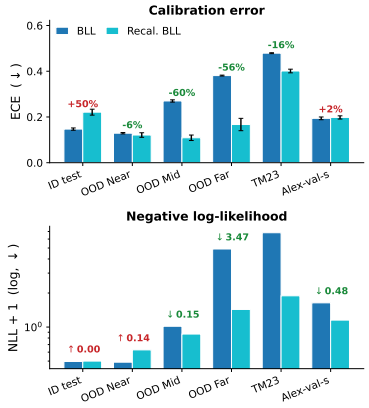

Figure 8: **Previous version of fig. 1(a, c), including TM23 and Alex-val-s. Recalibration improves both calibration and likelihood across most splits.** Top: expected calibration error (ECE) before and after two-parameter scaling on the BLL predictive variance. Bottom: negative log-likelihood (NLL+1 on a log scale, since raw NLL can be negative). Annotations show the relative change. Recalibration delivers large ECE reductions on the harder OOD splits (-56% to -60%) and a $\sim$3.5 unit NLL drop on OOD Far, at the cost of slight over-correction on the well-calibrated ID test set. MAE and Spearman correlation are unchanged by recalibration and reported in Table 9 and Figure 2

Table 9: Model performance on the MPtrj and rMD17 data splits. BLL uses the Gaussian (a, b) recalibration of section 4.3. Best results for each split are highlighted in bold. (↓ indicates lower is better; ↑ indicates higher is better).

Data Split	Model	MAE (↓)	ECE (↓)	Spearman (↑)	NLL (↓)
MPtrj ID test	Recalibrated BLL (ours)	0.038 ± 0.000	0.221 ± 0.013	**0.603** ± 0.009	-0.500 ± 0.018
	BLIP	**0.029** ± 0.001	0.268 ± 0.010	0.504 ± 0.001	-
	Ensemble (LoRA)	0.046 ± 0.002	0.267 ± 0.024	0.579 ± 0.024	-0.514 ± 0.013
	SWAG	0.043 ± 0.001	**0.117** ± 0.023	0.329 ± 0.014	**-0.521** ± 0.062
MPtrj TM23 test	Recalibrated BLL (ours)	0.542 ± 0.029	0.401 ± 0.008	**0.866** ± 0.027	**0.889** ± 0.052
	BLIP	**0.195** ± 0.004	**0.400** ± 0.004	0.718 ± 0.014	-
	Ensemble (LoRA)	0.780 ± 0.022	0.403 ± 0.021	0.664 ± 0.029	1.647 ± 0.221
	SWAG	0.361 ± 0.026	0.474 ± 0.001	0.644 ± 0.007	1.060 ± 0.186
MPtrj Alex-val-s test	Recalibrated BLL (ours)	0.077 ± 0.001	0.198 ± 0.007	0.764 ± 0.015	0.152 ± 0.014
	BLIP	**0.052** ± 0.000	0.162 ± 0.003	**0.781** ± 0.003	-
	Ensemble (LoRA)	0.106 ± 0.022	0.224 ± 0.001	0.709 ± 0.025	**-0.062** ± 0.028
	SWAG	0.077 ± 0.001	**0.129** ± 0.007	0.366 ± 0.015	0.247 ± 0.017
rmd17 ID test	Recalibrated BLL (ours)	**0.030** ± 0.015	0.132 ± 0.067	**0.611** ± 0.056	-0.922 ± 0.181
	GP-rbf	0.057 ± 0.019	**0.101** ± 0.042	0.538 ± 0.58	13.11 ± 28.79
	GP-5/2	0.056 ± 0.019	0.116 ± 0.017	0.555 ± 0.024	**-1.321** ± 0.338
	Ensemble (LoRA)	0.098 ± 0.019	0.152 ± 0.039	**0.611** ± 0.234	0.445 ± 0.109
rmd17 OOD test	Recalibrated BLL (ours)	0.259 ± 0.045	0.463 ± 0.017	**0.821** ± 0.079	0.278 ± 0.200
	GP-rbf	0.177 ± 0.056	0.245 ± 0.092	0.581 ± 0.125	11.67 ± 28.28
	GP-5/2	0.166 ± 0.050	**0.199** ± 0.097	0.550 ± 0.131	0.876 ± 1.169
	Ensemble (LoRA)	**0.125** ± 0.011	0.260 ± 0.045	0.702 ± 0.071	**-0.356** ± 0.074