Secure Out-of-Distribution Task Generalization with Energy-Based Models

Shengzhuang Chen\textsuperscript{1}  Long-Kai Huang\textsuperscript{2}  Jonathan Richard Schwarz\textsuperscript{3}  Yilun Du\textsuperscript{4}  Ying Wei\textsuperscript{1,5,*}
\textsuperscript{1}City University of Hong Kong  \textsuperscript{2}Tencent AI Lab  \textsuperscript{3}University College London  
\textsuperscript{4}Massachusetts Institute of Technology  \textsuperscript{5}Nanyang Technological University
\texttt{szchen9-c@my.cityu.edu.hk}  \texttt{hlongkai, schwarzjn@gmail.com}
\texttt{yilundu@mit.edu}  \texttt{ying.wei@ntu.edu.sg}

Abstract

The success of meta-learning on out-of-distribution (OOD) tasks in the wild has proved to be hit-and-miss. To safeguard the generalization capability of the meta-learned prior knowledge to OOD tasks, in particularly safety-critical applications, necessitates detection of an OOD task followed by adaptation of the task towards the prior. Nonetheless, the reliability of estimated uncertainty on OOD tasks by existing Bayesian meta-learning methods is restricted by incomplete coverage of the feature distribution shift and insufficient expressiveness of the meta-learned prior. Besides, they struggle to adapt an OOD task, running parallel to the line of cross-domain task adaptation solutions which are vulnerable to overfitting. To this end, we build a single coherent framework that supports both detection and adaptation of OOD tasks, while remaining compatible with off-the-shelf meta-learning backbones. The proposed Energy-Based Meta-Learning (EBML) framework learns to characterize any arbitrary meta-training task distribution with the composition of two expressive neural-network-based energy functions. We deploy the sum of the two energy functions, being proportional to the joint distribution of a task, as a reliable score for detecting OOD tasks; during meta-testing, we adapt the OOD task to in-distribution tasks by energy minimization. Experiments on four regression and classification datasets demonstrate the effectiveness of our proposal.

1 Introduction

Meta-learning\textsuperscript{[43,6]} that builds general-purpose learners with limited data has been under constant investigation, recently demonstrating its potential to even advance few-shot learning of large language models\textsuperscript{[36,44]}. Analogous to the notorious domain shift\textsuperscript{[23]} that degrades the performance of deep learning, meta-testing tasks that are out of the distribution of meta-training tasks (a.k.a. out-of-distribution (OOD) tasks) put the meta-learned prior knowledge at high risk of losing effectiveness\textsuperscript{[46]}. In real-world applications, though, out-of-distribution tasks are highly prevalent, e.g., bin picking for a robot that has never been meta-trained on environments involving bins\textsuperscript{[55]}, MRI-based pancreas segmentation given a host of meta-training tasks with pathology images\textsuperscript{[35]}, and etc. Thus, it is imperative to secure the generalization ability of the meta-learned prior (i.e., meta-generalization) to OOD tasks, especially in safety-critical applications such as medical image analysis.

The first step to securing meta-generalization to a task is to develop awareness of whether the task is OOD or not, i.e., OOD task detection. Existing solutions in literature have pursued a variety of Bayesian meta-learning methods\textsuperscript{[7,54,41,10,43]} that balance between flexibility and tractability of

\*Correspondence to Ying Wei

The key contributions of this research are outlined below. (1) **Coherence and generality**: we provide a coherent probabilistic model that allows both detection and adaptation of OOD tasks. Also, EBML is agnostic to meta-learning backbones, being general to secure meta-generalization for arbitrary off-the-shelf meta-learning approaches against OOD tasks. (2) **Practical efficacy**: we conduct our experiments on three regression and one classification datasets, on which EBML outperforms SOTA Bayesian meta-learning methods for OOD task detection with an improvement of up to 7% on AUROC and cross-domain meta-learning approaches for OOD task adaption with up to 1.5% improvement.

2 Related Work

**Bayesian Meta-learning** There has been a line of literature on Bayesian meta-learning algorithms with predictive uncertainty estimation for safeguarding safety-critical and few-shot applications. Grant...
et al. [11] first recast gradient-based meta-learning as a tractable hierarchical Bayesian inference problem. Much of the subsequent research attempts to solve the problem with various approximations. Assuming a sufficient number of meta-training tasks, almost all works use a point estimate for the initialization [41, 19, 10]. However, estimates of exceptions including [54] rely on SVGD [32] for inference and require significant computation for an ensemble of task-specific weights. Several studies that estimate the uncertainty in task-specific parameters after inner-loop adaptation have explored MAP estimates [e.g. 47], sampling from a neural network [10, 53, 52], and variational inference [41, 7, 43]. The uncertainties considered in these methods are often modelled using isotropic Gaussians which suffer from limited expressiveness.

Meta-learning towards OOD Generalization Recent cross-domain meta-learning methods [e.g. 25, 4, 49, 34] deal with a distribution shift between meta-training and meta-testing tasks, by typically parameterizing deep networks with a large set of task-agnostic and a small set of task-specific weights that encode shared representations and task-specific representations for the training domains, respectively. The works of [42, 1, 34] augment a shared pre-trained backbone with task-specific FiLM [40] layers whose parameters are estimated through an encoder network conditioned on the task’s support set. TSA [28] and URL [29] propose to attach task-specific adaptors in matrix form to the pre-trained backbone at test time, inferring their parameters by gradient descent on the support set for each task from scratch. On the other hand, SUR [4] and URT [31] pre-train multiple backbones each for an ID training domain, and meta-learn an attention mechanism to selectively combine the pre-trained representations into task-specific ones for ID and OOD classification. While these methods generally have improved performance in the OOD domains of tasks, they nevertheless are not designed with any explicit mechanism for detecting OOD tasks, i.e., lacking OOD awareness.

EBMs for OOD Detection Recently, there has been increasing interest in leveraging EBMs for detecting testing samples that are OOD w.r.t. the training data distribution. Liu et al. [33] directly use the energy score for OOD input detection, while Grathwohl et al. in JEM [12] use gradient norm of the energy function as an alternative OOD score; both yield more superior OOD detection performance than traditional density-based detection methods. There are also a number of works that investigate the OOD detection capability of hybrid and latent variable EBMs [38, 14, 13], and more advanced training techniques for improving the density modelling hence OOD detection performance of EBMs [5, 2, 57, 3]. While all aforementioned works focus on the standard supervised and unsupervised learning scenarios, Willette et al. in [52] study OOD detection in meta-learning. However, their work differs from EBML in that (a) EBML aims to detect a meta-testing task that is OOD of the meta-training tasks whereas [52] focuses on detecting a query sample that is OOD of the support samples in a meta-testing task, and (b) EBML explicitly meta-learns the distribution of meta-training tasks via the two proposed EBMs and develops the Energy Sum to flag those high-energy tasks as OOD tasks; while [52] resorts to post-hoc OOD detection via energy scaling (akin to temperature scaling in softmax output) without learning any EBM. Moreover, we offer EBML as a generic and flexible probabilistic meta-learning framework that supports both detection and adaptation of OOD tasks.

3 Preliminaries: Energy-based Models

An energy-based model (EBM) [24] expresses a probability density \( p(x) \) for \( x \in \mathbb{R}^D \) as

\[
p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z(\theta)},
\]

where \( E_\theta(x) \) is the energy function parameterized by \( \theta \) that maps each point \( x \) in the input space to a scalar value known as the energy. \( Z(\theta) = \int \exp(-E_\theta(x))dx \) is the partition function that is a constant w.r.t. the variable \( x \). Training \( p_\theta(x) \) to fit some data distribution \( p_D(x) \) requires maximizing the log-likelihood \( \mathcal{L}(\theta) = \mathbb{E}_{x \sim p_D(x)}[\log p_\theta(x)] \) w.r.t. \( \theta \). Though an intractable integral in \( Z_\theta \) is involved in this objective, it is not a concern when computing the gradient [3, 12]

\[
\nabla_\theta \mathcal{L} = \mathbb{E}_{x \sim p_D}[\nabla_\theta E_\theta(x)] - \mathbb{E}_{x \sim p_D}[\nabla_\theta E_\theta(x)].
\]

Intuitively, Eqn. (2) encourages \( E_\theta \) to assign low energy to the samples from the real data distribution \( p_D \) while assigning high energy to those from the model distribution \( p_\theta \). Computing Eqn. (2), thus, requires drawing samples from \( p_\theta \), which is challenging. Recent approaches [12, 4] on training EBMs resort to stochastic gradient Langevin dynamics (SGLD) [51] which generates samples following

\[
x^0 \sim p_\theta(x), \quad x^{k+1} = x^k - \frac{\eta^2}{2} \frac{\partial E_\theta(x^k)}{\partial x^k} + \eta^k.
\]

3
The $K$-step sampling starts from an (typically uniform) initial distribution $p_0(x)$. $z^k \sim \mathcal{N}(0, I) \in \mathbb{R}^D$ is a perturbation, and $\eta \in \mathbb{R}^+$ controls the step size and noise magnitude. Denote the distribution $q_0$ by Eqn. (3), which signifies $x' = x^K \sim q_0$. When $\eta \to 0$ and $K \to \infty$, then $q_0 \to p_0$ under some regularity conditions [51]. Consequently, the gradient of Eqn. (2) is approximated in practice [3, 12] by

$$\nabla_x \mathcal{L} = \mathbb{E}_{x' \sim \text{grad}(q_0)}[\nabla_x E_\theta(x')] - \mathbb{E}_{x \sim p_D}[\nabla_x E_\theta(x)],$$

(4)

where the gradient does not back-propagate into SGLD sampling.

4 Energy-Based Meta-learning

For clarity, we use the notation $\mathcal{P}_{ID}$ to denote the unknown meta-training ID task distribution where the $i$-th training task is $T_i$. We let $X_i$, $Y_i$ to denote sets of samples $\{x_{ij}, y_{ij}\}$ in $T_i$, and $T_i^s$, $T_i^t$ to denote support and query sets, respectively. The size of $T_i$, $T_i^s$, $T_i^t$ is denoted by $N_i, N_i^s, N_i^t$, respectively. The subscript $i$ denotes the task index, and $j$ denotes the sample index.

4.1 Energy-based Modelling of Task Distribution

As illustrated in Introduction, existing probabilistic meta-learning methods maximizing the predictive likelihood $p(Y|X)$ suffer from incomplete OOD coverage. To this end, we model the meta-training task distribution by (1) formulating the joint distribution $p(X_i, Y_i)$ of each task $T_i$ and (2) maximizing the log-likelihood of all meta-training tasks. Concretely, by Kolmogorov’s extension and de Finetti’s theorems [22], we have the expected log-likelihood of the meta-training tasks as $\mathbb{E}_{\mathcal{P}_{ID}}[\log p(T_i)] = \mathbb{E}_{\mathcal{P}_{ID}}[\log p(X_i, Y_i)] = \mathbb{E}_{\mathcal{P}_{ID}}[\log \prod_{j=1}^{N_i} p(x_{ij}, y_{ij}|\phi_i)p(\phi_i)d\phi_i]$. Each $p(T_i)$ is written in a factorized form over $N_i$ conditional independent distributions with $\phi_i$ being the task-specific latent variable. Due to the intractable integral over $\phi_i$ in high dimension, we resort to amortized inference [8, 41] and learn with a lower-bound instead. This gives the ELBO

$$\mathbb{E}_{\mathcal{P}_{ID}}[\log p(T_i)] \geq \mathbb{E}_{\phi_i \sim q_0(\phi_i|T_i)} \left[ \log \prod_{j=1}^{N_i} p(x_{ij}, y_{ij}|\phi_i) \right] - \text{KL}(q_0(\phi_i|T_i)||p(\phi_i)).$$

(5)

Following the conventional wisdom [41, 28, 6], $q_0$ is conditioned on the support set only during meta-training to align the inference procedure, i.e., $\phi_i \sim q_0(\phi_i|T_i^s)$, for meta-training and meta-testing. It remains now to parameterize the three distributions in Eqn. (5) including (a) the task-specific data distribution $p(x_{ij}, y_{ij}|\phi_i)$, (b) the prior latent distribution $p(\phi_i)$, and (c) the posterior latent distribution $q_0(\phi_i|T_i^s)$. Prior works parameterize these distributions in simple known forms, e.g., Gaussians [41, 3, 43] or MAP estimation [6, 53], which may be insufficient to match the complex probabilistic structure of the meta-training task distribution. To increase the expressiveness, we turn to EBMs for parameterizing the two distributions of $p(x_{ij}, y_{ij}|\phi_i)$ and $p(\phi_i)$. For one reason, EBMs are known to be sufficiently flexible and expressive for characterizing complex arbitrary density functions [3] not limiting to only uni-modal distributions like isotropic Gaussians and MAP estimation; for another, the energy function of an EBM is directly proportional to the negative log-likelihood, paving the way for OOD detection in Section 4.2.

(a) Task-specific data EBM We model $p(x_{ij}, y_{ij}|\phi_i)$ by an energy function parameterized with $\omega$,

$$p(x_{ij}, y_{ij}|\phi_i) = p_\omega(x_{ij}, y_{ij}|\phi_i) = \frac{\exp(-E_\omega(x_{ij}, y_{ij}, \phi_i))}{Z(\omega, \phi_i)},$$

(6)

where $E_\omega$ denotes the task-specific data energy function conditioned on the latent $\phi_i$, and $Z(\omega, \phi_i)$ is the corresponding partition function. Note that the parameter $\omega$ of this EBM is shared by all tasks.

(b) Latent prior EBM Inspired by [39], we model the prior latent distribution $p(\phi_i)$ as an unconditional EBM parameterized by $\lambda$; training such a EBM offers expressiveness benefits over a fixed non-informative prior distribution, e.g., isotropic Gaussian distribution. Specifically,

$$p(\phi_i) = \frac{\exp(-E_\lambda(\phi_i))}{Z(\lambda)}, \forall i.$$

(7)

(c) Latent posterior As many meta-learning algorithms have already carefully designated the posterior latent distribution $q_0(\phi_i|T_i^s)$, we simply follow the same implementation of $q_0$ in the chosen base meta-learning algorithm, e.g., MAP estimation in [8, 42, 53]. This design favorably empowers EBML to be a generic and flexible framework compatible with off-the-shelf meta-learning algorithms.
Grounded on the above parameterization, we are now ready to derive our EBML meta-training objective as below by plugging the two EBMs defined in Eqn. (6) and Eqn. (7) into Eqn. (5). The derivation shares the spirit with Eqn. (4), and more details can be found in Appendix A.1.

$$\arg\max_{\omega, \phi, \lambda} \mathbb{E}_{T_i \sim P_{ID}} \left[ \mathbb{E}_{\phi_i \sim q_{\phi_i}|T_i^*}} \left( \sum_{j=1}^{N_i} -E_\omega(x_{ij}, y_{ij}, \phi_i) + \mathbb{E}_{p_{\lambda}}(x', y'|\phi_i)[E_\omega(x'_ij, y'_ij, \phi_i)] \right) 
- \mathbb{E}_{q_{\phi_i}|T_i^*}[E_\lambda(\phi_i)] + \mathbb{E}_{p_\lambda(\phi_i)}[E_\lambda(\phi_i)] + \mathcal{H}(q_{\phi_i}|T_i^*) \right].$$

(8)

Solving the above meta-training objective involves sampling of $x', y'$ from $p_{\omega}$ and $\phi'$ from $p_{\lambda}$, in order to compute the expectations $\mathbb{E}_{p_{\omega}}(x', y'|\phi_i)$ and $\mathbb{E}_{p_\lambda(\phi_i)}$ as Monte-Carlo averages. We follow the similar SGD sampling procedure in Eqn. (4). Besides, since the majority of state-of-the-art meta-learning algorithms adopt the MAP estimation of the latent posterior $q_{\phi_i}$ which is deterministic, the last entropy term of $\mathcal{H}$ essentially becomes zero and the expectations in the first and second terms are trivial to solve. For this reason, we focus on base meta-learning algorithms with MAP approximation in the following sections, which not only simplifies computation but also maintains the state-of-the-art performance. We left a discussion on EBML with distributional $q_{\phi_i}$ in Appendix C.3. The complete pseudo codes for meta-training of EBML are available in Appendix E.

### 4.2 EBML for OOD Detection

Detecting an OOD task w.r.t. the meta-training distribution constitutes an essential first step to guard successful meta-generalization. A straightforward solution is density-based OOD detection, for which the OOD score of a task following the Bayesian principle boils down to its log-likelihood $\log p(x_i^*, y_i^*) = \log \mathbb{E}_{\phi_i \sim p_{\lambda}(\phi_i)}[p_{\omega}(X_i^*, Y_i^*|\phi_i)]$. Despite the meta-learned latent prior EBM $p_{\lambda}(\phi_i)$ that is readily available, estimating this log-likelihood still presents daunting challenges. First, when the latent prior is expressed in the form of a distribution over model parameters in very high dimension, MCMC sampling from $p_{\lambda}(\phi_i)$ is almost computationally infeasible. Second, especially when the latent prior exhibits multi-modality, drawing a considerable number of samples to achieve a low-variance MC estimation of the integral is prohibitively costly.

On this account, we define the OOD score of a task to be faithful to our proposed ELBO approximation of its log-likelihood in Eqn. (5), which gives

$$\mathbb{E}_{q_{\phi_i}|T_i^*}[\sum_{j}^{N_i} E_\omega(x_{ij}', y_{ij}', \phi_i)] + E_\lambda(\phi_i)].$$

(9)

We dub this OOD score tailored to EBML Energy Sum, whose full derivation is deferred to Appendix A.2. This energy sum enjoys not only the theoretical advantage, i.e., being provably proportional to the negative log-likelihood of a task, but also simple computation benefits. During meta-testing, evaluating the score of Eqn. (9) for each task requires only a single forward pass of the support set samples through the two energy functions.

More remarkably, the energy sum is intuitively appealing in the sense that it characterizes (1) how far a task is from the overall ID meta-training task distribution via the latent prior energy score $E_\lambda$ and (2) how difficult it is to predict the observed support set conditioned on $\phi_i$ via the task-specific data energy score $E_\omega$. First, the terms in the last line of Eqn. (8) for learning the latent prior EBM altogether correspond to maximizing the likelihood $\mathbb{E}_{T_i \sim P_{ID}} \mathbb{E}_{q_{\phi_i}|T_i^*}[\log p_{\lambda}(\phi_i)]$, which enforces the latent prior energy score $E_\lambda$ to capture the overall ID meta-training distribution. As illustrated in
Within this region, minimizing the first term in Eqn. (11) continuously improves generalization. In when optimizing Eqn. (11) for OOD few-shot clas-

Therefore, we introduce task-specific parameters that we differentiate our meta-testing procedures for effective meta-generalization.

Provided that the task has already been identified within the ID region, the second energy $E_\lambda(\phi)$ is negligibly small. Consequently, we reduce the above optimization problem to consider only the first term $E_s(x_i^j, y_i, \phi)$, and solve it via gradient descent. We provide the pseudo codes in Appendix E.

Meta-testing for OOD tasks For an OOD task, its meta-learned prior $\phi \sim q_\psi(\phi | T^*)$ is located out of the ID meta-training task distribution and likely loses its effectiveness. We seek a solution that adapts this inadequate meta-learned prior back to the ID region, so as to make the most of the ID latent priors with guaranteed meta-generalization. This shares the idea with classifier editing in [45], where the editing parameters are trained to map an OOD image to an ID one for improving generalization. Therefore, we introduce task-specific parameters $\zeta$ which are optimized via the following,

$$\arg \min_\zeta \mathbb{E}_{\phi \sim q_\psi(\phi | T^*)} \left[ \sum_{j=1}^{N^*} E_s(x_i^j, y_i^j, \phi) + \max(E_\lambda(\phi) - m, 0) \right], \quad (11)$$

where $m$ is a hyper-parameter. We find that setting $m$ as the empirical average of the latent prior energy over all ID training tasks works well in practice, i.e., $m = \mathbb{E}_{p_{ID}}[\mathbb{E}_{\phi \sim q_\psi(\phi | T)}[E_\lambda(\phi)]]$.

As a result of optimizing the second term in Eqn. (11), the task-specific parameters $\zeta$ enable $q_\psi(\phi | T^*)$ to accommodate for OOD tasks by mapping the meta-learned prior back to ID meta-training tasks; while optimizing the first term preserves the data-level predictive ability of the model. We highlight that the task energy minimization approximates the minimization of a KL divergence between the task-specific posterior and the meta-learned prior, thereby inducing a meta-regularization effect during adaptation. See Appendix A.3 for details. Eventually, we use the adapted task-specific parameters for final prediction on query samples as in Eqn. (10). Pseudo code for the EBML adaptation and inference algorithms described above can be found in Appendix E.

In Figure 3, we visualize the adaptation process when optimizing Eqn. (11) for OOD few-shot class-
fication tasks in Meta-dataset [49]. As the prior energy of these OOD tasks decreases, their $\phi_i$ gradually shift towards the ID region as desired. Within this region, minimizing the first term in Eqn. (11) continuously improves generalization. In contrast, given only a few support samples, existing SOTA methods that solely rely on feed-forward inference [11] and gradient-based optimization [28] for OOD task adaptation without a prior are both prone to overfitting. We provide more empirical evidence on this in Appendix C. On the other hand, meta-learning a BNN, which imposes a prior distribution on the parameter space during adaptation may be computationally cumbersome and often lead to sub-optimal performance in comparison to their non-Bayesian counterparts.
5 Experiments

In the experiments, we test EBML on both few-shot regression and image classification tasks in search for answers to the following key questions: **RQ1**: Whether the improved expressiveness of EBML over traditional Bayesian meta-learning methods can lead to a more accurate model of the meta-training ID task distribution, hence a more reliable OOD task detector. **RQ2**: Whether Energy Sum can be an effective score for detection of OOD meta-testing tasks. **RQ3**: Whether EBML instantiated with SOTA algorithms can exploit the meta-learned EBM prior in OOD task adaptation to achieve better prediction performance on OOD tasks.

5.1 Implementation Details

We now discuss two instantiations of the EBML framework with SOTA meta-learning algorithms for regression and classification. We illustrate our approach in Figure 4 below and defer a more detailed description for our models to Appendix B.

Regression. Take CNPs [8] as an example base model. CNPs implements $q_\psi(\phi_i | T_s^i)$ as a neural network encoder that outputs a function embedding in finite vector form, i.e., $\phi_i \in \mathbb{R}^D$, from a given support set, $T_s^i$. That said, we let the prior EBM to model the empirical distribution over such finite-dimension function embedding, i.e., $E_\lambda(\phi_i) : \mathbb{R}^D \rightarrow \mathbb{R}$.

Classification Many cross-domain few-shot classification algorithms [28, 42, 1] rely on a metric-based classifier for prediction, which assigns query sample to the class with nearest prototype to the query representation based on some distance measure. In these cases, it is natural to specify the task-specific latent $\phi_i$ as the set of class prototypes in each ID training task. Since $\phi_i$ is a set of variables, we build the prior EBM model as a permutation-invariant neural network function. Suitable choices include DeepSets [56] and set transformer [26].

To align with the state-of-the-art prediction performance, we follow the practice in [50, 37] to train another decoder $\omega_2$ with the loss function (e.g., cross entropy) in the base meta-learning model, which serves as a surrogate for $E_\omega(x_q^i, y, \phi)$ in Eqn. (10) and Eqn. (11). We use this decoder for prediction.

Baseline Models For regression, we compare against: 1) MAML [6] which is a deterministic meta-learning method, and 2) Bayesian meta-learning methods that use Gaussians for prediction or prior, including ABML [41], MetaFun [53], CNPs [8] and F-PACOH-GP [43]. For classification, we consider Simple-CNAPs [1] and TSA [28], which respectively resort to amortized variational inference and gradient-based optimization for estimating the task-specific parameters from the support set. Both are SOTA cross-domain few-shot classification approach on the Meta-dataset [49] benchmark. For more experimental details, hyper-parameter configurations, and additional experimental results, please refer to Appendix B and C.
5.2 Datasets and Evaluation Metrics

**Sinusoids Few-shot Regression** We consider 1D sinusoids regression tasks in the form \( y(x) = A \sin(B(x + C)) \). For ID meta-training, we consider frequency \( B = 1 \), while sample amplitude \( A \) and phase \( C \) uniformly from a set of equally-spaced points \( \{1, 1.1, 1.2, ..., 4\} \) and \( \{0, 0.1, 0.2, ..., 0.5\pi\} \), respectively. Each training task consists of 2 to 5 support and 10 query points with \( x \) uniformly sampled from \( X \in [-5.0, 5.0] \) during training, and we evaluate the models on 500 ID and OOD tasks each with 512 equal-distant query points in \( X \). For ID testing, we expand the range of the tasks by uniformly sampling \( A \in [1, 4] \) and \( C \in [0, 0.5\pi] \). For OOD tasks, we randomly change either the phase distribution to \( C \in [0.6\pi, 0.75\pi] \), amplitude to \( A \in [0.1, 0.8] \cup [4.2, 5.0] \) or frequency to \( B \in [1.1, 1.25] \). Details for the multi-sinusoids regression experiment can be found in Appendix B.1. We use MSE and negative log-likelihood on query samples to evaluate the regression performance.

**Drug Activity Prediction Few-shot Regression** In each task, we aim to predict the drug-target binding affinity of query molecular compounds given 10 to 50 labelled examples from the same domain defined by molecular size. We use the \( l_bap-general-ic50-size \) ID/OOD task split in the DrugOOD [21] benchmark, which divides the molecules into 222/145/23 domains by molecular size for ID Train / ID Test / OOD Test, respectively. The regression performance is evaluated by the square of Pearson coefficient (\( R^2 \)) between predictions and the ground-truth values. We report the mean and median \( R^2 \) on 500 tasks sampled from ID and OOD testing domains.

**Meta-dataset [49] 5-way 1-shot Classification** This experiment considers image classification problems on Meta-dataset [49]. Each task contains up to 10 query images per class from the same domain. Following the current state-of-the-art practice [28, 1], we use Aircraft, dtd, cub, vgg-flower, fungi, quickdraw and omniglot as the ID datasets for meta-training and meta-testing, while traffic, mscoco, cifar10, cifar100 and mnist are treated as OOD datasets for meta-testing only.

**OOD Task Detection Evaluation** We compare the OOD task detection performance of Energy Sum against several model-agnostic OOD detection baselines. Concretely, for classification, we compare against max-softmax score [16], ODIN [30], MAH [27], and max-logits score [15]; for regression, we consider Averaged Bayesian prediction uncertainty in standard deviation (Std) on support samples, and Averaged Support samples Negative Log-Likelihood (SNLL) under model’s task-specific predictive probability, i.e., \(-\mathbb{E}_{\phi \sim q_\phi} \mathbb{E}_{y \mid y_s, \phi, \tau} \mathbb{E}_j [\log p(y_s) \mid x_s, \phi, \tau] \) for baselines and \(-\mathbb{E}_{\phi \sim q_\phi} \mathbb{E}_{y \mid y_s, \phi, \tau} \mathbb{E}_j [\log p(y_s) \mid x_s, \phi, \tau] \) for EBML. Following common practice [12, 16], we report AUROC, AUPR and FPR95 for OOD detection performance. Details for these metrics can be found in Appendix [3.1].

5.3 OOD Detection Results

**Energy sum performs best in OOD task detection.** Table [1] and [5] The proposed energy sum further improves our SNLL-only results in all three OOD detection metrics - with 15.2% and 11.8% significant reduction in FPR95, outperforming the best baseline methods by 20.0% and 39.1%, in single and multi-sinusoids situations respectively. In Table [2] for OOD classification task detection, Energy Sum consistently results in superior OOD detection performance, outperforming the best baselines by large margins of 36.84% and 20.19% in FPR95 for Simple-CNAPs and TSA, respectively.

**Table 1:** OOD task detection performance on single-sine and DrugOOD [21] few-shot regression tasks.

<table>
<thead>
<tr>
<th>OOD Scores</th>
<th>Models</th>
<th>Sinusoids</th>
<th>DrugOOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AUROC↑</td>
<td>AUPR↑</td>
</tr>
<tr>
<td>Std</td>
<td>ABML [41]</td>
<td>50.14</td>
<td>54.80</td>
</tr>
<tr>
<td></td>
<td>F-PACOH-GP [33]</td>
<td>49.52</td>
<td>51.30</td>
</tr>
<tr>
<td></td>
<td>Metafun [53]</td>
<td>76.57</td>
<td>80.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>82.48</td>
<td>81.31</td>
</tr>
<tr>
<td>SNLL</td>
<td>ABML [41]</td>
<td>91.78</td>
<td>93.23</td>
</tr>
<tr>
<td></td>
<td>F-PACOH-GP [33]</td>
<td>95.63</td>
<td>96.46</td>
</tr>
<tr>
<td></td>
<td>Metafun [53]</td>
<td>98.25</td>
<td>97.11</td>
</tr>
<tr>
<td></td>
<td>EBML-CNPs (Ours)</td>
<td>96.46</td>
<td>97.41</td>
</tr>
<tr>
<td>Energy Sum</td>
<td></td>
<td>97.74</td>
<td>98.31</td>
</tr>
</tbody>
</table>
Modelling the joint distribution improves OOD detection under Domain-shift. In Table 1, DrugOOD regression tasks, using either our SNLL or Energy Sum as OOD scores can achieve better detection performance than baselines. In particular, our method outperforms the best OOD detection results obtained using Gaussian SNLL and Std by 43.84% and 11.6% in FPR95, respectively.

Qualitative Illustration. In Figure 5, we visualize the predictive distribution $p(y_{ij}|x_{ij}, \phi_i)$ learned using an EBM decoder and a Gaussian decoder on a sampled ID multi-sinusoids task. The EBM clearly shows two prediction modes at all non-overlapping positions, whereas the Gaussian decoder is unable to model the multi-modality, resulting in a blurry prediction.

Computational Complexity Analysis. We conduct a computational complexity analysis for EBML by comparing its wall-clock training time and convergence to baselines in Figure 6 below. EBML-CNP achieves better OOD detection performance than baseline CNPs meanwhile matching its regression performance at all training epochs. In Table 15 Appendix C.4, we show EBML-CNP is computationally cheaper and faster than traditional Bayesian methods, namely, F-PACOH-GP [43] which requires matrix inversion for inference with Gaussian processes prior, and ABML [41] which imposes a Gaussian prior over the entire parameter space of the model.

Energy sum achieves better OOD detection results when learning the joint distribution. In Table 16, we compare EBML-joint, which is exactly our proposed training procedure in the paper, and we investigate the contribution of the prior EBM in improving the modelling of meta-training task distribution. We train CNPs and ABML using diagonal Gaussian distribution as the prior in ELBO, and compute OOD scores as (a) SNLL, and (b) the sum between SNLL and the NLL of task-specific latent evaluated under the learned Gaussian prior (indicated by +Gauss Prior). The results show that energy sum using an EBM prior outperforms all ablated models. The OOD detection performance of our model benefits from adding the prior EBM energy to the data EBM energy (SNLL), resulting in the most reduction in FPR95 on both single and multi-sinusoids tasks (15.2% and 11.8%, respectively). This suggests the improved expressiveness of EBM over simple distributions can indeed lead to learning a more accurate model of the meta-training ID task distribution.

<table>
<thead>
<tr>
<th>Models</th>
<th>OOD Scores</th>
<th>Sinusoids</th>
<th>AUROC</th>
<th>AUPR</th>
<th>FPR95</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABML [41]</td>
<td>SNLL</td>
<td>94.21</td>
<td>96.34</td>
<td>38.40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+EBM Prior</td>
<td>96.40</td>
<td>98.21</td>
<td>20.40</td>
<td></td>
</tr>
<tr>
<td>EBML-CNP</td>
<td>SNLL</td>
<td>94.00</td>
<td>96.10</td>
<td>34.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+EBM Prior</td>
<td>96.46</td>
<td>97.41</td>
<td>14.20</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6: Left: Wall-clock convergence in seconds, and Right: performance vs number of training epochs, for EBML-CNP vs CNPs in single-sinusoid few-shot regression tasks. The plots show the regression (MSE ↓) and combined OOD tasks detection (1-AUROC)+(1-AUPR)+FPR95 ↓ performance on single sine few-shot regression tasks during training. Curves are moving averages with window size 3. EBML-CNP achieves better final performance than CNPs.

Energy sum achieves better OOD detection results with EMB prior than Gaussian. In Table 3 and 9, we investigate the contribution of the prior EBM in improving the modelling of meta-training task distribution. We train CNPs and ABML using diagonal Gaussian distribution as the prior in ELBO, and compute OOD scores as (a) SNLL, and (b) the sum between SNLL and the NLL of task-specific latent evaluated under the learned Gaussian prior (indicated by +Gauss Prior). The results show that energy sum using an EBM prior outperforms all ablated models. The OOD detection performance of our model benefits from adding the prior EBM energy to the data EBM energy (SNLL), resulting in the most reduction in FPR95 on both single and multi-sinusoids tasks (15.2% and 11.8%, respectively). This suggests the improved expressiveness of EBM over simple distributions can indeed lead to learning a more accurate model of the meta-training ID task distribution.

Table 2: OOD task detection performance on Meta-dataset 5-way 1-shot classification tasks.

<table>
<thead>
<tr>
<th>OOD Scores</th>
<th>Simple-CNAPs [4]</th>
<th>TSA [23]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUROC</td>
<td>AUPR</td>
</tr>
<tr>
<td>max-softmax [23]</td>
<td>85.50</td>
<td>85.54</td>
</tr>
<tr>
<td>max-logits</td>
<td>50.00</td>
<td>70.83</td>
</tr>
<tr>
<td>MAH</td>
<td>96.49</td>
<td>89.42</td>
</tr>
<tr>
<td>Domain Classifier</td>
<td>71.71</td>
<td>69.76</td>
</tr>
<tr>
<td>EBML Energy Sum</td>
<td>83.10</td>
<td>73.18</td>
</tr>
</tbody>
</table>

Table 3: Ablation study on Energy Sum for OOD detection on single-sinusoids.
This paper proposes a new energy-based meta-learning (EBML) framework for the first time, which directly characterizes any arbitrary meta-training task distribution using two data and prior energy functions. EBML is compatible with many existing SOTA meta-learning algorithms and allows both detection and adaption of OOD tasks. The sum of the two learned energy functions gives an unnormalized probability distribution proportional to the underlying task likelihood, deployable as OOD scores. The experiment results show the superiority of Energy Sum over traditional methods in detecting both OOD regression and classification tasks, and the possibility of achieving improved OOD adaptation performance with EBML through minimizing the task energy. One limitation of EBML is that our current OOD task adaption strategy does not consider the effect of negative transfer, as some OOD tasks may benefit from adapting from scratch without ID energy prior regularization. Thus, in future works, we are interested in designing task-specific adaptation strategies for EBML that can selectively adapt OOD tasks for better performance.
References


# Contents

1 Introduction  

2 Related Work  

3 Preliminaries: Energy-based Models  

4 Energy-Based Meta-learning  
   4.1 Energy-based Modelling of Task Distribution  
   4.2 EBML for OOD Detection  
   4.3 EBML for OOD Generalization  

5 Experiments  
   5.2 Datasets and Evaluation Metrics  
   5.3 OOD Detection Results  
   5.4 OOD Generalization Results  

6 Conclusion and Limitation  

A Derivation for EBML  
   A.1 Derivation for Training Objective in Eqn. (8)  
   A.2 Derivation for Energy Sum in Eqn. (9)  
   A.3 Derivation for Eqn. (11) as an Approximation to Bayesian Posterior Inference  

B Experiment Details  
   B.1 OOD Detection Evaluation Metrics  
   B.2 Single and Multi-sinusoid Regression  
   B.3 Drug Activity Prediction  
   B.4 Meta-dataset Few-shot Classification  
   B.4.1 Details of EBML-SimpleCNAPs and EBML-TSA  
   B.4.2 Baseline OOD Detection Methods for Classification  

C Additional Experiment Results  
   C.1 Multi-sinusoids Few-shot Regression and OOD Task Detection  
   C.2 Meta-dataset 5-way-1-shot Classification and OOD Task Detection  
   C.2.1 TSA-EBML vs TSA on Unshuffled 5-way-1-shot Meta-dataset Tasks  
   C.2.2 5-way-1-shot OOD classification results in Meta-dataset [49] for more EBML variants  
   C.2.3 Classification Accuracy Trajectories During OOD Task Adaptation  
   C.2.4 More OOD Adaptation Baselines on Meta-dataset  
   C.2.5 Baseline OOD Adaptation is Less Reliable without Sufficient Support Samples  
   C.3 EBML with Probabilistic Posterior Distribution $q_\psi(\phi_1 | Y_1)$  

---

Page 15
A Derivation for EBML

A.1 Derivation for Training Objective in Eqn. (8)

We start the derivation from maximizing the ELBO (Eqn. (5)) for a single training task \( T_i \) w.r.t. the parameters of the EBMs and the task latent posterior distribution, i.e., \( \lambda, \omega \) and \( \psi \).

\[
\arg \max_{\lambda, \psi, \omega} \log p(T_i) \overset{\text{Eqn. (5)}}{=} \arg \max_{\lambda, \psi, \omega} \mathbb{E}_{\phi_i \sim q_\psi(\phi_i | T_i)} [\log p_\omega(X_i, Y_i | \phi_i)] - \text{KL}(q_\psi(\phi_i | T_i) || p_\lambda(\phi_i))
\]

where \( q_\psi, p_\omega \) and \( p_\lambda \) denote the model distributions of the latent posterior, the data EBM and the prior EBM, respectively. Recall \( p_\lambda \) and \( p_\omega \) are EBMs where \( p_\omega(X_i, Y_i | \phi_i) = \prod_j \exp \left( \frac{-E_{\omega}(x_i, y_j, \phi_i)}{Z(\omega, \phi_i)} \right) \) and \( p_\lambda = \exp \left( \frac{-E_{\lambda}(\phi_i)}{Z(\lambda)} \right) \). We rewrite the KL term as

\[
-\text{KL}(q_\psi(\phi_i | T_i) || p_\lambda(\phi_i)) = -E_{q_\psi(\phi_i | T_i)} \left[ \log \frac{q_\psi(\phi_i | T_i)}{p_\lambda(\phi_i)} \right]
\]

\[
\quad = -E_{q_\psi(\phi_i | T_i)} \left[ \log q_\psi(\phi_i | T_i) - E_{q_\psi(\phi_i | T_i)} \left[ \frac{1}{\log p_\lambda(\phi_i)} \right] \right]
\]

\[
\quad = \mathcal{H}(q_\psi(\phi_i | T_i)) + E_{q_\psi(\phi_i | T_i)} [\log p_\lambda(\phi_i)]
\]

The two log-likelihood terms for EBMs, (I) and (II), can be rewritten using the learning gradient in Eqn. (4), i.e.,

\[
\mathbb{E}_{\phi_i \sim q_\psi(\phi_i | T_i)} [\log p_\omega(X_i, Y_i | \phi_i)] \overset{\text{Eqn. (4)}}{=} \mathbb{E}_{\phi_i \sim q_\psi(\phi_i | T_i)} \left[ \sum_j -E_{\omega}(x_{ij}, y_{ij}, \phi_i) \right]
\]

\[
\quad + \mathbb{E}_{p_\omega(x', y' | \phi_i)} [E_{\omega}(x'_{ij}, y'_{ij}, \phi_i)]
\]

\[
\mathbb{E}_{q_\psi(\phi_i | T_i)} [\log p_\lambda(\phi_i)] \overset{\text{Eqn. (4)}}{=} \mathbb{E}_{q_\psi(\phi_i | T_i)} \left[ -E_{\lambda}(\phi_i) + E_{p_\lambda(\phi_i)} [E_{\lambda}(\phi_i')] \right]
\]

Combining with the entropy term in (III) and take the expectation w.r.t. the training task distributing \( \mathcal{P}_{1D} \) we have our training objective in Eqn. (8).

A.2 Derivation for Energy Sum in Eqn. (9)

The log-likelihood of a task \( T_i \) writes

\[
\log p(T_i) = \log \prod_{j=1}^{N_i} p(x_{ij}, y_{ij} | \phi_i) p_\lambda(\phi_i) \ d\phi_i:
\]

\[
\quad \geq E_{\phi_i \sim q_\psi(\phi_i | T_i)} [\log p_\omega(X_i, Y_i | \phi_i)] - \text{KL}(q_\psi(\phi_i | T_i) || p_\lambda(\phi_i))
\]

\[
\quad = E_{q_\psi(\phi_i | T_i)} [\log p_\omega(X_i, Y_i | \phi_i)] + E_{q_\psi(\phi_i | T_i) [\log p_\lambda(\phi_i)] + \mathcal{H}(q_\psi(\phi_i | T_i))]
\]

which is lower-bounded by the ELBO in Eqn. (20) characterized by the learned \( q_\psi, p_\omega \) and \( p_\lambda \). The ELBO is proportional to the sum of two energy functions and the entropy of the posterior distribution \( q_\psi \), all of which can be easily calculated via feed-forward passes of the training samples.

Since the majority of the state-of-the-art meta-learning algorithms (including CNP [8] and SimpleC-NAPS [1]) adopt the MAP estimation of the posterior \( q_\psi \) which is deterministic, the entropy essential becomes zero, and the expectations \( E_{q_\psi(\phi_i | T_i)} \) in the first and second terms in Eqn. (20) simplify to energy function evaluation at \( X_i, Y_i \) and \( \phi_i \), respectively. Finally, we negate Eqn. (20) so that the OOD scores for in-distribution tasks are lower than that of the out-of-distribution ones.
A.3 Derivation for Eqn. (11) as an Approximation to Bayesian Posterior Inference

Given a new test task $\mathcal{T}_i$, Bayesian inference aims to infuse the meta-learned prior knowledge with the observed support set $\mathbf{X}_s^i$, $\mathbf{Y}_s^i$ for inferring a small set of unknown task-specific parameters $\zeta$. This is akin to maximizing the log-likelihood of the support set w.r.t. the task-specific parameters $\zeta$ which defines the posterior energy approximates the minimization of the KL-divergence between the task-specific posterior inference objective as

$$\arg\min_{\zeta} E_{q_{\psi}} [\log p_{\lambda,\psi}(\mathcal{T}_s^i)] - KL(q_{\psi}(\phi_s| \mathcal{T}_s^i)||p_{\lambda}(\phi_s)).$$

(22)

Next, we introduce the task-specific parameter $\zeta$ in the latent posterior and formulate the Bayesian posterior inference objective as

$$\arg\min_{\zeta} E_{q_{\psi}} [\log p_{\lambda,\psi}(\mathcal{T}_s^i)] - KL(q_{\psi}(\phi_s| \mathcal{T}_s^i)||p_{\lambda}(\phi_s)).$$

(23)

Assuming a maximum a posterior (MAP) estimate of the task-specific latent distribution which approximates $q_{\psi|\zeta}(\phi_s| \mathcal{T}_s^i)$ by a Dirac-delta function $q_{\psi}(\phi_s| \mathcal{T}_s^i) = \delta(\phi_s - \phi_s | \mathcal{T}_s^i)$. As a result, the second KL term reduces to a likelihood evaluation, i.e., $KL(q_{\psi}(\phi_s| \mathcal{T}_s^i)||p_{\lambda}(\phi_s)) = E_{q_{\psi}(\phi_s| \mathcal{T}_s^i)} [\log q_{\psi}(\phi_s| \mathcal{T}_s^i)] = -\log p_{\lambda}(\phi_s) = E_{\lambda}(\phi_s) + \log Z(\lambda)$. Since the (log-)partition function $\log Z(\lambda)$ is a constant w.r.t. the arg min parameter $\zeta$, we then have $\arg\min_{\zeta} KL(q_{\psi}(\phi_s| \mathcal{T}_s^i)||p_{\lambda}(\phi_s)) = \arg\min_{\zeta} E_{\lambda}(\phi_s)$. From here, we see that minimizing the task prior energy approximates the minimization of the KL-divergence between the task-specific posterior $q_{\psi|\zeta}$ and the meta-learned ID prior $p_{\lambda}$ in the Bayesian posterior inference objective, thus it acts as a meta-regularizer to combat over-fitting in adaptation. In practice, we found that using a margin loss for this prior energy minimization, i.e., $\arg\min_{\zeta} \max(E_{\lambda}(\phi_s) - m, 0)$, can yield better empirical performance.

While for the first log-likelihood term inside the expectation, $-\sum_{j=1}^{N_s^i} \log p_{\omega}(\mathbf{x}_{ij}, y_{ij}|\phi_s)$ which is equivalent to the sum of data energy scores $\sum_{j=1}^{N_s^i} [E_{\omega}(\mathbf{x}_{ij}, y_{ij}, \phi_s) + \log Z(\omega, \phi_s)]$, we use the decoder $\omega_2$ with the task-specific prediction loss i.e., cross-entropy loss, in the base meta-learning algorithm as a tractable surrogate, as discussed in implementation of the Experiment section [5] and also in Appendix [B.4.1] This thus maintains the data-level predictive ability of model during adaptation.
B Experiment Details

B.1 OOD Detection Evaluation Metrics

Conventionally \cite{17, 16, 27}, OOD detection is treated as a binary classification problem in which the trained detector is expected to assign a positive label for an OOD task if its estimated OOD score exceeds some threshold $\tau$. To evaluate the performance of the OOD detector, we use three metrics: area under the receiver operating characteristic curve (AUROC), area under the precision-recall curve (AUPR), and the false positive rate at N% true positive rate (FPRN), where N=95 in our experiments.

As discussed in \cite{17}, the AUROC and AUPR are holistic metrics that summarize the performance of a detection method across multiple thresholds. The AUROC can be thought of as the probability that an OOD example is given a higher OOD score than an ID example. Thus, a higher AUROC is better, and an uninformative detector has an AUROC of 50%. The AUPR is useful when OOD inputs are infrequent, as it takes the base rate of OOD inputs into account.

Whereas the previous two metrics represent the detection performance across various thresholds, the FPRN metric represents performance at one strict threshold. By observing performance at a strict threshold, we can make clear comparisons among strong detectors. The FPRN metric is the probability that an in-distribution example (ground-truth negative sample) raises a false alarm (detected as a positive sample) when N% of ground-truth OOD examples (positive samples) are correctly detected, so a lower FPRN is better. Capturing nearly all anomalies with few false alarms can be of high practical value.

B.2 Single and Multi-sinusoid Regression

Model Architecture Both the the data EBM $E_\omega(x, y, \phi_i)$ and the prior EBM are MLPs. Follow the encoder implementation in CNPs \cite{8}, $q_\psi(\phi_i | T_s) = \text{MLP}_1(\frac{1}{N} \sum_{j=1}^{N_s} \text{MLP}_2([x_{sij}, y_{sij}]))$. The output of $q_\psi$ is the task latent variable $\phi_i \in \mathbb{R}^2$.

Hyperparameters We use a training batch size of 50 and learning rate of 0.0005 for all methods. The additional method-specific hyperparameters are stated below

Metafun \cite{53} num-inner-loop: 5; initial representation: zero; outer learning rate: $10^{-4}$; initial inner learning rate: 0.1; Dropout rate: 0.0; Orthogonality penalty weight: 0.0; L2 penalty weight: 0.0.

MAML \cite{6} batch size: 4; num-inner-loop: 5; inner learning rate: 0.01; outer learning rate: 0.001;

ABML \cite{41} batch size: 4; num-inner-loop: 5; inner learning rate: 0.005; outer learning rate: 0.001; alpha 1.0; beta 0.01; num reparameterizatio samples: 4;

F-POACH-GP \cite{43} prior outputscale: 2.0; prior lengthscale: 0.2; prior weight: 0.001; learnable prior mean: True; learnable prior covariance: True.

EBML-CNPs Prior SGLD $\eta$: 0.01; Data SGLD $\eta$: 0.001; num-SGLD-iter: 20; energy L2 penalty 1.0;

B.3 Drug Activity Prediction

Preprocessing Molecular Graph Inputs Each input $x$ is a SMILES representation of a chemical molecule, which essentially is a list of string characters of variable length. To transform the SMILE representation into numerical values, we use a pre-trained SMILES-transformer \cite{18} for converting the sring input $x$ into vector representation $\tilde{x}$ in $\mathbb{R}^{1024}$. We treat this $\tilde{x}$ as the inputs for all methods.

Model Architecture We use the same EBML-CNPs architecture as in the sinusoids experiments. However, we expand the latent variable dimension to $\mathbb{R}^{128}$, and the number of neurons in each hidden layer to 256. Furthermore, additional Batch Normalization layers \cite{20} are interleaved with layers of the MLPs.

Hyperparameters We use a training batch size of 10 and learning rate of 0.0005 for all methods. The additional method-specific hyperparameters are stated below
Metafun \([53]\) num-inner-loop: 5; initial representation: zero; outer learning rate: \(10^{-4}\); initial inner learning rate: 0.1; Dropout rate: 0.0; Orthogonality penalty weight: 0.0; L2 penalty weight: 0.0.

MAML \([6]\) batch size: 4; num-inner-loop: 5; inner learning rate: 0.001; outer learning rate: 0.001;

ABML \([41]\) batch size: 4; num-inner-loop: 5; inner learning rate: 0.001; outer learning rate: 0.001; alpha 1.0; beta 0.01; num reparameterizatio samples: 4;

F-POACH-GP \([43]\) prior outputscale: 1.0; prior lengthscale: 0.5; prior weight: 0.001; learnable prior mean: True; learnable prior covarance: True.

EBML-CNPs Prior SGLD \(\eta\): 0.1; Data SGLD \(\eta\): 0.1; num-SGLD-iter: 40; energy L2 penalty 0.1;

B.4 Meta-dataset Few-shot Classification

B.4.1 Details of EBML-SimpleCNAPs and EBML-TSA

Model Architecture  TSA \([28]\) pre-trains a feature representation using available ID training domains, and incooperates additional task-specific adaptation modules at test time in the form of residual-connected transformation matrices to each convolution block. The parameters of these modules are inferred by gradient descent on the support set from scratch at meta-testing. The transformed feature representations of the support set samples are then used to build the class prototypes in a non-parametric classifier for inference of the query sample labels. SimpleCNAPs \([11]\) also first pre-trains a feature extractor on a large dataset, i.e., ImageNet. However, unlike TSA, SimpleCNAPs meta-learns task-specific adaptations during meta-training by learning a parametrized task-encoding function that estimates the task-specific modules in the form of additional FILM parameters from the support set. Similar to TSA, the adapted support set features are used to construct the class centers in a non-parametric predictive function for classification of the query samples.

Thus for both methods, the set of prototypes in each ID training task resemble a task-specific predictive function, and is a suitable choice as the meta-learned prior knowledge. By specifying the latent variable \(\phi\) to be a set of class prototypes used in the cosine classifier of each ID meta-training task, our EBM prior \(p_{\lambda}(\phi)\) resembles a distribution over task-specific predictive functions from the ID domains. The architecture for the Prior EBM is depicted in Figure 7.

![Figure 7: Model architecture for the prior EBM for classification.](image)

To simultaneously achieve the best of both classification and OOD detection, we follow similar strategies in \([50, 37]\) which learn an another decoder for prediction. The modified EBML architecture is shown in Figure 8.

Hyper-parameters  We use the reported hyperparameters in TSA \([28]\) and SimpleCNAPs \([11]\) for training the base models. We report here the additional hyperparameters specific to EBML below in Table 6 which are found on the validation split of Meta-dataset \([49]\).
Figure 8: Overview of the EBML framework for image classification tasks. The task latent variable $\phi_i$ are inferred from the support set $T_i^s$ following the implementation of the base algorithm. The data and task energy scores are evaluated by the data and prior EBMs $E_\omega$ and $E_\lambda$, respectively; while the query labels are predicted by the classifier $p_{\omega_2}$ of the base algorithm. The feature extractor $f$ is a pre-trained ResNet-18 identical to the one used in SimpleCNAPs [1] and TSA [28].

Table 6: Hyperparameters for EBML on Meta-dataset 5-way 1-shot classification tasks.

<table>
<thead>
<tr>
<th>Groups</th>
<th>Hyperparameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>num SGLD steps</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>Data SGLD $\eta$</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>Prior SGLD $\eta$</td>
<td>10.0</td>
</tr>
<tr>
<td></td>
<td>Energy L2 penalty</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>EBM Spectral Norm</td>
<td>True</td>
</tr>
<tr>
<td>Adaptation</td>
<td>num steps</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>TSA learning rate $\beta$</td>
<td>0.00091</td>
</tr>
<tr>
<td></td>
<td>TSA learning rate $\alpha$</td>
<td>0.000267</td>
</tr>
<tr>
<td></td>
<td>weight on prior energy</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>$m$ in prior energy margin loss</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>optimizer</td>
<td>Adam</td>
</tr>
</tbody>
</table>

B.4.2 Baseline OOD Detection Methods for Classification

We used the following traditional OOD input detection methods as baselines for computing the task OOD scores in Table 2 main body of the paper. We compute the task OOD score for a task $T_i$, as the average instance-level OOD scores over its input images $\{x_i\}^{N_i}$. More specifically:

max softmax score [16], is taken as the maximum softmax prediction probability over $N$ possible classes, that is $S_{\hat{y}}(x) = \max_c \exp(\logit_c(x)) / \sum_{c=0}^{N-1} \exp(\logit_c(x))$.

ODIN [30], extends the softmax-score by introducing temperature scaling and input pre-processing. More concretely, the input is perturbed following $\hat{x} = x + \epsilon \text{sign}(\nabla_x \log S_{\hat{y}}(x; T))$, which moves $x$ in the direction that increases the temperature-scaled softmax score $\exp(\logit_c(\hat{x})/T)$. The final ODIN confidence score writes $S_{\hat{y}}(\hat{x}; T)$.

max logits score [15]. This is simply the maximal prediction logit of input image $x$, which is an alternative to the max-softmax prediction score.

MAH Detector [27]. We compute the MAH OOD score of $x$ as the Mahalanobis distance from its feature representation to its nearest class mean which we estimate for each class using the empirical average of training inputs in class $c$. The shared covariance matrix used in estimating the Mahalanobis distance is computed on a subset of training samples from all training classes.

Domain Selector. When given the domain-IDs during training, a few methods [34] on Meta-dataset classification problems adapt a domain classifier network for inferring the task-specific parameters from a set of candidate domain-specific feature modulation parameters. The max softmax score of the trained domain selector can be used to compute the OOD score for $x$.
The perturbation magnitude $\epsilon$ and the temperature scale $T$ used in ODIN and MAH are determined using the validation set of meta-dataset, with a grid-search over the parameter space for $\epsilon \in \{0, 0.00002, 0.00005, 0.0001, 0.0002, 0.0005, 0.001, 0.002, 0.005\}$ and $T \in \{1, 2, 10, 50, 100, 200, 500\}$. ODID and MAH on average improves the OOD task detection results using max softmax and max logits scores for the same baseline model.
C Additional Experiment Results

C.1 Multi-sinusoids Few-shot Regression and OOD Task Detection

Task generation The multi-target regression tasks are synthesised by superposing each generated sinusoid from the ID and OOD distribution with a phase-shifted version of itself at a constant phase lag of 0.3π, such that now in each task every input x has two possible target values y. We give both y values for each x in the support set.

Results In Table 7, we see that EBML-CNPs achieved both the best regression and OOD detection performance. In Table 8, our proposed energy outperform all ablated models.

In Table 9, we give both y values for each x in the support set. This explains the observed differences between TSA and EBML-TSA on OOD detection tasks. In this section, we verify that the improved classification performance of EBML-TSA over TSA is indeed not a result of the change in evaluation protocols in Meta-dataset. To do so, we evaluate EBML-TSA under identical settings to TSA in Table 8 of [28], i.e., 5-way-1-shot settings with shuffle_buffer_size = 0.

In Table 8, we see that EBML-CNPs achieved both the best regression and OOD detection performance.

C.2 Meta-dataset 5-way-1-shot Classification and OOD Task Detection

C.2.1 TSA-EBML vs TSA on Unshuffled 5-way-1-shot Meta-dataset Tasks

To ensure our few-shot classification results in Table 5 are fair and up-to-date, we follow the latest evaluation protocols in Meta-dataset which sets shuffle_buffer_size=1000, and test TSA (reproduced using their official code) and EBML-TSA on the same set of sampled testing tasks. However, the official 5-way-1-shot classification results of TSA (Table 8 in [28]) are reported on an earlier version of Meta-dataset before the fix of issue 54. This explains the observed differences between TSA results in Table 5 and Table 8 in [28].

In this section, we verify that the improved classification performance of EBML-TSA over TSA is indeed not a result of the change in evaluation protocols in Meta-dataset. To do so, we evaluate EBML-TSA under identical settings to TSA in Table 8 of [28], i.e., 5-way-1-shot settings with shuffle_buffer_size = 0.

In Table 10, we compare our results with the official results of TSA. The performance of both methods are largely similar to that in Table 5, except that the classification accuracy for

\*As mentioned in https://github.com/google-research/meta-dataset/issues/54, the shuffle_buffer_size was set to zero in an earlier version of Meta-dataset which can lead to some biased results in evaluation.
Table 11: 5-way-1-shot classification accuracy of OOD tasks in Meta-datasets [49].

<table>
<thead>
<tr>
<th>Datasets</th>
<th>SimpleCNAPs [1] (Ours)</th>
<th>EBML-SimpleCNAPs [29] (Ours)</th>
<th>URL [29] (Ours)</th>
<th>EBML-URL (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSCOCO</td>
<td>49.37 ± 0.99</td>
<td>51.75 ± 0.96</td>
<td>59.14 ± 0.95</td>
<td>59.82 ± 0.97</td>
</tr>
<tr>
<td>Traffic Sign</td>
<td>55.63 ± 0.96</td>
<td>56.12 ± 0.97</td>
<td>57.62 ± 0.90</td>
<td>58.26 ± 0.90</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>50.79 ± 0.85</td>
<td>51.16 ± 0.89</td>
<td>54.37 ± 0.83</td>
<td>54.39 ± 0.86</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>54.15 ± 0.95</td>
<td>55.23 ± 0.93</td>
<td>63.03 ± 0.97</td>
<td>62.74 ± 0.96</td>
</tr>
<tr>
<td>MNIST</td>
<td>80.25 ± 0.85</td>
<td>81.01 ± 0.85</td>
<td>78.85 ± 0.86</td>
<td>79.78 ± 0.87</td>
</tr>
</tbody>
</table>

Both methods improved on a few datasets e.g., MNIST, MSCOCO. Noticeably, EBML-TSA still outperforms TSA, on 5/7 ID domains (2/5 equal performance) with an average increase of 0.4% in accuracy, and 5/5 OOD domains with an average improvement of 1.0%. The results validate the effectiveness of our proposed method and verify that our methods indeed is not favoured by the latest evaluation protocol in Meta-dataset.

C.2.2 5-way-1-shot OOD classification results in Meta-dataset [49] for more EBML variants

We instantiate EBML with two additional baseline Meta-learning algorithms including SimpleCNAPs [1] and URL [29] and report their 5-way-1-shot classification accuracy of OOD tasks in Meta-datasets [49] in Table 11 above. OOD-adaptation for EBML is performed by optimizing Eqn. (11) w.r.t. to the task encoder that produces the task-specific FiLM in SimpleCNAPAS, and w.r.t. the feature projection matrix in URL.

C.2.3 Classification Accuracy Trajectories During OOD Task Adaptation

In Figure 9 and 10, we visualize the average query set classification accuracy throughout the OOD task adaptation for TSA and EBML-TSA. Results for TSA are produced using the official optimal hyperparameters reported in [28]; while for EBML-TSA, we use the hyperparameters reported in Table 6.

The y-axis in plot represents the average classification accuracy on the query set, while the x-axis represents the steps during OOD adaptation. We observe that our objective (Blue) in Eqn. (11) generally alleviates the over-fitting behaviour in the adaptation process caused by minimizing the cross-entropy loss alone in TSA (Orange). This meta-regularization effect is more apparent on tasks from the OOD domains.

Figure 9: TSA vs EBML-TSA query set classification accuracy during adaptation on ID datasets

C.2.4 More OOD Adaptation Baselines on Meta-dataset

In the experiment on Meta-dataset 5-way 1-shot classification tasks, we assigned pseudo-labels to the query inputs and used them together with the labelled support set samples in calculation of the class prototypes, hence the prior energy score for each task, in Eqn. (11). Therefore, In this study, we compare our results with several baselines that also utilize the unlabelled query set in for adaptation, including:
Query Entropy (TSA+QE), which minimizes the average entropy of the prediction distribution on the query samples in addition to the support set classification loss.

Confidently-predicted Pseudo-labels (TSA+PL), which tunes a confidence threshold, assigns query predictions over the threshold as the pseudo-labels, and then minimizes the classification loss on the support set and these confidently-predicted query samples before testing.

The optimal results are reported in Table 12. The results show that while the two baseline methods exploiting the query set information can achieve better performance than TSA on some datasets, they do not outperform EBML-TSA with using Eqn. (11) for task adaptation.

Table 12: Additional 5-way 1-shot classification results when using TSA with 1) entropy minimization on the query set, and 2) cross-entropy on confidently-predicted pseudo-labelled query samples, for OOD task adaptation.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>EBML-TSA</th>
<th>TSA+EM</th>
<th>TSA+PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Omniglot</td>
<td>76.35 ± 0.26</td>
<td>76.54 ± 0.26</td>
<td>76.78 ± 0.26</td>
</tr>
<tr>
<td>Textures</td>
<td>52.35 ± 0.88</td>
<td>52.35 ± 0.88</td>
<td>52.35 ± 0.88</td>
</tr>
<tr>
<td>Aircraft</td>
<td>74.70 ± 0.86</td>
<td>74.70 ± 0.86</td>
<td>74.70 ± 0.86</td>
</tr>
<tr>
<td>Birds</td>
<td>75.52 ± 0.90</td>
<td>75.52 ± 0.90</td>
<td>75.52 ± 0.90</td>
</tr>
<tr>
<td>VGG Flower</td>
<td>80.30 ± 0.83</td>
<td>80.30 ± 0.83</td>
<td>80.30 ± 0.83</td>
</tr>
<tr>
<td>Fungi</td>
<td>72.29 ± 0.94</td>
<td>72.29 ± 0.94</td>
<td>72.29 ± 0.94</td>
</tr>
<tr>
<td>Quickdraw</td>
<td>80.27 ± 0.85</td>
<td>80.27 ± 0.85</td>
<td>80.27 ± 0.85</td>
</tr>
<tr>
<td>MSCOCO</td>
<td>55.03 ± 0.97</td>
<td>55.03 ± 0.97</td>
<td>55.03 ± 0.97</td>
</tr>
<tr>
<td>Traffic Sign</td>
<td>58.85 ± 1.01</td>
<td>58.85 ± 1.01</td>
<td>58.85 ± 1.01</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>50.04 ± 0.89</td>
<td>50.04 ± 0.89</td>
<td>50.04 ± 0.89</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>62.77 ± 0.95</td>
<td>62.77 ± 0.95</td>
<td>62.77 ± 0.95</td>
</tr>
<tr>
<td>MNIST</td>
<td>76.08 ± 0.88</td>
<td>76.08 ± 0.88</td>
<td>76.08 ± 0.88</td>
</tr>
<tr>
<td>Avg ID</td>
<td>76.84</td>
<td>76.86</td>
<td>76.86</td>
</tr>
<tr>
<td>Avg OOD</td>
<td>60.15</td>
<td>59.46</td>
<td>59.38</td>
</tr>
<tr>
<td>Avg All</td>
<td>69.89</td>
<td>68.16</td>
<td>69.28</td>
</tr>
</tbody>
</table>

C.2.5 Baseline OOD Adaptation is Less Reliable without Sufficient Support Samples

We intend to use this experiment as an empirical evidence to support our argument on that SOTA cross-domain meta-learning algorithms produce unreliable task-specific adaptation without sufficient support set samples. We train Simple-CNAPs [1] and TSA [28] following the official experimental setup of Meta-dataset [49], except that we have excluded ImageNet in the ID training datasets due to limited computation resources.

Following [1][28], in the varying-way varying-shot testing configuration, the number of classes in each task varies between 5 to 50, while the total number of support samples per task varies between 5 to 500. The maximal number of support samples per class is capped at 100. We report the average testing accuracy over 600 tasks for the varying-way varying shot and 5-way 1-shot settings in Table [13] below. We observe that the average classification accuracy for both ID and OOD domains generally
decrease for 5-way 1-shot tasks, which suggests that the adaptation with without sufficient support set samples is inherently difficult.

Table 13: Classification accuracy of SimpleCNAPs and TSA on meta-dataset for varying-way varying-shot vs 5-way 1-shot meta-testing configurations.

<table>
<thead>
<tr>
<th>Varying-Way Varying-Shot</th>
<th>5-Way 1-Shot</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Datasets</strong></td>
<td><strong>SimpleCNAPs</strong></td>
</tr>
<tr>
<td>Omniglot</td>
<td>91.61 ± 0.57</td>
</tr>
<tr>
<td>Textures</td>
<td>65.47 ± 0.71</td>
</tr>
<tr>
<td>Aircraft</td>
<td>81.28 ± 0.69</td>
</tr>
<tr>
<td>Birds</td>
<td>73.80 ± 0.81</td>
</tr>
<tr>
<td>VGG Flower</td>
<td>90.15 ± 0.50</td>
</tr>
<tr>
<td>Fungi</td>
<td>44.82 ± 1.13</td>
</tr>
<tr>
<td>Quickdraw</td>
<td>73.30 ± 0.81</td>
</tr>
<tr>
<td>MSCOCO</td>
<td>35.19 ± 0.94</td>
</tr>
<tr>
<td>Traffic Sign</td>
<td>42.59 ± 1.01</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>56.65 ± 0.80</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>44.13 ± 1.10</td>
</tr>
<tr>
<td>MNIST</td>
<td>93.89 ± 0.37</td>
</tr>
<tr>
<td>Avg ID</td>
<td>74.35</td>
</tr>
<tr>
<td>Avg OOD</td>
<td>54.49</td>
</tr>
<tr>
<td>Avg All</td>
<td>66.07</td>
</tr>
</tbody>
</table>

C.3 EBML with Probabilistic Posterior Distribution $q_\psi(\phi_i | T^*_i)$

Since EBML is a flexible plug-in, it is definitely compatible with those methods that use probabilistic posterior $q_\psi$. However, we currently mainly focus on the MAP estimate in the main paper because the majority of the state-of-the-art algorithms [8, 28] that EBML has applied to resort to a MAP estimate for $q_\psi$.

Nevertheless, in the Table 14 below, we show the results of EBML with Neural Processes (NPs) [9], named EBML-NPs, on sine regression tasks. NPs parameterizes $q_\psi$ as a multivariate Gaussian distribution whose task-specific mean and standard deviation are determined by a learnable neural network encoder conditioned on the task support set.

For training EBML-NPs, we include the entropy term $\mathcal{H}(q_\psi(\phi_i | T^*_i))$ in the training objective in Eqn. (8), which has a closed-form solution when $q_\psi$ is a Gaussian. We resort to the re-parameterization trick for computing the expectations $E_{q_\psi}$ in Eqn. (8). When using energy sum for OOD detection, we also add the entropy term $\mathcal{H}(q_\psi(\phi_i | T^*_i))$ for EBML-NPs for consistency.

Table 14: Regression and OOD detection performance on sinusoids few-shot regression tasks for EBML-NPs vs EBML-CNPs.

<table>
<thead>
<tr>
<th>OOD Scores</th>
<th><strong>EBML-NPs</strong></th>
<th><strong>EBML-CNPs</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>MSE</strong></td>
<td><strong>AUROC</strong></td>
</tr>
<tr>
<td>SNLL</td>
<td>95.94</td>
<td>97.16</td>
</tr>
<tr>
<td>Energy Sum w/ Entropy</td>
<td>97.10</td>
<td>97.97</td>
</tr>
<tr>
<td>Energy Sum w/o Entropy</td>
<td>97.58</td>
<td>97.79</td>
</tr>
</tbody>
</table>

In Table 14 we conclude that EBML-NPs with probabilistic $q_\psi$ achieves comparable performance to its deterministic version, EBML-CNPs; there is no significant improvement in performance by switching to a probabilistic $q_\psi$. Nevertheless, we observe that both the prior energy function and the extra entropy terms bring in positive contribution to the OOD detection performance compared to SNLL alone.

C.4 Computational Complexity Analysis

We conduct a computational complexity analysis for EBML by comparing its wall-clock training time and convergence to baselines, results are shown above in Table 15 and Figure 11.
Table 15: Training time in seconds. * ABML is much slower due to the gradient-based inner loop optimizations and learning with a Bayesian Neural Network, which makes it challenging to parallelize training over a batch of tasks.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Training time / 500 tasks</td>
<td>0.67</td>
<td>1.83</td>
<td>3.13</td>
<td>11.95*</td>
</tr>
<tr>
<td>Meta-dataset</td>
<td>SimpleCNAPs [1]</td>
<td>EBML-SimpleCNAPs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training time / 1000 tasks</td>
<td>1.02</td>
<td>1.75</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 11: Left: Wall-clock convergence in seconds, and Right: performance vs number of training epochs, for EBML-CNPs vs CNPs in single-sinusoid few-shot regression tasks. The plots show the regression (MSE ↓) and combined OOD tasks detection (1-AUROC)+(1-AUPR)+FPR95 ↓ performance on single sine few-shot regression tasks during training. Curves are moving averages with window size 3. EBML-CNPs achieves better final performance than CNPs.

From the results above, EBML is computationally cheaper and faster than the two Bayesian methods, namely, F-PACOH-GP [43] which requires matrix inversion for inference with Gaussian processes prior, and ABML [41] which imposes a Gaussian prior over the entire parameter space of the model. Meanwhile, in Table 4 and 1 in the paper, EBML achieves the best regression and OOD detection performance out of all baselines.
D Empirical Study on Distribution Shift in the Input Space

In real-world applications, distribution shift in input space, i.e. the distribution shift in $X$, is a very common phenomenon. Take AI-aided drug discovery as an example, when predicting the bio-activities of a molecule for a given target protein, we may encounter molecules with very different molecular sizes, scaffolds etc, from the training examples. Such input distribution shift, unfortunately, cannot always be correctly reflected by models trained to maximize the predictive probability $p_\omega(y|x, \phi_i)$.

We first conduct a controlled experiment and show that modelling the joint distribution can lead to superior performance in OOD task detection which further substantiate our claim. We base our experiment on drug activity prediction tasks as described in Section 5, where $p(x)$ changes across tasks. The experimental details are as follow.

Setup There are three major factors affecting OOD detection performance, including a) whether we model the conditional or joint distribution, b) the model capacity, e.g., Gaussians or EBMs, and c) OOD scores, e.g., energy sum or sum of negative log-likelihood (SNLL) of the support samples. To investigate the effect of (a) specifically, we fix the controlled variables (b) with the same EBM architectural capacity, and (c) with either energy sum or SNLL. Consequently, we compare EBML-joint, which is exactly our proposed training procedure in the paper, and EBML-conditional, which follows the same training with EBML-joint but models $p(Y|X)$ instead of $p(X, Y)$ of the meta-training task distribution. Concretely, the training objective for EBML-conditional becomes

$$\arg \max_{\omega, \lambda, \psi} \log p_{\omega, \lambda, \psi}(Y_i|X_i) := \arg \max_{\omega, \lambda, \psi} \mathbb{E}_{\phi_i \sim q_{\psi}(\phi_i | T_s^i)} \left[ \sum_j -E_\omega(y''_{ij}, x_{ij}, \phi_i) + \mathbb{E}_{y'' \sim p_\omega(y''|x_{ij}, \phi_i)}[E_\omega(y''_{ij}, x_{ij}, \phi_i)] \right]$$

$$- \mathbb{E}_{\phi_i \sim q_{\psi}(\phi_i | T_s^i)}[-E_\lambda(\phi_i)] + \mathbb{E}_{\phi_i \sim p_\lambda(\phi_i')}[E_\lambda(\phi_i')] + \mathcal{H}(q_\psi(\phi_i | T_s^i)), \quad (24)$$

which only differs from the training objective of EBML-joint in Eqn. (8) in the sampling $y'' \sim p_\omega(y''|x_{ij}, \phi_i)$.

Results In Table 16 we observe that EBML-joint outperforms EBML-conditional by large margins in detecting OOD tasks (molecules with larger molecular size).

For an additional illustration, in Figure 12 we show the histogram of the averaged support samples negative log-likelihood (SNLL) of a CNPs model trained with $p(Y|X)$. We see that CNPs still outputs relatively high likelihood for some of the OOD tasks making their prediction indistinguishable from the ones on ID tasks

These empirical evidence support our motivation for modelling the joint distribution instead of the conditional distribution for potentially achieving better OOD task detection performance.

<table>
<thead>
<tr>
<th>OOD Scores</th>
<th>EBML-joint Mean $R^2$</th>
<th>AUROC</th>
<th>AUPR</th>
<th>FPR95</th>
<th>EBML-Conditional Mean $R^2$</th>
<th>AUROC</th>
<th>AUPR</th>
<th>FPR95</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNLL</td>
<td>0.533</td>
<td>99.71</td>
<td>99.71</td>
<td>2.20</td>
<td>54.91</td>
<td>54.14</td>
<td>66.60</td>
<td></td>
</tr>
<tr>
<td>Energy Sum</td>
<td>0.533</td>
<td>99.79</td>
<td>99.78</td>
<td>1.40</td>
<td>63.74</td>
<td>58.15</td>
<td>66.20</td>
<td></td>
</tr>
</tbody>
</table>

Figure 12: Histogram for SNLL of CNPs trained with $p(Y|X)$ of ID and OOD tasks, where OOD tasks contain molecules with much larger molecular sizes than the ones in ID tasks.
Algorithm 1 EBML Meta-training

Input: Meta-training tasks \{T_1, T_2, ..., T_N\}
Output: Meta-learned optimal parameters \(\omega^*, \psi^*, \lambda^*\)
1: Initialize the parameters \(\omega, \psi, \lambda\) for data EBM, latent posterior, and prior EBM
2: while not converged do
3: \(B \sim \{T_1, T_2, ..., T_N\}\) \text{ sample a batch of tasks}
4: for \(i = 1, 2, ..., |B|\) do \text{ for each sampled task in } B
5: \(T_s^i = \{X_s^i, Y_s^i\}_N^i\), \(T_q^i = \{X_q^i, Y_q^i\}_N^i\) \text{ sample support and query sets}
6: \(\phi_i \sim q(\phi_i | T_s^i)\) \text{ infer the task latent variable by the base algorithm}
7: \(x', y' \sim p_\omega(x', y' | \phi_i), \phi_i' \sim p_\lambda(\phi_i')\) \text{ Sampling by SGLD in Eqn. (3)}
8: Compute loss for \(T_i\) as \(L_i(\omega, \psi, \lambda)\) using Eqn. (8).
9: end for
10: \(\omega, \psi, \lambda \leftarrow \text{Opt}(\nabla_{\omega, \psi, \lambda} \frac{1}{|B|} \sum_{i \in B} L_i)\) \text{ Update parameters in the outer-loop}
11: end while

Algorithm 2 EBML Meta-testing

Input: Meta-testing tasks \{T_1, T_2, ..., T_N\}, parameters \(\lambda^*, \omega^*, \psi^*\), num adaptation steps \(K\)
Output: Query Prediction for each task \(\{Y_q^1, Y_q^2, ..., Y_q^N\}\)
1: for \(i = 1, 2, ..., N\) do \text{ for each test task}
2: \(T_i \rightarrow T_s^i = \{X_s^i, Y_s^i\}_N^i, T_q^i = \{X_q^i, Y_q^i\}_N^i\) \text{ get support and unlabelled query sets}
3: if \(K > 0\) then \text{ Using EBML OOD task adaptation}
4: \(\zeta \leftarrow \text{Alg. (3)}(\lambda^*, \omega^*, \psi^*, T_s^i, K)\) \text{ EBML OOD Task Adaptation}
5: else \(\zeta \leftarrow \emptyset\)
6: end if
7: \(y_j^q = \arg \min_y E_{\phi \sim q(\phi | T_s^i)} [E_{\omega^*}(x_j^q, y, \phi) + E_{\lambda^*}(\phi)], \forall j \in X_q^i\) \text{ query prediction}
8: end for

Algorithm 3 EBML OOD Task Adaptation

Input: Model parameters \(\lambda^*, \omega^*, \psi^*\), task support set \(T_s^i\), num adaptation steps \(K\)
Output: Task-specific parameter \(\zeta\)
1: for \(k = 1, 2, ..., K\) do \text{ for } K \text{ adaptation steps}
2: \(\phi_i \sim q(\phi_i | T_s^i)\) \text{ infer task latent variable by base algorithm}
3: Compute loss on \(T_s^i\) as \(L(\zeta)\) using Eqn. (11).
4: \(\zeta \leftarrow \text{Opt}(\nabla_\zeta L(\zeta))\) \text{ Update task-specific parameter } \zeta
5: end for