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

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

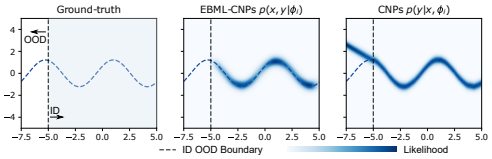
1 Introduction

Meta-learning [48, 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 [36, 44]. Analogous to the notorious domain shift [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 [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 [55], MRI-based pancreas segmentation given a host of meta-training tasks with pathology images [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.

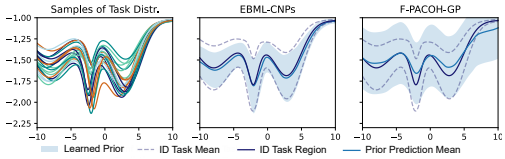
The *first* step to securing meta-generalization to a task is to develop awareness of whether the task is OOD or not, i.e., **OOD task detection**. Existing solutions in literature have pursued a variety of Bayesian meta-learning methods [7, 54, 41, 10, 43] that balance between flexibility and tractability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

*Correspondence to Ying Wei

37th Conference on Neural Information Processing Systems (NeurIPS 2023).



(a) **Incomplete OOD Coverage**. EBML-CNPs (middle, ours) successfully identifies OOD input with low likelihood, whereas CNPs (right) trained by $p(y|x, \phi_i)$ makes overconfidently wrong prediction.



(b) **Limited Expressiveness**. The meta-training task distribution learned by EBML-CNPs (middle, ours) outperforms F-PACOH-GP [43] whose prior distribution is built on GP.

Figure 1: Comparison of EBML and Bayesian meta-learning baselines for OOD detection.

solving the hierarchical probabilistic model $p(\mathbf{Y}_i|\mathbf{X}_i) = \int\int p(\mathbf{Y}_i|\mathbf{X}_i, \phi_i)p(\phi_i|\theta)p(\theta)\, d\phi_i d\theta$, where $\mathcal{T}_i = \{\mathbf{X}_i, \mathbf{Y}_i\}$ represents the i-th task. θ and ϕ_i denote parameters of the meta-model and task-specific model, respectively. Unfortunately, these methods present some limitations in their practical usage. (1) *Incomplete OOD coverage*: given that the Bayesian uncertainty is trained via maximizing the posterior $p(\mathbf{Y}_i|\mathbf{X}_i)$ above, it is not necessarily high when encountering an OOD task that shares the predictive function $p(\mathbf{Y}_i|\mathbf{X}_i)$ with some meta-training tasks but differs substantially in feature distributions $p(\mathbf{X}_i)$. We verify this in Figure 1a and Appendix D. (2) *Limited expressiveness*: for tractability purpose, the meta-learned prior $p(\phi_i|\theta)$ predicates on simple known distributions, e.g., Maximum A Posterior (MAP) estimation [6, 53] and Gaussian [41, 7, 43], which may struggle to align with the complex probabilistic structure of the meta-training task distribution (see Figure 1b). This misalignment inevitably leads to unreliable estimation of OOD tasks.

Upon detection of an OOD task, *secondly*, adaptation of the meta-learned prior promotes its generalization to this OOD task. We dub this strategy during meta-testing as **OOD task adaptation**, which is closely related to cross-domain meta-learning [4, 25, 34, 49]. The core philosophy behind cross-domain meta-learning is the introduction of task-specific parameters which are inferred via either gradient descent [28, 29] or feed-forward amortized encoder [42, 8] on the support set of each OOD task. Learning task-specific parameters, however, is *prone to overfitting* given the usually very limited size of a support set (e.g., 5 examples only in 5-way 1-shot classification).

The limitations are further complicated by the detachment of the existing solution to OOD task detection from that to OOD task adaptation. An explicit prior model is absent in existing Bayesian meta-learning methods for OOD task detection, so that adapting the prior during meta-testing to accommodate an OOD task is ambitious to achieve. On the other hand, cross-domain meta-learning approaches by design do not offer uncertainty estimation, thereby being a risky OOD task detector. Pursuing a coherent framework that supports both detection and adaptation of OOD tasks remains an open question, which motivates our proposal of a novel probabilistic meta-learning framework.

By virtue of the flexibility and expressiveness of energy-based models [24] in modelling complex data distributions, we propose the Energy-Based Meta-Learning (EBML) framework that overcomes the above-mentioned limitations. Specifically, we derive an energy-based model to explicitly model any meta-training task distribution, resulting in the composition of an explicit prior energy function and a complexity energy function. The sum of the two energy functions, trained directly to meet the joint distribution $p(\mathbf{X}_i, \mathbf{Y}_i)$ and parameterized with neural networks, has *completeness and expressiveness* advantages that give it an edge in detection of OOD tasks. During meta-testing, we iteratively update the parameter for a task that has been identified as OOD by gradient descent of energy minimization, which eventually adapts the prior towards in-distribution tasks and maximally leverages the meta-learned prior for *alleviating overfitting*.

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, 42], 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 hybird 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 $p(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^D$ as

$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)}, \tag{1}$$

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

$$\nabla_\theta \mathcal{L} = \mathbb{E}_{\mathbf{x}'\sim p_\theta}[\nabla_\theta E_\theta(\mathbf{x}')] - \mathbb{E}_{\mathbf{x}\sim p_D}[\nabla_\theta E_\theta(\mathbf{x})]. \tag{2}$$

Intuitively, Eqn. (2) encourages E_θ 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_θ. Computing Eqn. (2), thus, requires drawing samples from p_θ, which is challenging. Recent approaches [12, 3] on training EBMs resort to stochastic gradient Langevin dynamics (SGLD) [51] which generates samples following

$$\mathbf{x}^0 \sim p_0(\mathbf{x}), \qquad \mathbf{x}^{k+1} = \mathbf{x}^k - \frac{\eta^2}{2}\frac{\partial E_\theta(\mathbf{x}^k)}{\partial \mathbf{x}^k} + \eta \mathbf{z}^k. \tag{3}$$

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

$$\nabla_\theta \mathcal{L} = \mathbb{E}_{\mathbf{x}' \sim \text{stop\_grad}(q_\theta)}[\nabla_\theta E_\theta(\mathbf{x}')] - \mathbb{E}_{\mathbf{x} \sim p_D}[\nabla_\theta E_\theta(\mathbf{x})], \tag{4}$$

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 $\mathcal{T}_i$. We let $\mathbf{X}_i$, $\mathbf{Y}_i$ to denote sets of samples $\{\mathbf{x}_{ij}, y_{ij}\}$ in $\mathcal{T}_i$, and $\mathcal{T}_i^s$, $\mathcal{T}_i^q$ to denote support and query sets, respectively. The size of $\mathcal{T}_i$, $\mathcal{T}_i^s$, $\mathcal{T}_i^q$ is denoted by N_i, N_i^s, N_i^q, 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 $p(\mathbf{Y}|\mathbf{X})$ suffer from incomplete OOD coverage. To this end, we model the meta-training task distribution by (1) formulating the **joint distribution** $p(\mathbf{X}_i, \mathbf{Y}_i)$ of each task $\mathcal{T}_i$ and (2) maximizing the log-likelihood of all meta-training tasks. Concretely, by Kolmogorov's extension and de Finneti's theorems [22], we have the expected log-likelihood of the meta-training tasks as $\mathbb{E}_{\mathcal{P}_{ID}}[\log p(\mathcal{T}_i)] = \mathbb{E}_{\mathcal{P}_{ID}}[\log p(\mathbf{X}_i, \mathbf{Y}_i)] = \mathbb{E}_{\mathcal{P}_{ID}}[\log \int_{\boldsymbol{\phi}_i} \prod_{j=1}^{N_i} p(\mathbf{x}_{ij}, y_{ij}|\boldsymbol{\phi}_i) p(\boldsymbol{\phi}_i) d\boldsymbol{\phi}_i]$. Each $p(\mathcal{T}_i)$ is written in a factorized form over N_i conditional independent distributions with $\boldsymbol{\phi_i}$ being the task-specific latent variable. Due to the intractable integral over $\boldsymbol{\phi_i}$ in high dimension, we resort to amortized inference [8, 41] and learn with a lower-bound instead. This gives the ELBO

$$\mathbb{E}_{\mathcal{P}_{ID}}[\log p(\mathcal{T}_i)] \geq \mathbb{E}\left[\mathbb{E}_{\boldsymbol{\phi}_i \sim q_\psi(\boldsymbol{\phi}_i|\,\mathcal{T}_i^s)}\left[\log \prod_{j=1}^{N_i} p(\mathbf{x}_{ij}, y_{ij}|\boldsymbol{\phi}_i)\right] - \mathrm{KL}\big(q_\psi(\boldsymbol{\phi}_i|\,\mathcal{T}_i^s)||p(\boldsymbol{\phi}_i)\big)\right]. \tag{5}$$

Following the conventional wisdom [41, 28, 6], q_ψ is conditioned on the support set only during meta-training to align the inference procedure, i.e., $\boldsymbol{\phi}_i \sim q_\psi(\boldsymbol{\phi}_i|\,\mathcal{T}_i^s)$, for meta-training and meta-testing. It remains now to parameterize the three distributions in Eqn. (5) including **(a)** the task-specific data distribution $p(\mathbf{x}_{ij}, y_{ij}|\boldsymbol{\phi}_i)$, **(b)** the prior latent distribution $p(\boldsymbol{\phi}_i)$, and **(c)** the posterior latent distribution $q_\psi(\boldsymbol{\phi}_i|\,\mathcal{T}_i^s)$. Prior works parameterize these distributions in simple known forms, e.g., Gaussians [41, 7, 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 $p(\mathbf{x}_{ij}, y_{ij}|\boldsymbol{\phi}_i)$ and $p(\boldsymbol{\phi}_i)$. 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 $p(\mathbf{x}_{ij}, y_{ij}|\boldsymbol{\phi}_i)$ by an energy function parameterized with ω,

$$p(\mathbf{x}_{ij}, y_{ij}|\boldsymbol{\phi}_i) = p_\omega(\mathbf{x}_{ij}, y_{ij}|\boldsymbol{\phi}_i) = \frac{\exp(-E_\omega(\mathbf{x}_{ij}, y_{ij}, \boldsymbol{\phi}_i))}{Z(\omega, \boldsymbol{\phi}_i)}, \tag{6}$$

where E_ω denotes the task-specific data energy function conditioned on the latent $\boldsymbol{\phi}_i$, and $Z(\omega, \boldsymbol{\phi}_i)$ is the corresponding partition function. Note that the parameter ω of this EBM is shared by all tasks.

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

$$p(\boldsymbol{\phi}_i) = p_\lambda(\boldsymbol{\phi}_i) = \frac{\exp(-E_\lambda(\boldsymbol{\phi}_i))}{Z(\lambda)}, \forall i. \tag{7}$$

(c) Latent posterior As many meta-learning algorithms have already carefully designated the posterior latent distribution $q_\psi(\boldsymbol{\phi}_i|\,\mathcal{T}_i^s)$, we simply follow the same implementation of q_ψ in the chosen base meta-learning algorithm, e.g., MAP estimation in [8, 42, 1, 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.

$$\arg\max_{\omega,\psi,\lambda} \mathbb{E}_{\mathcal{T}_i\sim\mathcal{P}_{ID}} \Bigg[\mathbb{E}_{\boldsymbol{\phi}_i\sim q_\psi(\boldsymbol{\phi}_i|\mathcal{T}_i^s)}[\sum_{j=1}^{N_i} -E_\omega(\mathbf{x}_{ij}, y_{ij}, \boldsymbol{\phi}_i) + \mathbb{E}_{p_\omega(\mathbf{x}',y'|\phi_i)}[E_\omega(\mathbf{x}'_{ij}, y'_{ij}, \boldsymbol{\phi}_i)]] \\ - \mathbb{E}_{q_\psi(\boldsymbol{\phi}_i|\mathcal{T}_i^s)}[E_\lambda(\boldsymbol{\phi}_i)] + \mathbb{E}_{p_\lambda(\boldsymbol{\phi}'_i)}[E_\lambda(\boldsymbol{\phi}'_i)] + \mathcal{H}(q_\psi(\boldsymbol{\phi}_i|\,\mathcal{T}_i^s)) \Bigg]. \tag{8}$$

Solving the above meta-training objective involves sampling of $\mathbf{x}', y'$ from p_ω and $\boldsymbol{\phi}'_i$ from p_λ, in order to compute the expectations $\mathbb{E}_{p_\omega(\mathbf{x}',y'|\phi_i)}$ and $\mathbb{E}_{p_\lambda(\boldsymbol{\phi}'_i)}$ as Monte-Carlo averages. We follow the similar SGLD sampling procedure in Eqn. (3). Besides, since the majority of state-of-the-art meta-learning algorithms [8, 42, 1, 53] adopt the MAP estimation of the latent posterior q_ψ 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_ψ 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 $\log p(\mathbf{X}_i^s, \mathbf{Y}_i^s) = \log \mathbb{E}_{\boldsymbol{\phi}_i\sim p_\lambda(\boldsymbol{\phi}_i)}[p_\omega(\mathbf{X}_i^s, \mathbf{Y}_i^s|\boldsymbol{\phi}_i)]$. Despite the meta-learned latent prior EBM $p_\lambda(\boldsymbol{\phi}_i)$ 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 $p_\lambda(\boldsymbol{\phi}_i)$ 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

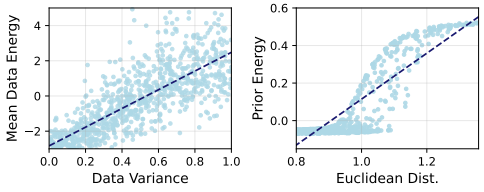


Figure 2: The roles of E_ω and E_λ in Energy Sum in detecting OOD tasks. Each dot denotes a task. **Left**: We perturb each support sample of a task $\mathcal{T}_i$ by $\boldsymbol{\eta}_{ij} \sim \mathcal{N}_i(0, \sigma_i)$ where we sample σ_i from $[0, 1]$ uniformly. The y-axis shows the average energy $\mathbb{E}_{\mathbf{x}_{ij}^s, \mathbf{y}_{ij}^s\sim\mathcal{T}_i^s, \boldsymbol{\eta}_{ij}\sim\mathcal{N}_i}[E_\omega(\mathbf{x}_i^s, \mathbf{y}_i^s, \boldsymbol{\phi}_i)]$ and the x-axis plots the variance σ_i^2. **Right**: We first compute the mean of the overall ID task latent prior as $\boldsymbol{\phi}_{ID} = \mathbb{E}_{\boldsymbol{\phi}_i\sim p_{ID}}[\boldsymbol{\phi}_i]$. The y-axis shows the energy $E_\lambda(\boldsymbol{\phi}_{ID} + \boldsymbol{\eta}_i)$ where $\boldsymbol{\eta}_i \sim \mathcal{N}(0, 1)$ for the i-th task and the x-axis plots the Euclidean distance of the perturbed latent from $\boldsymbol{\phi}_{ID}$.

$$\mathbb{E}_{q_\psi(\boldsymbol{\phi}_i|\,\mathcal{T}_i^s)} \big[\sum_j^{N_i^s} E_\omega(\mathbf{x}_{ij}^s, y_{ij}^s, \boldsymbol{\phi}_i) + E_\lambda(\boldsymbol{\phi}_i) \big]. \tag{9}$$

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_λ and (2) *how difficult it is to predict the observed support set conditioned on* $\boldsymbol{\phi}_i$ via the task-specific data energy score E_ω. First, the terms in the last line of Eqn. (8) for learning the latent prior EBM altogether correspond to maximizing the likelihood $\mathbb{E}_{\mathcal{T}_i\sim p(\mathcal{T})}\, \mathbb{E}_{q_\psi(\boldsymbol{\phi}_i|\mathcal{T}_i^s)}[\log\, p_\lambda(\boldsymbol{\phi}_i)]$, which enforces the latent prior energy score E_λ to capture the overall ID meta-training distribution. As illustrated in

Figure 2 (right), the further away a task is from the overall ID meta-training distribution measured in Euclidean distance, the larger the energy score E_λ is as expected. Second, conditioned on even the ID latent prior $\boldsymbol{\phi}_i$, those tasks with support samples as scattered as possible are especially difficult to predict. These tasks are considered to be OOD, as evidenced in higher values of E_ω in Figure 2 (left).

4.3 EBML for OOD Generalization

The Energy Sum proposed in Section 4.2 develops OOD awareness of a meta-testing task, based on which we differentiate our meta-testing procedures for effective meta-generalization.

Meta-testing for ID tasks Given the support set $\mathcal{T}^s$ of a meta-testing task, prediction of the label for its query $\mathbf{x}_j^q$ amounts to maximizing our approximated log-likelihood (see Eqn. (5)) of the task, i.e.,

$$y_j^q = \arg\min_y \mathbb{E}_{\boldsymbol{\phi}\sim q_\psi(\boldsymbol{\phi}|\,\mathcal{T}^s)}\left[E_\omega(\mathbf{x}_j^q, y, \boldsymbol{\phi}) + E_\lambda(\boldsymbol{\phi})\right]. \tag{10}$$

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 $E_\omega(\mathbf{x}_j^q, y, \boldsymbol{\phi})$, 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 $\boldsymbol{\phi} \sim q_\psi(\boldsymbol{\phi}|\,\mathcal{T}^s)$ 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 ζ which are optimized via the following,

$$\arg\min_\zeta \mathbb{E}_{\boldsymbol{\phi}\sim q_{\psi\cup\zeta}(\boldsymbol{\phi}|\,\mathcal{T}^s)}\Big[\sum_{j=1}^{N^s} E_\omega(\mathbf{x}_j^s, y_j^s, \boldsymbol{\phi}) + \max(E_\lambda(\boldsymbol{\phi}) - m, 0)\Big], \tag{11}$$

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., $m = \mathbb{E}_{p_{ID}}[\mathbb{E}_{\boldsymbol{\phi}_i\sim q_\psi(\boldsymbol{\phi}_i|\,\mathcal{T}_i^s)}[E_\lambda(\boldsymbol{\phi}_i)]]$.

As a result of optimizing the second term in Eqn. (11), the task-specific parameters ζ enable $q_{\psi\cup\zeta}(\boldsymbol{\phi}|\,\mathcal{T}^s)$ 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.

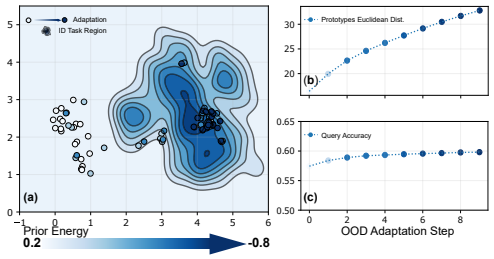


Figure 3: Illustration of the OOD task adaptation process on OOD domains of the meta-dataset [49] where each dot in (a) represents an OOD task in latent space $\boldsymbol{\phi}$. Minimizing Eqn. (11) leads to **(a)** the latent $\boldsymbol{\phi}$ of the OOD task moving to the ID region (contour plot), **(b)** the Euclidean distance between class prototypes enlarging, and consequently **(c)**, the classification accuracy on query samples increasing.

In Figure 3, we visualize the adaptation process when optimizing Eqn. (11) for OOD few-shot classification tasks in Meta-dataset [49]. As the prior energy of these OOD tasks decreases, their $\boldsymbol{\phi_i}$ gradually shift towards to 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 [1] 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.

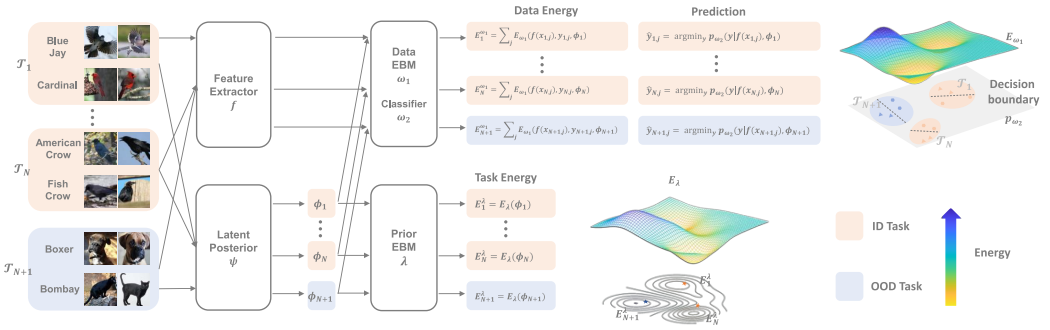


Figure 4: Overview of the EBML framework. The task latent variable ϕ_i is inferred from the support set $\mathcal{T}_i^s$ following the implementation of the base algorithm. The data and task energy scores are evaluated by the data and prior EBMs E_{ω_1} and E_λ, respectively; while the query labels are predicted by the classifier p_{ω_2} of the base algorithm.

Regression. Take CNPs [8] as an example base model. CNPs implements $q_\psi(\phi_i|\,\mathcal{T}_i^s)$ 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, $\mathcal{T}_i^s$. That said, we let the prior EBM to model the empirical distribution over such finite-dimension function embedding, i.e., $E_\lambda(\phi_i) : \mathbb{R}^D \to \mathbb{R}$.

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 ϕ_i as the set of class prototypes in each ID training task. Since ϕ_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 ω_2 with the loss function (e.g., cross entropy) in the base meta-learning model, which serves as a surrogate for $E_\omega(x_j^q, y, \phi)$ 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 experimetal 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 $\mathcal{X} \in [-5.0, 5.0]$. During testing, we evaluate the models on 500 ID and OOD tasks each with 512 equal-distant query points in $\mathcal{X}$. 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 C.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 *lbap-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 ($\mathbf{R^2}$) 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 **S**upport samples **N**egative **L**og-**L**ikelihood (**SNLL**) under model's task-specific predictive probability, i.e., $-\mathbb{E}_{\boldsymbol{\phi}_i \sim q_\psi(\boldsymbol{\phi}_i|\mathcal{T}_i^s)}[\mathbb{E}_j[\log p_\omega(y_{ij}^s|x_{ij}^s, \boldsymbol{\phi}_i)]]$ for baselines and $\mathbb{E}_{\boldsymbol{\phi}_i \sim q_\psi(\boldsymbol{\phi}_i|\mathcal{T}_i^s)}[\mathbb{E}_j[E_\omega(y_{ij}^s, x_{ij}^s, \boldsymbol{\phi}_i)]]$ for EBML. Following common practice [17, 16], we report **AUROC**, **AUPR** and **FPR95** for OOD detection performance. Details for these metrics can be found in Appendix B.1.

5.3 OOD Detection Results

Energy sum performs best in OOD task detection. Table 1 and 8. 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.

OOD Scores	Models	Sinusoids			DrugOOD		
		AUROC↑	AUPR↑	FPR95↓	AUROC↑	AUPR↑	FPR95↓
Std	ABML [41]	50.14	54.80	97.20	57.82	50.31	74.80
	F-PACOH-GP [43]	49.52	51.30	94.20	81.74	71.99	32.00
	CNPs [8]	22.72	35.34	99.60	93.56	89.58	13.00
	Metafun [53]	76.57	80.33	82.40	85.68	80.55	58.18
SNLL	ABML [41]	82.48	81.31	61.00	80.99	79.12	47.60
	F-PACOH-GP [43]	91.78	93.23	52.40	37.73	45.01	85.21
	CNPs [8]	95.63	96.46	34.22	17.25	34.07	91.40
	Metafun [53]	96.25	97.11	32.00	83.54	85.54	65.17
	EBML-CNPs (Ours)	96.46	97.41	29.40	99.71	99.71	2.20
Energy Sum	EBML-CNPs (Ours)	**97.74**	**98.31**	**14.20**	**99.79**	**99.78**	**1.40**

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 $p(y_{ij}|\mathbf{x}_{ij}, \boldsymbol{\phi}_i)$ 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.

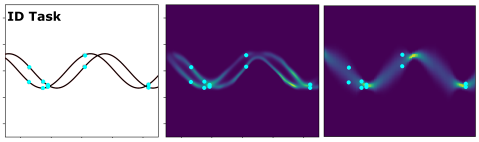


Figure 5: Predictive distribution of **Middle** an data EBM vs **Right** a Gaussian for an ID task.

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-CNPs eventually 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-CNPs 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.

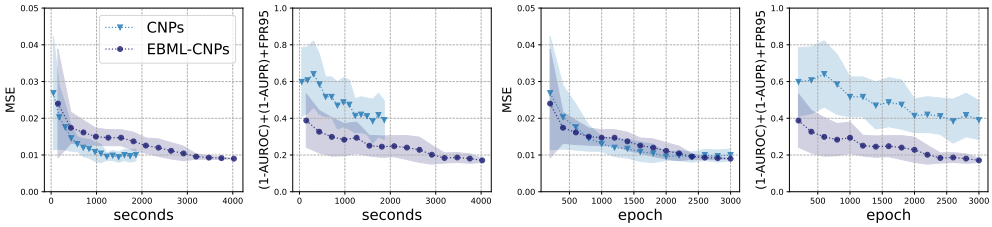


Figure 6: **Left** : Wall-clock convergence in seconds, and **Right**: performance vs number of training epochs, for EBML-CNPs vs CNPs in single-sinosoid few-shot regression tasks. The plots show the regression (MSE ↓) and combined OOD tasks detection (1-AUROC)+(1-AUPR)+FPR95 ↓ performance on single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

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 expressive of EBM over simple distributions can indeed lead to learning a more accurate model of the meta-training ID task distribution.

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,

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

OOD Scores	Simple-CNAPs [1]			TSA [28]		
	AUROC↑	AUPR↑	FPR95↓	AUROC↑	AUPR↑	FPR95↓
max-softmax [16]	85.50	85.54	65.43	89.25	87.14	46.02
max-logits [15]	50.00	70.83	95.00	50.14	44.64	95.28
ODIN [30]	90.49	89.42	43.57	92.02	90.18	37.36
MAH [27]	71.18	69.76	90.52	94.54	93.95	23.83
Domain Classifier	83.10	73.18	53.17	n/a	n/a	n/a
EBML Energy Sum	**97.01**	**94.92**	**6.74**	**99.10**	**98.48**	**3.64**

Table 3: Ablation study on Energy Sum for OOD detection on single-sinusoids.

Models	OOD Scores	Sinusoids		
		AUROC↑	AUPR↑	FPR95↓
ABML [41]	SNLL	82.48	81.31	61.00
	+Gauss Prior	86.95	86.64	52.20
CNPs [8]	SNLL	94.81	96.34	38.40
	+Gauss Prior	94.61	96.10	34.40
EBML-CNPs	SNLL	96.46	97.41	29.40
	+EBM Prior	**97.74**	**98.31**	**14.20**

Table 4: Few-shot regression performance on single-sinusoids and DrugOOD [21] tasks.

Models	Sinusoids	DrugOOD			
	ID MSE ↓	ID Mean R^2 ↑	ID Median R^2 ↑	OOD Mean R^2 ↑	OOD Median R^2 ↑
F-PACOH-GP [43]	$0.068_{\pm 0.016}$	0.492	0.454	0.055	0.027
Metafun[53]	**0.009**$_{\pm 0.002}$	0.537	0.541	0.054	0.027
CNPs [8]	**0.009**$_{\pm 0.002}$	**0.540**	0.549	0.066	**0.046**
ABML [41]	$0.127_{\pm 0.013}$	0.452	0.443	0.051	0.029
MAML [6]	$0.119_{\pm 0.013}$	0.462	0.475	0.055	0.024
EBML-CNPs	**0.009**$_{\pm 0.002}$	0.533	**0.553**	**0.071**	0.043

and EBML-conditional, which follows the same training with EBML-joint but models $p(\mathbf{Y} \mid \mathbf{X})$ instead of $p(\mathbf{X}, \mathbf{Y})$. With all other factors being the same, EBML-joint significantly outperform EBML-conditional in OOD detection on DrugOOD regression tasks with domain shift in $\mathbf{X}$. This supports our motivation for using the joint distribution instead of the conditional distribution for training a potentially better OOD detector. Detail of this ablation study can be found in Appendix D.

5.4 OOD Generalization Results

EBML achieves SOTA regression performance. In Table 4, for single-sinusoids, EBML is able to match the MSE of the best-performing baseline methods; while on multi-sinusoids in Table 7, EBML obtains the lowest ID NLL, specifically 0.58 lower than the best baseline, thanks to our energy-based decoder which is sufficiently expressive for modelling the multi-modality at each input.

Task adaption using Eqn. (11) **improves few-shot classification performance.** In Table 5, we report the average classification accuracy computed over 600 test tasks per ID and OOD domains. In meta-testing, we obtain classification results for EBML-TSA by running gradient descent on the objective in Eqn. (11) to optimize the task-specific modules in TSA from scratch. With this addition of prior energy in the OOD adaption objective, EBML-TSA further improves TSA results in 5/7 ID domains and all 5 OOD domains. Additional OOD classification results in Table 11 Appendix C further confirm the superiority of our proposed OOD task adaptation strategy in Eqn. (11) over prior baselines.

Table 5: Classification performance on 5-way 1-shot tasks for both ID and OOD domains in Meta-dataset.

Datasets	TSA [28]	EBML-TSA (Ours)
Omniglot	$98.63_{\pm 0.26}$	**98.67**$_{\pm 0.26}$
Textures	$51.93_{\pm 0.87}$	**52.35**$_{\pm 0.88}$
Aircraft	**78.91**$_{\pm 0.86}$	$78.47_{\pm 0.86}$
Birds	$75.02_{\pm 0.90}$	**75.52**$_{\pm 0.90}$
VGG Flower	**80.37**$_{\pm 0.80}$	$80.30_{\pm 0.83}$
Fungi	$70.89_{\pm 0.93}$	**72.29**$_{\pm 0.94}$
Quickdraw	$79.02_{\pm 0.84}$	**80.27**$_{\pm 0.85}$
MSCOCO	$52.28_{\pm 0.94}$	**53.03**$_{\pm 0.97}$
Traffic Sign	$57.40_{\pm 0.94}$	**58.85**$_{\pm 1.01}$
CIFAR10	$49.16_{\pm 0.82}$	**50.04**$_{\pm 0.89}$
CIFAR100	$62.25_{\pm 1.01}$	**62.77**$_{\pm 1.05}$
MNIST	$74.72_{\pm 0.83}$	**76.08**$_{\pm 0.88}$
Avg ID	76.40	**76.84**
Avg OOD	59.16	**60.15**
Avg All	69.22	**69.89**

6 Conclusion and Limitation

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 adaptation strategy does not consider the effect of negative transfer, as some OOD tasks may benefit from adaptating 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

[1] Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[2] Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved Contrastive Divergence Training of Energy-Based Models. In *Proceedings of the 38th International Conference on Machine Learning*, pages 2837–2848. PMLR, July 2021. ISSN: 2640-3498.

[3] Yilun Du and Igor Mordatch. Implicit Generation and Modeling with Energy Based Models. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[4] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Selecting relevant features from a multi-domain representation for few-shot classification. In *European Conference on Computer Vision*, 2020.

[5] Sven Elflein, Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. On Out-of-distribution Detection with Energy-based Models. page 13.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135. PMLR, July 2017. ISSN: 2640-3498.

[7] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9537–9548, Red Hook, NY, USA, 2018. Curran Associates Inc.

[8] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional neural processes. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1704–1713. PMLR, 10–15 Jul 2018.

[9] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural Processes. Technical Report arXiv:1807.01622, arXiv, July 2018. arXiv:1807.01622 [cs, stat] type: article.

[10] Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard Turner. Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations*, 2019.

[11] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *International Conference on Learning Representations*, 2018.

[12] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.

[13] Will Sussman Grathwohl, Jacob Jin Kelly, Milad Hashemi, Mohammad Norouzi, Kevin Swersky, and David Duvenaud. No {mcmc} for me: Amortized sampling for fast and stable training of energy-based models. In *International Conference on Learning Representations*, 2021.

[14] Tian Han, Erik Nijkamp, Linqi Zhou, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. Joint Training of Variational Auto-Encoder and Latent Energy-Based Model. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7975–7984, Seattle, WA, USA, June 2020. IEEE.

[15] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Xiaodong Song. Scaling out-of-distribution detection for real-world settings. In *International Conference on Machine Learning*, 2022.

[16] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*, 2017.

[17] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

[18] Shion Honda, Shoi Shi, and Hiroki R. Ueda. Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery. 2019.

[19] Ekaterina Iakovleva, Jakob Verbeek, and Karteek Alahari. Meta-learning with shared amortized variational inference. In *ICML*, pages 4572–4582. PMLR, 2020.

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[21] Yuanfeng Ji, Lu Zhang, Jiaxiang Wu, Bingzhe Wu, Long-Kai Huang, Tingyang Xu, Yu Rong, Lanqing Li, Jie Ren, Ding Xue, Houtim Lai, Shaoyong Xu, Jing Feng, Wei Liu, Ping Luo, Shuigeng Zhou, Junzhou Huang, Peilin Zhao, and Yatao Bian. DrugOOD: Out-of-Distribution (OOD) Dataset Curator and Benchmark for AI-aided Drug Discovery – A Focus on Affinity Prediction Problems with Noise Annotations. *arXiv:2201.09637 [cs, q-bio]*, January 2022. arXiv: 2201.09637.

[22] Achim Klenke. *Probability theory: a comprehensive course*. Springer Science Business Media, 2013.

[23] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[24] Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A Tutorial on Energy-Based Learning. page 59, 2006.

[25] Hae Beom Lee, Hayeon Lee, Donghyun Na, Saehoon Kim, Minseop Park, Eunho Yang, and Sung Ju Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. In *International Conference on Learning Representations*, 2020.

[26] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3744–3753, 2019.

[27] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[28] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Cross-domain few-shot learning with task-specific adapters. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7151–7160, 2021.

[29] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Universal representation learning from multiple domains for few-shot classification. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9506–9515, 2021.

[30] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

[31] Lu Liu, William Hamilton, Guodong Long, Jing Jiang, and H. Larochelle. A universal representation transformer layer for few-shot image classification. *ArXiv*, abs/2006.11702, 2020.

[32] Qiang Liu and Dilin Wang. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[33] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based Out-of-distribution Detection. In *Advances in Neural Information Processing Systems*, volume 33, pages 21464–21475. Curran Associates, Inc., 2020.

[34] Yanbin Liu, Juho Lee, Linchao Zhu, Ling Chen, Humphrey Shi, and Yi Yang. A Multi-Mode Modulator for Multi-Domain Few-Shot Classification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8433–8442, Montreal, QC, Canada, October 2021. IEEE.

[35] Shuai Luo, Yujie Li, Pengxiang Gao, Yichuan Wang, and Seiichi Serikawa. Meta-seg: A survey of meta-learning for image segmentation. *Pattern Recognition*, page 108586, 2022.

[36] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, 2022.

[37] Yifei Ming, Ying Fan, and Yixuan Li. Poem: Out-of-distribution detection with posterior sampling. In *International Conference on Machine Learning*. PMLR, 2022.

[38] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning Latent Space Energy-Based Prior Model. In *Advances in Neural Information Processing Systems*, volume 33, pages 21994–22008. Curran Associates, Inc., 2020.

[39] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning Latent Space Energy-Based Prior Model. In *Advances in Neural Information Processing Systems*, volume 33, pages 21994–22008. Curran Associates, Inc., 2020.

[40] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

[41] Sachin Ravi and Alex Beatson. Amortized bayesian meta-learning. In *International Conference on Learning Representations*, 2019.

[42] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7957–7968. Curran Associates, Inc., 2019.

[43] Jonas Rothfuss, Dominique Heyn, Jinfan Chen, and Andreas Krause. Meta-learning reliable priors in the function space. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[44] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022.

[45] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems*, 34:23359–23373, 2021.

[46] Amrith Setlur, Oscar Li, and Virginia Smith. Two sides of meta-learning evaluation: In vs. out of distribution. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[47] Zhuo Sun, Jijie Wu, Xiaoxu Li, Wenming Yang, and Jing-Hao Xue. Amortized bayesian prototype meta-learning: A new probabilistic meta-learning approach to few-shot image classification. In *International Conference on Artificial Intelligence and Statistics*, pages 1414–1422. PMLR, 2021.

[48] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[49] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2020.

[50] Haotao Wang, Aston Zhang, Yi Zhu, Shuai Zheng, Mu Li, Alex J Smola, and Zhangyang Wang. Partial and asymmetric contrastive learning for out-of-distribution detection in long-tailed recognition. In *International Conference on Machine Learning*, pages 23446–23458, 2022.

[51] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress.

[52] Jeffrey Ryan Willette, Hae Beom Lee, Juho Lee, and Sung Ju Hwang. Meta learning low rank covariance factors for energy based deterministic uncertainty. In *International Conference on Learning Representations*, 2022.

[53] Jin Xu, Jean-Francois Ton, Hyunjik Kim, Adam R Kosiorek, and Yee Whye Teh. Metafun: Meta-learning with iterative functional updates. In *International Conference on Machine Learning (ICML)*, 2020.

[54] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian Model-Agnostic Meta-Learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[55] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, pages 1094–1100, 2020.

[56] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep Sets. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[57] Yang Zhao, Jianwen Xie, and Ping Li. Learning Energy-Based Generative Models via Coarse-to-Fine Expanding and Sampling. February 2022.

Contents

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 $\mathcal{T}_i$ w.r.t. the parameters of the EBMs and the task latent posterior distribution, i.e., λ, ω and ψ,

$$\arg\max_{\lambda,\psi,\omega} \log\, p(\mathcal{T}_i) \xrightarrow{\text{Eqn. (5)}} \arg\max_{\lambda,\psi,\omega} \underbrace{\mathbb{E}_{\phi_i\sim q_\psi(\phi_i|\,\mathcal{T}_i^s)}\,[\log p_\omega(\mathbf{X}_i,\mathbf{Y}_i|\phi_i)]}_{(I)} - \mathrm{KL}(q_\psi(\phi_i|\,\mathcal{T}_i^s)||p_\lambda(\phi_i)) \tag{12}$$

where q_ψ, p_ω and p_λ denote the model distributions of the latent posterior, the data EBM and the prior EBM, respectively. Recall p_λ and p_ω are EBMs where $p_\omega(\mathbf{X}_i,\mathbf{Y}_i\mid\phi_i)=\prod_j \frac{\exp\,(-E_\omega(\mathbf{x}_{ij},y_{ij},\phi_i))}{Z(\omega,\phi_i)}$ and $p_\lambda=\frac{\exp\,(-E_\lambda(\phi_i))}{Z(\lambda)}$. We rewrite the KL term as

$$-\mathrm{KL}(q_\psi(\phi_i|\,\mathcal{T}_i^s)||p_\lambda(\phi_i)) = -\,\mathbb{E}_{q_\psi(\phi_i|\,\mathcal{T}_i^s)}\,\big[\log\frac{q_\psi(\phi_i|\,\mathcal{T}_i^s)}{p_\lambda(\phi_i)}\big] \tag{13}$$

$$= -\,\mathbb{E}_{q_\psi(\phi_i|\,\mathcal{T}_i^s)}\,\big[\log q_\psi(\phi_i|\,\mathcal{T}_i^s\,\big] - \mathbb{E}_{q_\psi(\phi_i|\,\mathcal{T}_i^s)}\,\big[\frac{1}{\log\,p_\lambda(\phi_i)}\big] \tag{14}$$

$$= \underbrace{\mathcal{H}(q_\psi(\phi_i|\,\mathcal{T}_i^s)}_{(III)} + \underbrace{\mathbb{E}_{q_\psi(\phi_i|\,\mathcal{T}_i^s)}[\log\,p_\lambda(\phi_i)]}_{(II)} \tag{15}$$

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

$$\mathbb{E}_{\phi_i\sim q_\psi(\phi_i|\,\mathcal{T}_i^s)}\,[\log p_\omega(\mathbf{X}_i,\mathbf{Y}_i|\phi_i)] \xrightarrow{\text{Eqn. (4)}} \mathbb{E}_{\boldsymbol{\phi}_i\sim q_\psi(\boldsymbol{\phi}_i|\mathcal{T}_i^s)}\Bigg[\sum_j^{N_i} -E_\omega(\mathbf{x}_{ij},y_{ij},\boldsymbol{\phi}_i) + \mathbb{E}_{p_\omega(\mathbf{x}',y'|\phi_i)}[E_\omega(\mathbf{x}'_{ij},y'_{ij},\boldsymbol{\phi}_i)]\Bigg] \tag{16}$$

$$\mathbb{E}_{q_\psi(\phi_i|\,\mathcal{T}_i^s)}[\log\,p_\lambda(\phi_i)] \xrightarrow{\text{Eqn. (4)}} \mathbb{E}_{q_\psi(\boldsymbol{\phi}_i|\mathcal{T}_i^s)}\Big[-E_\lambda(\boldsymbol{\phi}_i) + \mathbb{E}_{p_\lambda(\boldsymbol{\phi}'_i)}[E_\lambda(\boldsymbol{\phi}'_i)\Big]. \tag{17}$$

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

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

The log-likelihood of a task $\mathcal{T}_i$ writes

$$\log\,p(\mathcal{T}_i) = \log\int\prod_{j=1}^{N_i} p(\mathbf{x}_{ij},y_{ij}|\phi_i)p_\lambda(\phi_i)\,d\phi_i. \tag{18}$$

$$\geq\ \mathbb{E}_{\phi_i\sim q_\psi(\phi_i|\,\mathcal{T}_i^s)}\,[\log p_\omega(\mathbf{X}_i,\mathbf{Y}_i|\phi_i)] - \mathrm{KL}(q_\psi(\phi_i|\,\mathcal{T}_i^s)||p_\lambda(\phi_i)) \tag{19}$$

$$= \mathbb{E}_{q_\psi(\phi_i|\,\mathcal{T}_i^s)}\,[\underbrace{\log p_\omega(\mathbf{X}_i,\mathbf{Y}_i|\phi_i)}_{\propto\ -E_\omega(\mathbf{X}_i,\mathbf{Y}_i,\phi_i)}] + \mathbb{E}_{q_\psi(\phi_i|\,\mathcal{T}_i^s)}[\underbrace{\log\,p_\lambda(\phi_i)}_{\propto\ -E_\lambda(\phi_i)}] + \mathcal{H}(q_\psi(\phi_i|\,\mathcal{T}_i^s)\big], \tag{20}$$

which is lower-bounded by the ELBO in Eqn. (20) characterized by the learned q_ψ, p_ω and p_λ. The ELBO is proportional to the sum of two energy functions and the entropy of the posterior distribution q_ψ, 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 SimpleCNAPS [1]) adopt the MAP estimation of the posterior q_ψ which is determinisitc, the entropy essential becomes zero, and the expectations $\mathbb{E}_{q_\psi(\phi_i|\mathcal{T}_i^s)}$ in the first and second terms in Eqn. (20) simplify to energy function evaluation at $\mathbf{X}_i$, $\mathbf{Y}_i$ and ϕ_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 $\mathbf{X}_i^s, \mathbf{Y}_i^s$ for inferring a small set of unknown task-specific parameters ζ. This is akin to maximizing the log-likelihood of the support set w.r.t. the task-specific parameters ζ which defines the posterior latent distribution $q_{\psi\cup\zeta}(\phi_i \mid \mathcal{T}_i^s)$ under the regularization of a prior. First, the tractable ELBO for the prior predictive likelihood is

$$\log\ p_{\lambda,\psi,\omega}(\mathcal{T}_i^s) = \log \int \prod_{j=1}^{N_i^s} p_\omega(\mathbf{x}_{ij}, y_{ij}|\phi_i) p_\lambda(\phi_i)\, d\phi_i. \tag{21}$$

$$\geq\ \mathbb{E}_{q_\psi(\phi_i|\,\mathcal{T}_i^s)}\,[\sum_{j=1}^{N_i^s} \log p_\omega(\mathbf{x}_{ij}, y_{ij}|\phi_i)] - \mathrm{KL}(q_\psi(\phi_i|\,\mathcal{T}_i^s)||p_\lambda(\phi_i)). \tag{22}$$

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

$$\arg\min_{\zeta} \mathbb{E}_{q_{\psi\cup\zeta}(\phi_i|\,\mathcal{T}_i^s)}\,[-\sum_{j=1}^{N_i^s} \log p_\omega(\mathbf{x}_{ij}, y_{ij}|\phi_i)] + \mathrm{KL}(q_{\psi\cup\zeta}(\phi_i|\,\mathcal{T}_i^s)||p_\lambda(\phi_i)). \tag{23}$$

Assuming a maximum a posterior (MAP) estimate of the task-specific latent distribution which approximates $q_{\psi\cup\zeta}(\phi_i|\,\mathcal{T}_i^s)$ by a Dirac-delta function $q_\psi(\phi_i|\,\mathcal{T}_i^s) = \delta(\phi_i - \hat{\phi}_i \mid \mathcal{T}_i^s)$. As a result, the second KL term reduces to a likelihood evaluation, i.e., $\mathrm{KL}(q_\psi(\phi_i|\,\mathcal{T}_i^s)||p_\lambda(\phi_i)) = \mathbb{E}_{q_\psi(\phi_i|\,\mathcal{T}_i^s)}[\log \frac{q_\psi(\phi_i|\,\mathcal{T}_i^s)}{p_\lambda(\phi_i)}] = -\log\ p_\lambda(\phi_i) = E_\lambda(\phi_i) + \log\ Z(\lambda)$. Since the (log-)partition function $\log\ Z(\lambda)$ is a constant w.r.t. the $\arg\min$ parameter ζ, we then have $\arg\min_\zeta \mathrm{KL}(q_\psi(\phi_i|\,\mathcal{T}_i^s)||p_\lambda(\phi_i)) = \arg\min_\zeta E_\lambda(\phi_i)$. From here, we see that minimizing the task prior energy approximates the minimization of the KL-divergence between the task-specific posterior $q_{\psi\cup\zeta}$ and the meta-learned ID prior p_λ 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., $\arg\min_\zeta \max(E_\lambda(\phi_i) - m, 0)$, can yield better empirical performance.

While for the first log-likelihood term inside the expectation, $-\sum_{j=1}^{N_i^s} \log p_\omega(\mathbf{x}_{ij}, y_{ij}|\phi_i)$ which is equivalent to the sum of data energy scores $\sum_{j=1}^{N_i^s}[E_\omega(\mathbf{x}_{ij}, y_{ij}, \phi_i) + \log\ Z(\omega, \phi_i)]$, we use the decoder ω_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 [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 τ 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 [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 correcty 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 $E_\omega(\mathbf{x}, \mathbf{y}, \phi_i)$ and the prior EBM are MLPs. Follow the encoder implementation in CNPs [8], $q_\psi(\phi_i | \mathcal{T}_i^s)$ composes of a within-task mean pooling operation sandwiched between two arbitrary learnable transformations parameterized by MLP, i.e., $q_\psi(\phi_i | \mathcal{T}_i^s) = \text{MLP}_{\psi_1}(\frac{1}{N_i^s}\sum_{j=1}^{N_i^s} \text{MLP}_{\psi_2}([\mathbf{x}_{ij}^s, y_{ij}^s]))$. The output of q_ψ 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 [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: 0.01; outer learning rate: 0.001;

ABML [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 [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 η: 0.01; Data SGLD η: 0.001; num-SGLD-iter: 20; energy L2 penalty 1.0;

B.3 Drug Activity Prediction

Preprocessing Molecular Graph Inputs Each input $\mathbf{x}$ 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 [18] for converting the sring input $\mathbf{x}$ into vector representation $\tilde{\mathbf{x}}$ in $\mathbb{R}^{1024}$. We treat this $\tilde{\mathbf{x}}$ 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 [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 covariance: True.

EBML-CNPs Prior SGLD η: 0.1; Data SGLD η: 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 [1] 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 ϕ_i to be a set of class prototypes used in the cosine classifier of each ID meta-training task, our EBM prior $p_\lambda(\phi_i)$ resembles a distribution over task-specific predictive functions from the ID domains. The architecture for the Prior EBM is depicted in Figure 7.

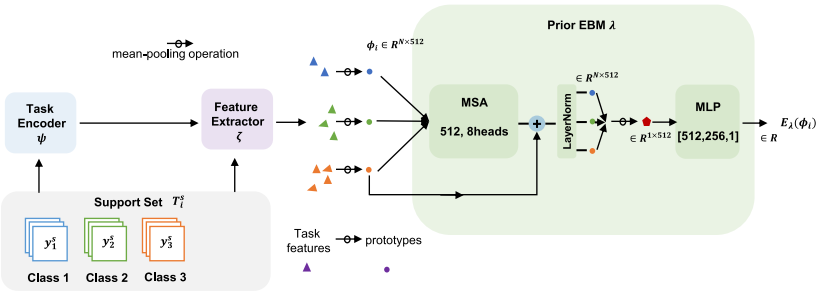


Figure 7: Model architecture for the prior EBM for classification.

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 [1] 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].

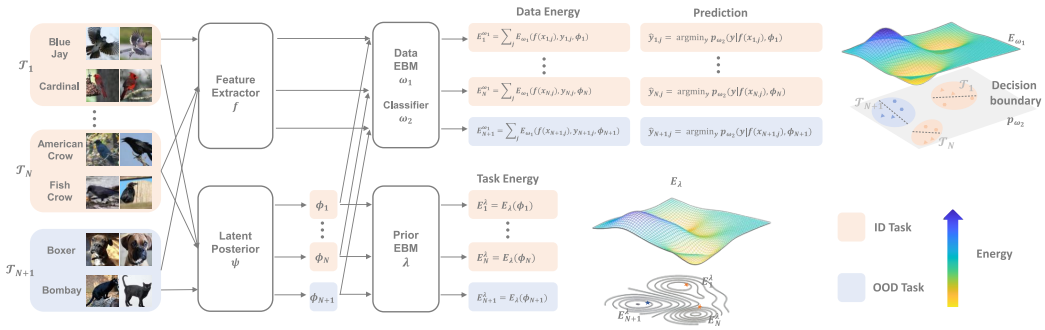


Figure 8: Overview of the EBML framework for image classification tasks. The task latent variable ϕ_i are inferred from the support set $\mathcal{T}_i^s$ following the implementation of the base algorithm. The data and task energy scores are evaluated by the data and prior EBMs E_{ω_1} and E_λ, respectively; while the query labels are predicted by the classifier p_{ω_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.

Groups	Hyperparameters	Values
Training	num SGLD steps	40
	Data SGLD η	1.0
	Prior SGLD η	10.0
	Energy L2 penalty	1.0
	EBM Spectral Norm	True
Adaptation	num steps	10
	TSA learning rate β [28]	0.00091
	TSA learning rate α [28]	0.000267
	weight on prior energy	0.1
	m in prior energy margin loss	0.4
	optimizer	Adam

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 $\mathcal{T}_i$. as the average instance-level OOD scores over its input images $\{\mathbf{x}\}^{N_i}$. More specifically:

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

ODIN [30], extends the softmax-score by introducing temperature scaling and input preprocessing. More concretely, the input is perturbed following $\hat{\mathbf{x}} = \mathbf{x} + \epsilon\, sign(\nabla_{\mathbf{x}} \log S_{\hat{y}}(\mathbf{x}; T))$, which moves $\mathbf{x}$ in the direction that increases the temperature-scaled softmax score $S_{\hat{y}}(\mathbf{x}; T))$, computed as $\max_c \frac{\exp(logit_{[c]}(\mathbf{x})/T)}{\sum_{c=0}^{N-1}\exp(logit_{[c]}(\mathbf{x})/T)}$. The final ODIN confidence score writes $S_{\hat{y}}(\hat{\mathbf{x}}; T))$.

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 $\mathbf{x}$ 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 $\mathbf{x}$.

The perturbation magnitude ϵ and the temperature scale T used in ODIN and MAH are determined using the validation set of meta-dataset, with a 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, 8 we see that EBML-CNPs achieved both the best regression and OOD detection performance. In Table 9, our proposed energy outperform all ablated models.

Table 7: Regression performance for multi-sinusoids experiments.

Models	ID NLL↓
ABML [41]	0.886±0.048
F-PACOH-GP [43]	1.289±0.023
CNPs [8]	0.865±0.069
Metafun[53]	0.874±0.051
EBML-CNPs	**0.282** ±0.041

Table 8: OOD detection performance on multi-sinusoids tasks.

OOD Scores	Models	AUROC↑	AUPR↑	FPR95↓
Std	ABML [41]	74.14	72.15	73.67
	Metafun [53]	76.52	77.34	88.12
SNLL	ABML [41]	54.75	56.32	99.60
	F-PACOH-GP [43]	55.24	68.95	100.00
	CNPs [8]	70.43,	79.71	92.4
	Metafun [53]	79.21	77.03	90.98
	EBML-CNPs (Ours)	92.77	94.25	46.20
Energy Sum	EBML-CNPs (Ours)	**94.91**	**96.15**	**34.60**

Table 9: Ablation study for Energy Sum on Multi-sinusoids few-shot regression tasks.

OOD Scores	Models	AUROC↑	AUPR↑	FPR95↓
ABML [41]	SNLL	54.75	56.32	99.60
	+Gauss Prior	86.64	86.45	50.00
CNPs [8]	SNLL	70.19	79.49	95.20
	+Gauss Prior	82.90	87.23	76.60
EBML-CNPs	SNLL	92.77	94.25	46.20
	+EBM Prior	**94.91**	**96.15**	**34.60**

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 [2]. 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

Table 10: Classification performance on Meta-dataset 5-way 1-shot tasks, with *shuffle_buffer_size*=0. * indicates results reported by [28].

Datasets	TSA* [28]	EBML-TSA (Ours)
Omniglot	**96.3**±0.4	**96.3**±0.5
Textures	**54.5**±0.9	**54.5**±0.8
Aircraft	**79.6**±0.9	79.0±0.9
Birds	74.5±0.9	**75.3**±0.9
VGG Flower	**80.3**±0.8	80.2±0.8
Fungi	75.3±1.0	**77.1**±0.9
Quickdraw	79.3±0.9	**79.9**±0.9
MSCOCO	59.9±1.0	**60.2**±1.0
Traffic Sign	57.2±1.0	**58.2**±0.9
CIFAR10	55.8±0.9	**56.8**±0.9
CIFAR100	63.7±1.0	**64.6**±1.0
MNIST	80.1±0.9	**82.0**±0.9
Avg ID	77.1	**77.5**
Avg OOD	63.4	**64.4**
Avg All	71.4	**72.0**

[2] 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].

Datasets	**SimpleCNAPs** [1]	**EBML-SimpleCNAPs** (Ours)	**URL** [29]	**EBML-URL** (Ours)
MSCOCO	$49.37_{\pm 0.99}$	$51.75_{\pm 0.96}$	$59.14_{\pm 0.95}$	$59.82_{\pm 0.97}$
Traffic Sign	$55.63_{\pm 0.96}$	$56.12_{\pm 0.97}$	$57.62_{\pm 0.90}$	$58.26_{\pm 0.90}$
CIFAR10	$50.79_{\pm 0.85}$	$51.16_{\pm 0.89}$	$54.37_{\pm 0.83}$	$54.39_{\pm 0.86}$
CIFAR100	$54.15_{\pm 0.95}$	$55.23_{\pm 0.93}$	$63.03_{\pm 0.97}$	$62.74_{\pm 0.96}$
MNIST	$80.25_{\pm 0.85}$	$81.01_{\pm 0.85}$	$78.85_{\pm 0.86}$	$79.78_{\pm 0.87}$

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

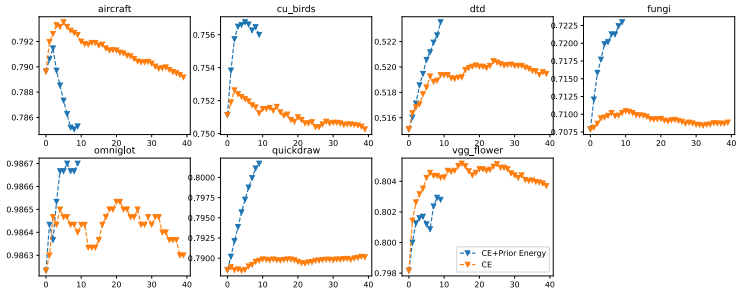
We instantiate EBML with two additional baseline Meta-learning algorithms including SimpleCNAPs [1] and URL [29] and report their 5-way-1-shot classification accuracy of OOD tasks in Meta-datasets [49] in Table 11 above. OOD-adaptation for EBML is performed by optimizing Eqn. (11) w.r.t. to the task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

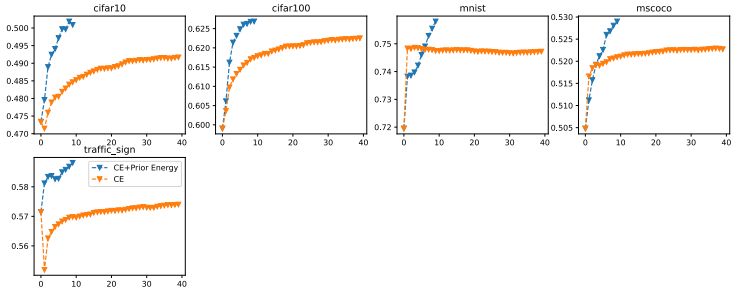
Figure 9: TSA vs EBML-TSA query set classification accuracy during adaptation on ID datasets



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:

Figure 10: TSA vs EBML-TSA query set classification accuracy during adaptation on OOD datasets



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.

Datasets	**EBML-TSA**	**TSA+EM**	**TSA+PL**
Omniglot	98.67±0.26	98.81±0.26	98.25±0.25
Textures	52.35±0.88	52.12±0.87	51.91±0.87
Aircraft	78.47±0.86	79.40±0.86	79.09±0.86
Birds	75.52±0.90	74.76±0.89	75.07±0.90
VGG Flower	80.30±0.83	80.00±0.81	80.71±0.80
Fungi	72.29±0.94	70.95±0.93	70.44±0.94
Quickdraw	80.27±0.85	79.18±0.85	78.96±0.85
MSCOCO	53.03±0.97	52.24±0.94	52.55±0.94
Traffic Sign	58.85±1.01	57.13±0.95	57.06±0.94
CIFAR10	50.04±0.89	50.22±0.81	49.43±0.83
CIFAR100	62.77±1.05	62.47±1.00	62.70±0.99
MNIST	76.08±0.88	75.23±0.88	75.14±0.85
Avg ID	76.84	76.86	76.35
Avg OOD	60.15	59.46	59.38
Avg All	**69.89**	68.16	69.28

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.

	Varying-Way Varying-Shot		**5-Way 1-Shot**	
Datasets	**SimpleCNAPs** [1]	**TSA** [28]	**Simple-CNAPs** [1]	**TSA** [28]
Omniglot	91.61±0.57	94.77±0.41	97.87±0.27	98.63±0.26
Textures	65.47±0.71	77.06±0.67	44.11±0.83	51.93±0.87
Aircraft	81.28±0.69	88.56±0.51	65.08±0.89	78.91±0.86
Birds	73.80±0.81	80.86±0.77	66.21±0.98	75.02±0.90
VGG Flower	90.15±0.50	92.48±0.52	76.57±0.83	80.37±0.80
Fungi	44.82±1.13	66.49±1.02	50.95±0.95	70.89±0.93
Quickdraw	73.30±0.81	82.33±0.58	67.61±0.95	79.02±0.84
MSCOCO	35.19±0.94	55.22±1.09	38.78±0.79	52.28±0.94
Traffic Sign	42.59±1.01	82.60±0.97	50.84±0.90	57.40±0.94
CIFAR10	56.65±0.80	80.40±0.71	37.59±0.67	49.16±0.82
CIFAR100	44.13±1.10	70.38±0.97	46.11±0.89	62.25±1.01
MNIST	93.89±0.37	96.44±0.43	76.52±0.83	74.72±0.83
Avg ID	**74.35**	**83.22**	66.91	76.40
Avg OOD	**54.49**	**77.01**	49.97	59.16
Avg All	**66.07**	**80.63**	59.85	69.22

C.3 EBML with Probabilistic Posterior Distribution $q_\psi(\phi_i|\,\mathcal{T}_i^s)$

Since EBML is a flexible plug-in, it is definitely compatible with those methods that use probabilistic posterior q_ψ. 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 EMBL has applied to resort to a MAP estimate for q_ψ.

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

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

OOD Scores	**EBML-NPs**				**EBML-CNPs**			
	MSE	**AUROC**	**AUPR**	**FPR95**	**MSE**	**AUROC**	**AUPR**	**FPR95**
SNLL		95.94	97.16	31.20		96.46	97.41	29.40
Energy Sum w/o Entropy	0.009±0.002	97.10	97.83	20.40	0.009±0.002	97.74	98.31	14.20
Energy Sum w Entropy		97.58	97.79	14.80		n/a	n/a	n/a

In Table 14, we conclude that EBML-NPs with probabilistic q_ψ achieves comparable performance to its deterministic version, EBML-CNPs; there is no significant improvement in performance by switching to a probabilistic q_ψ. 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.

Sinusoids	CNPs [8]	EBML-CNPs	f-PACOH-GP [43]	ABML [41]
Training time / 500 tasks	0.67	1.83	3.13	11.95*

Meta-dataset	SimpleCNAPs [1]	EBML-SimpleCNAPs
Training time / 1000 tasks	1.02	1.75

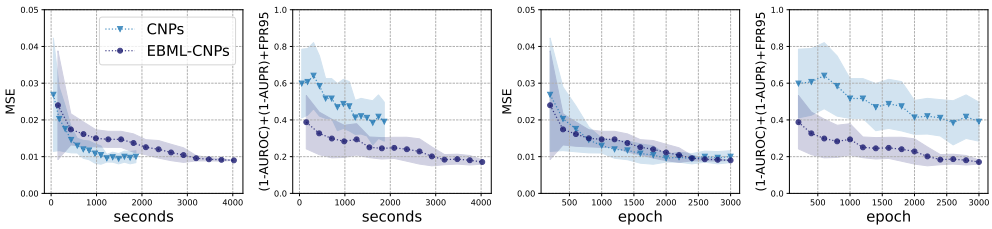


Figure 11: **Left** : Wall-clock convergence in seconds, and **Right**: performance vs number of training epochs, for EBML-CNPs vs CNPs in single-sinosoid few-shot regression tasks. The plots show the regression (MSE $\downarrow$) and combined OOD tasks detection (1-AUROC)+(1-AUPR)+FPR95 $\downarrow$ 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-word applications, distribution shift in input space, i.e. the distribution shift in $\mathbf{X}$, 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 [21]. Such input distribution shift, unfortunately, cannot always be correctly reflected by models trained to maximize the predictive probability $p_\omega(y|\mathbf{x}, \phi_i)$.

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 $p(\mathbf{Y}|\mathbf{X})$ instead of $p(\mathbf{X}, \mathbf{Y})$ of the meta-training task distribution. Concretely, the training objective for EBML-conditional becomes

$$
\begin{aligned}
&\arg\max_{\omega,\lambda,\psi} \log p_{\omega,\lambda,\psi}(Y_i|X_i) := \\
&\text{argmax}_{\omega,\lambda,\psi}\ \mathbb{E}_{\phi_i\sim q_\psi(\phi_i|\,\mathcal{T}_i^s)}\left[\sum_j -E_\omega(y_{ij}^s, x_{ij}^s, \phi_i) + \mathbb{E}_{y'\sim p_\omega(y'|x_{ij}^s,\phi_i)}[E_\omega(y'_{ij}, x_{ij}^s, \phi_i)]\right] \\
&- \mathbb{E}_{\phi_i\sim q_\psi(\phi_i|\,\mathcal{T}_i^s)}[-E_\lambda(\phi_i)] + \mathbb{E}_{\phi_i\sim p_\lambda(\phi'_i)}[E_\lambda(\phi'_i)] + \mathcal{H}(q_\psi(\phi_i|\,\mathcal{T}_i^s)),
\end{aligned} \tag{24}
$$

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

Table 16: EBML-joint vs EBML-conditional on DrugOOD few-shot regression and OOD task detection.

OOD Scores	EBML-joint				EBML-Conditional			
	Mean R^2	AUROC	AUPR	FPR95	Mean R^2	AUROC	AUPR	FPR95
SNLL	0.533	99.71	99.71	2.20	0.534	54.91	54.14	66.60
Energy Sum		99.79	99.78	1.40		63.74	58.15	66.20

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 $p(\mathbf{Y}|\mathbf{X})$. 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.

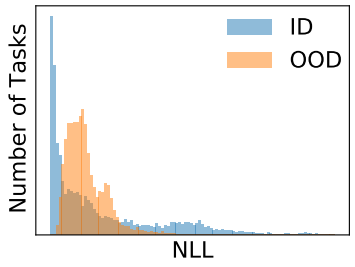


Figure 12: Histogram for SNLL of CNPs trained with $p(\mathbf{Y}|\mathbf{X})$ of ID and OOD tasks, where OOD tasks contain molecules with much larger molecular sizes than the ones in ID tasks.

E Pseudocode for EBML

Algorithm 1 EBML Meta-training

Input: Meta-training tasks $\{\mathcal{T}_1, \mathcal{T}_2, ... \mathcal{T}_N\}$
Output: Meta-learned optimal parameters $\omega*, \psi*, \lambda*$

Initialize the parameters ω,ψ,λ for data EBM, latent posterior, and prior EBM
while not converged **do**
  $B \sim \{\mathcal{T}_1, \mathcal{T}_2, ... \mathcal{T}_N\}$ ▷ sample a batch of tasks
  for $i = 1, 2, ..., |B|$ **do** ▷ for each sampled task in $\mathcal{B}$
    $\mathcal{T}_i^s = \{\mathbf{X}_i^s, \mathbf{Y}_i^s\}^{N_i^s}, \mathcal{T}_i^q = \{\mathbf{X}_i^q, \mathbf{Y}_i^q\}^{N_i^q} \sim \mathcal{T}_i$ ▷ sample support and query sets
    $\phi_i \sim q_\psi(\phi_i | \mathcal{T}_i^s)$ ▷ infer the task latent variable by the base algorithm
    $x', y' \sim p_\omega(x', y'|\phi_i)$, $\phi_i' \sim p_\lambda(\phi_i')$ ▷ Sampling by SGLD in Eqn. (3)
    Compute loss for $\mathcal{T}_i$ as $\mathcal{L}_i(\omega, \psi, \lambda)$ using Eqn. (8).
  end for
  $\omega, \psi, \lambda \leftarrow \text{Opt}(\nabla_{\omega,\psi,\lambda} \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathcal{L}_i)$ ▷ Update parameters in the outer-loop
end while

Algorithm 2 EBML Meta-testing

Input: Meta-testing tasks $\{\mathcal{T}_1, \mathcal{T}_2, ... \mathcal{T}_N\}$, parameters $\lambda^*, \omega^*, \psi^*$, num adaptation steps K
Output: Query Prediction for each task $\{\mathbf{Y}_1^q, \mathbf{Y}_2^q, ... \mathbf{Y}_N^q\}$

for $i = 1, 2, ..., N$ **do** ▷ for each test task
  $\mathcal{T}_i \rightarrow \mathcal{T}_i^s = \{\mathbf{X}_i^s, \mathbf{Y}_i^s\}^{N_i^s}, \mathcal{T}_i^q = \{\mathbf{X}_i^q, \}^{N_i^q}$ ▷ get support and unlabelled query sets
  if $K > 0$ **then** ▷ Using EBML OOD task adaptation
    $\zeta \leftarrow$ Alg. (3)$(\lambda*, \omega*, \psi*, \mathcal{T}_i^s, K)$ ▷ EBML OOD Task Adaptation
  else
    $\zeta \leftarrow \emptyset$
  end if
  $y_j^q = \arg\min_y \mathbb{E}_{\boldsymbol{\phi} \sim q_{\psi* \cup \zeta}(\boldsymbol{\phi} | \mathcal{T}_i^s)} \left[E_{\omega*}(\mathbf{x}_j^q, y, \boldsymbol{\phi}) + E_{\lambda*}(\boldsymbol{\phi}) \right], \forall j \in \mathbf{X}_i^q$ ▷ query prediction
end for

Algorithm 3 EBML OOD Task Adaptation

Input: Model parameters $\lambda^*, \omega^*, \psi^*$, task support set $\mathcal{T}_i^s$, num adaptation steps K
Output: Task-specific parameter ζ

for $k = 1, 2, ...K$ **do** ▷ for K adaptation steps
  $\phi_i \sim q_{\psi* \cup \zeta}(\phi_i | \mathcal{T}_i^s)$ ▷ infer task latent variable by base algorithm
  Compute loss on $\mathcal{T}_i^s$ as $\mathcal{L}(\zeta)$ using Eqn. (11).
  $\zeta \leftarrow \text{Opt}(\nabla_\zeta \mathcal{L}(\zeta))$ ▷ Update task-specific parameter ζ
end for