Maxitive Donsker-Varadhan Formulation for Possibilistic Variational Inference

Jasraj Singh[*,†]
Danmarks Tekniske Universitet

Shelvia Wongso[†]
Nanyang Technological University

Jeremie Houssineau
Nanyang Technological University

Badr-Eddine Chérief-Abdellatif
Sorbonne Université

Abstract

Variational inference (VI) is a cornerstone of modern Bayesian learning, enabling approximate inference in complex models. However, its formulation depends on expectations and divergences defined through high-dimensional integrals, often rendering analytical treatment impossible and necessitating heavy reliance on approximations. Possibility theory, an imprecise probability framework, allows us to directly model epistemic uncertainty instead of relying on a subjective interpretation of probabilities. While this framework provides robustness and interpretability under sparse or imprecise information, adapting VI to the possibilistic setting requires rethinking core concepts such as divergences, which presuppose additivity. In this work, we develop a principled formulation for performing possibilistic VI by establishing a maxitive analogue of the classical Donsker-Varadhan formulation. The resulting framework enables us to derive a learning rule for possibilistic VI with exponential-family candidates and practical update rules for neural-network training, giving rise to a family of optimizers termed `CBOpt`. Finally, we demonstrate that `CBOpt` achieves competitive performance on both in-domain and out-of-domain image classification tasks.

1 Introduction

Variational inference (VI) has become a cornerstone of modern Bayesian learning, enabling approximate inference in complex models that were previously intractable. For many years, purely Bayesian methods were hampered by computational cost, but advances in Monte Carlo and especially variational approximations have brought them within reach (Blei et al., 2017; Salimans et al., 2015). More recently, variational learning methods have been shown to perform well even at large scale and with little additional computational overhead (Shen et al., 2024). This motivates further development of variational learning methods, given their ability to naturally provide uncertainty estimates. In VI, one typically posits a simple (parametric) variational family $\mathcal{Q}$ and then finds the probability distribution $q \in \mathcal{Q}$ that best approximates the true posterior $q^\star_{\mathrm{add}}$ by maximizing the evidence lower bound (ELBO). This framework underlies many tools today, such as variational autoencoders (Kingma and Welling, 2013) and Bayesian neural networks (Blundell et al., 2015; Kingma et al., 2015), and blurs the line between tractable and intractable Bayesian analyses.

Introduced by Zadeh (1978), possibility theory, an imprecise probability framework, offers a complementary approach for modelling uncertainty. Unlike probability, possibility does not require additivity: events have a degree of plausibility (≤ 1) and a dual *necessity*, with logical rules based on min/max operators. This allows handling epistemic uncertainty naturally. For example, the analogue

[*] Work done while at Nanyang Technological University.
[†] Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

of probability distributions, referred to as *possibility functions*, can easily be made fully uninformative (Hieu et al., 2025), a difficult endeavour in probability theory. The flexibility and scalability of possibility theory make it suitable for challenging applications such as control problems under uncertainty (Xue et al., 2025) and point process-based inference (Houssineau, 2021).

A key challenge arises when we try to adapt variational methods to this possibilistic setting. Standard VI relies on notions like Shannon entropy, which do not directly carry over: for example, the ELBO includes an entropy term, but there is no obvious "possibilistic entropy" analogous to the Shannon entropy of a probability distribution (Shannon and Weaver, 1949). Similarly, divergences like Kullback–Leibler (KL) require additive measures (Kullback and Leibler, 1951). As a result, traditional VI objectives cannot be used out-of-the-box. Nonetheless, recent work has sought approximations in the possibilistic context. For instance, Cella and Martin (2025) developed a variational-like approximation for inferential models (Martin and Liu, 2013), and used a Monte Carlo-based strategy to search over the chosen family of possibility functions. In that spirit, our goal is to formulate a VI framework for general possibilistic models: we define an objective that aligns a tractable candidate possibility function with the (target) posterior possibility function, and which can be optimized in lieu of maximizing a probabilistic ELBO.

Contributions. In this work, we present a novel maxitive analogue of the classical Donsker–Varadhan (DV) formulation (Theorem 2), which admits two dual variational characterizations whose optima are possibility functions that respectively lower- and upper-bound the possibilistic posterior. The associated objective, the consistency bound (CBO), is built on a maxitive pseudo-divergence that plays a role analogous to the KL divergence in standard VI. Building on this formulation, we develop a practical framework for possibilistic VI within a special class of exponential-family possibility functions, highlighting their structural parallels with the probabilistic case and giving rise to special mathematical structures. From this framework, we derive a learning rule that leads to a new family of optimizers, `CBOpt`[3]. We empirically validate these methods on both in-domain and out-of-domain image classification tasks, demonstrating competitive predictive and OOD-detection performance. These developments provide a foundation for extending variational reasoning to possibility theory, offering new avenues for inference under epistemic uncertainty. Components of this framework, specifically the maxitive pseudo-divergence and the maxitive posterior under the uninformative prior, are also used by Ni et al. (2026) to derive a second-order predictor for image classification; the underlying theory, including the maxitive DV duality, the dual CBO formulations, and the exponential-family treatment, is developed in full here for the first time.

Outline. The remainder of the paper is organized as follows. Section 2 reviews the necessary background on possibility theory, and Section 3 recalls the classical Donsker–Varadhan formulation and its connection to Bayesian variational inference. Section 4 establishes our maxitive DV analogue and the dual CBO formulations. Section 5 develops the exponential-family treatment and derives the learning rule underlying `CBOpt`. Section 6 reports our empirical evaluation, and Section 7 concludes with a discussion of limitations and future directions. Additional related work and notation are provided in Appendix A and Appendix B, respectively.

2 Primer on Possibility Theory

Possibility theory provides a dedicated representation of epistemic uncertainty where, like probabilities, each event is assigned a degree of possibility between 0 and 1 and, unlike probabilities, the main operation on possibilities is optimization. In possibility theory, random variables are replaced by *uncertain variables* and operations like conditioning and marginalization are defined similarly but with a maximum or supremum rather than a sum or integral, as stated more formally in Appendix C.

The expected value for a possibility function f is $\mathbb{E}_f^\star[\boldsymbol{\theta}] \doteq \operatorname{argmax}_{\theta\in\Theta} f(\theta)$, which is a set in general, and satisfies $\mathbb{E}_f^\star[T(\boldsymbol{\theta})] = T(\mathbb{E}_f^\star[\boldsymbol{\theta}])$ for any mapping T for which f attains its supremum on each nonempty fibre, a property shared with the maximum likelihood estimate. When $\mathbb{E}_f^\star[\boldsymbol{\theta}]$ is a singleton $\{\theta^\star\}$, we do not make a distinction between this singleton and the element $\theta^\star$. If f is twice differentiable at $\theta^\star$, we can define the precision as $\mathcal{I}_f(\boldsymbol{\theta}) \doteq \mathbb{E}_f^\star[-\nabla^2 \log f(\boldsymbol{\theta})] = -\nabla^2 \log f(\theta)|_{\theta=\theta^\star}$, where the operator ∇^2 is the Hessian and *precision* is to be understood as an inverse covariance

[3]Code available at github.com/shelvia-w/cbopt.

matrix. The *normal possibility function* with expected value μ and covariance matrix $\Sigma \succ 0$ is defined as $f(\theta) = \overline{\mathrm{N}}(\theta; \mu, \Sigma) \doteq \exp(-\frac{1}{2}(\theta-\mu)^\top \Sigma^{-1}(\theta-\mu))$. It verifies $\mathbb{E}^\star_f[\boldsymbol{\theta}] = \mu$ and $\mathcal{I}_f(\boldsymbol{\theta}) = \Sigma^{-1}$, as is leveraged in the Gaussian (Laplace) approximation.

We define the set $\mathcal{F}(\Theta) \doteq \{f : \Theta \to [0,1] \ : \ \sup_{\theta\in\Theta} f(\theta) = 1\}$ of possibility functions over Θ. As opposed to the set $\mathcal{P}(\Theta)$ of probability distributions, $\mathcal{F}(\Theta)$ is a partially ordered set when equipped with the partial order $\preceq$ defined as $f \preceq g \iff f(\theta) \le g(\theta), \forall \theta \in \Theta$. There is a greatest element in $\mathcal{F}(\Theta)$, namely the function equal to 1 everywhere, which we denote by $\mathbf{1}$. For any subset $\mathcal{G}$ of $\mathcal{F}(\Theta)$, $\max \mathcal{G}$ and $\min \mathcal{G}$ denote its greatest and least elements under the pointwise order, when they exist. Table 2 summarizes other key notations used in this paper.

3 Bayesian Inference and the Donsker-Varadhan Variational Formula

We first provide a review of the probabilistic variational formulation of Bayesian inference and its variational approximations. We consider the problem of learning about an unknown parameter θ_0 in a set Θ, which can be thought of either as the true parameter in a statistical procedure or as the solution of an optimization problem. In this context, it is usual to alternate between i) the optimization viewpoint where the main objects are a loss ℓ and a regularizer R on Θ, and ii) the inference viewpoint where the main objects are a likelihood $L \propto \exp(-\ell)$, that is, $L(\theta) \propto \exp(-\ell(\theta))$ for all $\theta \in \Theta$, and a prior π with density $\propto \exp(-R)$ with respect to Lebesgue's measure. This leads to the (generalized) Bayesian posterior $q^\star_{\mathrm{add}}$ for the regularized loss $\hat{\ell} \doteq \ell + R$ with respect to π as

$$q^\star_{\mathrm{add}}(\mathrm{d}\theta) \ \doteq \ \frac{\exp(-\ell(\theta))\,\pi(\mathrm{d}\theta)}{\int \exp(-\ell(\theta'))\pi(\mathrm{d}\theta')} = \frac{\exp\left(-\hat{\ell}(\theta)\right)\mathrm{d}\theta}{\int \exp\left(-\hat{\ell}(\theta')\right)\mathrm{d}\theta'}. \tag{1}$$

We start with the following classical variational formula, known since at least the 1950s (see e.g. Kullback (1959, Exercise 8.28) for the finite case), generally attributed in the general setting to Donsker and Varadhan (1976), and later rediscovered in statistics by Zellner (1988). We refer the interested reader to Catoni (2004, Page 160) for a proof.

Theorem 1 (Donsker and Varadhan's variational formula)**.** *Let $(\Theta, \mathcal{T})$ be a measurable space and let ν be a probability measure on Θ. For any measurable function $h : \Theta \to \mathbb{R}$ bounded above, we have*

$$\log \int e^h \mathrm{d}\nu = \sup_{\rho\in\mathcal{P}(\Theta)} \left\{ \int h \mathrm{d}\rho - \mathrm{KL}\left(\rho \| \nu\right) \right\}, \tag{2}$$

where $\mathcal{P}(\Theta)$ denotes the set of probability measures on Θ and $\mathrm{KL}(\rho\|\nu)$ is the Kullback–Leibler divergence of ρ with respect to ν, with the convention $\infty - \infty = -\infty$. Moreover, the supremum on the right-hand side is achieved exactly at the Gibbs measure ν_h whose density with respect to ν is

$$\frac{\mathrm{d}\nu_h}{\mathrm{d}\nu}(\theta) \doteq \frac{e^{h(\theta)}}{\int e^h \mathrm{d}\nu}. \tag{3}$$

(Generalized) Bayesian inference via Donsker-Varadhan. Theorem 1 provides an information-theoretic characterization of the Bayesian posterior. For a loss ℓ bounded below, setting $h = -\ell$ and $\nu = \pi$ (the prior), we obtain that the normalizing constant $Z_{\mathrm{add}} = \int \exp(-\ell)\,\mathrm{d}\pi$ (also called the *evidence* in the statistical setting) satisfies

$$\log Z_{\mathrm{add}} = \sup_{q\in\mathcal{P}(\Theta)} \mathrm{ELBO}(q)\,, \qquad \text{with} \qquad \mathrm{ELBO}(q) \doteq -\mathbb{E}_{\theta\sim q}[\ell(\theta)] - \mathrm{KL}(q\|\pi). \tag{4}$$

The ELBO (*Evidence Lower Bound*) provides a lower bound on the log-evidence; the exact posterior can then be expressed as the solution of the infinite-dimensional optimization problem

$$q^\star_{\mathrm{add}} = \operatorname*{argmin}_{q\in\mathcal{P}(\Theta)} \left\{ \mathbb{E}_{\theta\sim q}[\ell(\theta)] + \mathrm{KL}(q\|\pi) \right\}. \tag{5}$$

Thus, Donsker and Varadhan's lemma simultaneously identifies the Bayesian posterior as the optimizer of a regularized expected loss (the negative ELBO) and expresses the log-evidence as the supremum of the ELBO over all probability measures.

(Generalized) Variational inference via Donsker-Varadhan. In practice, computing $q^\star_{\text{add}}$ exactly is generally intractable. *Generalized Variational Inference* (GVI) consists in restricting the optimization to a tractable family $\mathcal{Q} \subset \mathcal{P}(\Theta)$, turning the infinite-dimensional problem (5) into a simpler, typically parametric, optimization problem.

This restriction naturally induces a gap between the true log-evidence and the ELBO achievable within $\mathcal{Q}$, which can be quantified via $\mathrm{ELBO}(q) = \log Z_{\text{add}} - \mathrm{KL}(q \,\|\, q^\star_{\text{add}})$, for all $q \in \mathcal{Q}$. Maximizing the ELBO within $\mathcal{Q}$ therefore corresponds exactly to minimizing the KL divergence to the exact posterior restricted to the chosen family. In this sense, GVI can be interpreted both as the best tractable approximation (in the reverse KL sense) of the exact posterior within $\mathcal{Q}$, and as the optimal information-processing rule within $\mathcal{Q}$.

4 Possibilistic Inference and a Maxitive Donsker-Varadhan Analogue

In this section, we establish a maxitive analogue of the Donsker–Varadhan variational formula and use it to derive a possibilistic inference framework. We first observe that many standard regularizers naturally attain their minimum at 0, so that the prior $\pi = \exp(-R)$ is directly a possibility function whereas π could be improper as a probability distribution if $\exp(-R)$ is not integrable. This leads to the following formula for the (generalized) posterior $g^\star_{\text{max}}$ in possibility theory:

$$g^\star_{\text{max}}(\theta) \doteq \frac{\exp(-\ell(\theta))\pi(\theta)}{\sup_{\theta'} \exp(-\ell(\theta'))\pi(\theta')} = \frac{\exp(-\hat{\ell}(\theta))}{\sup_{\theta'} \exp(-\hat{\ell}(\theta'))}. \tag{6}$$

The function $\pi(\theta)$ now denotes a prior possibility function over Θ, typically $\pi(\theta) = \exp(-R(\theta))$. The normalizing constant in the maxitive posterior is defined as

$$Z_{\text{max}} \doteq \sup_{\theta' \in \Theta} \exp(-\ell(\theta'))\pi(\theta'), \tag{7}$$

and is referred to as the maxitive marginal likelihood. Contrary to its probabilistic analogue Z_{add}, which measures the overall fit of the model to the data, the quantity Z_{max} quantifies the *consistency* between the prior and the likelihood when the latter is normalised to have supremum one. This notion is central in robust inference and can be used, for instance, to detect outliers (Houssineau and Nott, 2022).

Theorem 2 (Maxitive Donsker-Varadhan formula)**.** *Let $\pi : \Theta \to [0,1]$ be a possibility function taking positive values. For any function $\ell : \Theta \to \mathbb{R}$ such that $0 < Z_{\text{max}} < \infty$, we have, with $\log 0 = -\infty$,*

$$\log \sup_{\theta \in \Theta} e^{-\ell(\theta)}\pi(\theta) = \sup_{g \in \mathcal{F}(\Theta)} \inf_{\theta \in \Theta} \left\{ -\ell(\theta) - \log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\}, \tag{8a}$$

$$= \inf_{g \in \mathcal{F}(\Theta)} \sup_{\theta \in \Theta} \left\{ -\ell(\theta) - \log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\}. \tag{8b}$$

Moreover, the supremum in (8a) is achieved by any possibility function lower-bounding the posterior $g^\star_{\text{max}}$, that is

$$\operatorname*{argmax}_{g \in \mathcal{F}(\Theta)} \inf_{\theta \in \Theta} \left\{ -\ell(\theta) - \log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\} = \left\{ g \in \mathcal{F}(\Theta) : g \preceq g^\star_{\text{max}} \right\}. \tag{9}$$

Similarly, the infimum in (8b) is achieved by any possibility function upper-bounding $g^\star_{\text{max}}$:

$$\operatorname*{argmin}_{g \in \mathcal{F}(\Theta)} \sup_{\theta \in \Theta} \left\{ -\ell(\theta) - \log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\} = \left\{ g \in \mathcal{F}(\Theta) : g^\star_{\text{max}} \preceq g \right\}. \tag{10}$$

The proof is provided in Appendix D.1. Note that if $\pi(\theta) = 0$ for some values of θ, then (8a) and (8b) still hold when just considering suprema/infima over the support Θ_π of π instead of Θ, and over the set $\mathcal{F}_\pi(\Theta)$ of possibility functions whose support is included in Θ_π instead of $\mathcal{F}(\Theta)$.

On the left-hand side of (8a) appears the log-consistency $\log Z_{\text{max}}$, while the quantity optimized on the right-hand side defines a *consistency bound* (CBO). Two dual versions can be introduced:

$$\underline{\mathrm{CBO}}(g) \doteq \inf_{\theta} \left\{ -\ell(\theta) - \log \frac{g(\theta)}{\pi(\theta)} \right\}, \qquad \overline{\mathrm{CBO}}(g) \doteq \sup_{\theta} \left\{ -\ell(\theta) - \log \frac{g(\theta)}{\pi(\theta)} \right\}. \tag{11}$$

These provide lower and upper bounds on the log-consistency, as shown in Theorem 2. For any $g \in \mathcal{F}(\Theta)$, we even have the decomposition

$$\underline{\mathrm{CBO}}(g) = \log Z_{\max} - D_{\max}(g \,\|\, g^\star_{\max}), \tag{12a}$$

$$\overline{\mathrm{CBO}}(g) = \log Z_{\max} + D_{\max}(g^\star_{\max} \,\|\, g), \tag{12b}$$

where the *max-relative entropy* between two possibility functions is defined as

$$D_{\max}(g \,\|\, f) \doteq \sup_{\theta \in \Theta} \log \frac{g(\theta)}{f(\theta)} \geq 0, \tag{13}$$

with the convention that the supremum is taken over all θ such that $f(\theta) > 0$, and that $D_{\max}(g \,\|\, f) = +\infty$ whenever $\mathrm{supp}(g) \nsubseteq \mathrm{supp}(f)$. The maxitive pseudo-divergence $D_{\max}$ is also used by Ni et al. (2026) to derive a second-order predictor for image classification.

Maximizing (resp. minimizing) the lower (resp. upper) CBO with respect to g is equivalent to minimizing the max-relative entropy to the Gibbs possibility function $g^\star_{\max}$. Any maximizer g in (8a) satisfies $g \preceq g^\star_{\max}$, whereas any minimizer g in (8b) satisfies $g \succeq g^\star_{\max}$. In particular, the standard posterior $g^\star_{\max}$ is always an optimizer, and pointwise bounds all other solutions (below or above, depending on the formulation). We visualize the behaviour of max-relative entropy in Appendix E.2.

Canonical choice of maximizer: Among all g that optimize the CBO, a natural canonical representative is the standard possibilistic posterior $g^\star_{\max}$. Under the pointwise order, $g^\star_{\max}$ is the greatest optimizer of the lower CBO and the least optimizer of the upper CBO, and is therefore the least/most informative representative, respectively. Choosing $g^\star_{\max}$ is natural because it is the sup/inf-envelope of all maximizers/minimizers and i) *for the lower CBO*, does not introduce any extra, arbitrary information, ii) *for the upper CBO*, does not forgo any of the available information in the likelihood and/or in the prior.

Recovering the posterior: Possibilistic Bayesian inference can thus be understood as the optimization of a possibilistic analogue of the ELBO: the CBO. Unlike the probabilistic case, this optimization generally admits multiple solutions, but the standard maxitive posterior $g^\star_{\max}$ stands out as the natural choice – it is an extremum among the optimizers and serves as the canonical representative of the updated possibility distribution.

Recall that the standard Bayesian posterior is $\{q^\star_{\mathrm{add}}\} = \mathrm{argmax}_{q \in \mathcal{P}(\Theta)} \mathrm{ELBO}(q)$, while the two characterizations of the Bayesian posterior in possibility theory are:

$$g^\star_{\max} = \max\Big(\operatorname*{argmax}_{g \in \mathcal{F}(\Theta)} \underline{\mathrm{CBO}}(g)\Big), \tag{14a}$$

$$= \min\Big(\operatorname*{argmin}_{g \in \mathcal{F}(\Theta)} \overline{\mathrm{CBO}}(g)\Big). \tag{14b}$$

The additional $\max/\min$ operations in (14a) and (14b) add a layer of complexity to the recovery of the posterior possibility function. However, it holds that $g^\star_{\max}$ is the only element in

$$\operatorname*{argmax}_{g \in \mathcal{F}(\Theta)} \underline{\mathrm{CBO}}(g) \cap \operatorname*{argmin}_{g \in \mathcal{F}(\Theta)} \overline{\mathrm{CBO}}(g), \tag{15}$$

that is, the posterior possibility function is the unique element selected simultaneously by the lower and upper characterisations.

Relationship with VI: Restricting (14a) and (14b) to a family $\mathcal{G} \subseteq \mathcal{F}(\Theta)$ yields two variational approximations of the posterior. The lower CBO is qualitatively related to standard VI, tending to underestimate possibilities, whereas the upper CBO provides an alternative that favours more conservative, overestimated possibilities. This latter behaviour is more in line with general statistical principles which tend to prefer pessimistic uncertainty estimates over optimistic ones. Assuming the restricted optima are attained, the two formulations correspond to directed $D_{\max}$ projections:

$$\operatorname*{argmax}_{g \in \mathcal{G}} \underline{\mathrm{CBO}}(g) = \operatorname*{argmin}_{g \in \mathcal{G}} D_{\max}(g \| g^\star_{\max}), \tag{16}$$

$$\operatorname*{argmin}_{g \in \mathcal{G}} \overline{\mathrm{CBO}}(g) = \operatorname*{argmin}_{g \in \mathcal{G}} D_{\max}(g^\star_{\max} \| g). \tag{17}$$

The approximation gaps $\varepsilon_-(g) = D_{\max}(g \| g^\star_{\max})$ and $\varepsilon_+(g) = D_{\max}(g^\star_{\max} \| g)$ yield, when finite,

$$g(\theta) \le e^{\varepsilon_-(g)} g^\star_{\max}(\theta), \qquad g^\star_{\max}(\theta) \le e^{\varepsilon_+(g)} g(\theta). \tag{18}$$

Thus, restricted optimizers need not provide exact pointwise bounds unless the corresponding gap vanishes, and may be non-unique. Furthermore,

$$\overline{\mathrm{CBO}}(g) - \underline{\mathrm{CBO}}(g) = D_{\max}(g \| g^\star_{\max}) + D_{\max}(g^\star_{\max} \| g) \tag{19}$$

provides a symmetric two-sided approximation criterion within $\mathcal{G}$.

Relationship with Generalized VI: The lower and upper CBOs provide two variational approximations with opposite pointwise behaviour, the lower formulation tends to underestimate possibility degrees, whereas the upper formulation tends to overestimate them. Conceptually, this mirrors the way different probabilistic variational bounds induce different approximation biases. A more detailed comparison with generalized VI bounds is deferred to Appendix E.1.

5 Possibilistic VI with Exponential Families

We develop a general framework for possibilistic variational inference based on exponential families by considering their conjugate priors. All the proofs for this section can be found in Appendix D.2. We begin by defining the structure of exponential families in the possibilistic setting. Assuming that Θ is a subset of $\mathbb{R}^{d_\theta}$ and Λ is a convex subset of $\mathbb{R}^{d_\lambda}$, we follow the probabilistic approach and define an exponential family as a set $\mathcal{G} = \{g_\lambda \in \mathcal{F}(\Theta) : \lambda \in \Lambda\}$ where g_λ is of the form

$$g_\lambda(\theta) = \exp\big(\lambda^\top T(\theta) - A(\lambda) - B(\theta)\big) \tag{20}$$

for a given negative log carrier function B and sufficient statistics T on Θ and the corresponding log-partition function A on Λ, which ensures proper normalisation, that is

$$A(\lambda) = \log \sup_{\theta \in \Theta} \exp\big(\lambda^\top T(\theta) - B(\theta)\big) = \sup_{\theta \in \Theta} \lambda^\top T(\theta) - B(\theta). \tag{21}$$

More details about these general exponential families are given in Appendix F.

5.1 Bayesian Inference with Conjugate Family

We consider a likelihood $p(\cdot \mid \theta)$, $\theta \in \Theta$, that is part of a regular and minimal exponential family with log-partition A, that is $p(x \mid \theta) = \exp(\theta^\top T(x) - A(\theta) - B(x))$. The following proposition rephrases known results on the relationship between exponential families and the Bregman divergence (Banerjee et al., 2005) in the context of possibilistic inference.

Proposition 1. *For an observation x, assume $\lambda = T(x) \in \nabla A(\Theta)$, so that the MLE is finite and interior. The posterior possibility function under the uninformative prior on Θ is of the form*

$$g_\lambda(\theta) = \exp\big(\lambda^\top \theta - A^\dagger(\lambda) - A(\theta)\big) \tag{22a}$$

$$= \exp\big(-D_A(\theta \| \theta^\star(\lambda))\big), \tag{22b}$$

with $\theta^\star(\lambda) \doteq \mathbb{E}^\star_{g_\lambda}[\boldsymbol{\theta}]$ the maximum likelihood estimator (MLE), $A^\dagger$ the Legendre transform of the log-partition A, and $D_A(\theta \| \theta') = A(\theta) - A(\theta') - \nabla_\theta A(\theta')^\top(\theta - \theta')$ the Bregman divergence.

Identifying the Bregman divergence in g_λ allows us to leverage its properties. For instance, it is obvious from (22b) that g_λ is a possibility function with expected value $\theta^\star(\lambda)$. We can also use the known relationship between D_A and the KL divergence to see that $-\log g_\lambda(\theta) = \mathrm{KL}(p(\cdot \mid \theta^\star(\lambda)) \| p(\cdot \mid \theta))$. We now highlight some important properties of possibility functions of the same form as g_λ in the following proposition.

Proposition 2. *Let $\mathcal{G}_A(\Theta)$ be the subset of $\mathcal{F}(\Theta)$ defined as*

$$\mathcal{G}_A(\Theta) \doteq \big\{\theta \mapsto \exp(\lambda^\top \theta - A^\dagger(\lambda) - A(\theta)) : \lambda \in \Lambda\big\}, \tag{23}$$

where $\Lambda \doteq \nabla_\theta A(\Theta)$ is the image of Θ by $\nabla_\theta A$, then, for any $g_\lambda \in \mathcal{G}_A(\Theta)$, it holds that

1. *The dual possibility function $f_\theta : \lambda \mapsto g_\lambda(\theta)$ is in $\mathcal{G}_{A^\dagger}(\Lambda)$.*

2. *For any $\nu > 0$, $g_\lambda^\nu \in \mathcal{G}_{\nu A}(\Theta)$; for $\nu = 0$, set $g_\lambda^0 = \mathbf{1}$. In either case, it is a conjugate prior for the likelihood $p(\cdot \mid \theta)$, and the posterior for n i.i.d. observations $\{x_i\}_{i=1}^n$ is $g_{\lambda'}^{\nu'}$, where*

$$\lambda' = \frac{\nu\lambda + \sum_{i=1}^n T(x_i)}{\nu + n}, \quad \nu' = \nu + n. \tag{24}$$

provided $\lambda' \in \Lambda$. That is, λ is updated as the running mean of the sufficient statistics, and ν as the number of observations.

The first point in Proposition 2 is a specific property of the considered class of conjugate prior possibility functions. Although it has additional properties, the possibilistic conjugate prior is often simply the renormalised version of its probabilistic counterpart. The second point is a possibilistic analogue of the result of Diaconis and Ylvisaker (1979), with the additional features that the log-partition function is simply the Legendre transform of A and that the ν parameter is a discount factor rather than an arbitrary parameter. This property is intuitive: ν can be interpreted as how many observations the prior is worth in terms of information content – as in the probabilistic case; yet, the possibilistic treatment allows for considering $\nu = 0$, in which case $g_\lambda^\nu = \mathbf{1}$ and the prior is uninformative.

We have the following properties of $g_\lambda \in \mathcal{G}_A(\Theta)$ as a possibility function describing the uncertain variable $\boldsymbol{\theta}$ in Θ: (i) The expected value can be expressed as $\mathbb{E}_{g_\lambda}^\star[\boldsymbol{\theta}] = \nabla_\lambda A^\dagger(\lambda)$ and is the singleton $\{\theta^\star(\lambda)\}$ under the above assumptions, excluding the uninformative member, and (ii) the precision can be expressed as

$$\mathcal{I}_{g_\lambda}(\boldsymbol{\theta}) \doteq \mathbb{E}_{g_\lambda}^\star\left[-\nabla_\theta^2 \log g_\lambda(\boldsymbol{\theta})\right] = \nabla_\theta^2 A(\theta)\,|_{\theta=\theta^\star(\lambda)} = (\nabla_\lambda^2 A^\dagger(\lambda))^{-1}. \tag{25}$$

5.2 Variational Inference with Conjugate Family

We now demonstrate how to perform possibilistic VI with exponential family candidates of the form in (22a). For compactness, we write both consistency bounds using a sign variable $s \in \{+1, -1\}$. Define

$$\mathrm{CBO}_s(g) = \operatorname*{ext}_{\theta\in\Theta}\left\{-\ell(\theta) - \log\frac{g(\theta)}{\pi(\theta)}\right\}, \qquad \mathrm{ext} = \begin{cases}\sup, & s = +1,\\ \inf, & s = -1.\end{cases}$$

Thus $s = +1$ corresponds to the upper CBO, whereas $s = -1$ corresponds to the lower CBO. The associated objective is to minimize $s\,\mathrm{CBO}_s(g)$. We derive a simple update rule for optimizing arbitrary minimization objectives, e.g. $\hat{\ell}(\theta) = \ell(\theta) - \log\pi(\theta)$, for a loss function ℓ and a prior π.

In the following proposition, we reparameterize g_λ in terms of its mode, using the form in (22b), and establish a Newton-like update for the corresponding parameter.

Proposition 3. *Consider a candidate g_η of the form $g_\eta(\theta) = \exp(-D_A(\theta\|\eta))$ and let $H_s(\eta) \doteq s(\nabla^2\hat{\ell}(\eta) - \nabla^2 A(\eta))$. Assume $H_s(\eta) \succ 0$, so that the local quadratic approximation of the inner problem is concave for $s = +1$ and convex for $s = -1$. Assume that the inner extremum is finite and uniquely attained at $\theta_s \doteq \operatorname{argext}_{\theta\in\Theta}\{-\hat{\ell}(\theta) - \log g_\eta(\theta)\}$, under suitable compactness or coercivity conditions. Applying Danskin's theorem then gives*

$$\nabla\mathrm{CBO}_s(\eta) = \nabla^2 A(\eta)\,(\eta - \theta_s). \tag{26}$$

Approximating the inner optimizer using the 2nd-order Taylor expansion around η, and substituting it into the natural gradient of $s\,\mathrm{CBO}_s$ with metric $\nabla^2 A(\eta)$, yields the Newton-like update

$$\eta_{t+1} = \eta_t - \rho_t H_s(\eta_t)^{-1}\nabla\hat{\ell}(\eta_t), \tag{27}$$

with step size $\rho_t > 0$.

Remark 1. The use of natural gradient for the update follows similar motivation from Khan and Rue (2023). In fact, the update rule in (27) is equivalent to taking a mirror descent step with the approximate gradient in the space of the dual coordinate $\lambda = \nabla_\eta A(\eta)$, using mirror map $A^\dagger$. The natural gradient emerges naturally from the Legendre duality between the spaces of η and λ (Wainwright et al., 2008).

Remark 2. The condition $H_s(\eta) \succ 0$ ensures a nondegenerate local quadratic extremum and a well-defined inverse update; it is not necessary for the exact inner extremum to be finite. If the iterates converge to the unique global minimizer $\eta^\star$ of $\hat{\ell}$, the first moment matches that of the posterior in (6), and the second moment is matched when $\nabla^2 A(\eta^\star) = \nabla^2\hat{\ell}(\eta^\star)$; this is in the spirit of the Laplace approximation used for fitting Gaussian candidates.

Practical diagonal approximation. We now specialise the signed update to a Gaussian candidate $g_\eta(\theta) = \overline{\mathrm{N}}(\theta; \eta, \vartheta^{-1} I)$, so that $\nabla^2 A(\eta) = \vartheta I$. We also use an isotropic Gaussian prior with precision δ, equivalently an L_2 regularization term $\delta\|\eta\|^2/2$, which gives $\nabla\hat{\ell}(\eta) = \nabla\ell(\eta) + \delta\eta$ and $\nabla^2\hat{\ell}(\eta) = \nabla^2\ell(\eta) + \delta I$. For $s \in \{+1, -1\}$, the signed curvature gap is therefore $H_s(\eta) = s(\nabla^2\ell(\eta) + (\delta - \vartheta)I)$. Substituting these expressions into the signed CBO update gives

$$\eta_{t+1} = \eta_t - \rho_t\, s \left(\nabla^2\ell(\eta_t) + (\delta - \vartheta)I\right)^{-1}\left(\nabla\ell(\eta_t) + \delta\eta_t\right). \tag{28}$$

For neural-network objectives, the Hessian is not formed explicitly. We therefore use the standard diagonal squared-gradient approximation $\nabla^2\ell(\eta_t) \approx \mathrm{diag}\big(\nabla\ell(\eta_t) \odot \nabla\ell(\eta_t)\big)$, and replace the stochastic gradient and squared-gradient estimates by exponential moving averages (EMAs): with $g_t = \mathrm{EMA}(\hat{\nabla}\ell(\eta_t))$ and $h_t = \mathrm{EMA}(\hat{\nabla}\ell(\eta_t) \odot \hat{\nabla}\ell(\eta_t))$, where $\odot$ denotes the Hadamard product, we obtain an implementable signed update (see Alg. 1 for the case $s = +1$) as

$$\eta_{t+1} = \eta_t - \rho_t\, s\, \frac{\bar{g}_t + \delta\eta_t}{h_t + \delta - \vartheta}, \tag{29}$$

with elementwise division, and where $\bar{g}_t$ is the bias-corrected first moment based on g_t. In the case $s = +1$, the condition $\vartheta < \delta$ ensures that the diagonal preconditioner remains positive whenever h_t is non-negative, and the limiting case $\vartheta \to 0^+$ corresponds to ignoring the curvature of the Gaussian candidate. This diagonal squared-gradient approximation follows the same heuristic as Vprop and Vadam (Khan et al., 2018), while the use of exponential moving averages makes the resulting update reminiscent of Adam (Kingma and Ba, 2015) and IVON (Shen et al., 2024). For $s = -1$, the analogous curvature condition is instead $h_t + \delta - \vartheta < 0$ elementwise, corresponding to the opposite local curvature regime associated with the lower CBO. To further assess the effect of these approximations, we evaluate full-Hessian `CBOpt` on regression tasks without the diagonal and EMA approximations in Appendix J.4.

Relationship with Sharpness-Aware Minimization (SAM). The lower CBO admits a direct robust-optimization interpretation through SAM (Foret et al., 2020), which seeks to avoid sharp minima by minimizing the worst-case loss within a neighbourhood of the model parameters:

$$\min_{\eta} \sup_{|\theta-\eta|\le\rho} \hat{\ell}(\theta). \tag{30}$$

For the indicator possibility function $g_{\eta,\rho}(\theta) = \mathbf{1}_{|\theta-\eta|\le\rho}$, the lower CBO satisfies

$$\underline{\mathrm{CBO}}(g_{\eta,\rho}) = -\sup_{|\theta-\eta|\le\rho} \hat{\ell}(\theta). \tag{31}$$

Thus, maximizing the lower CBO recovers SAM exactly under an uninformative prior, where $\hat{\ell} = \ell$. With an L^2 prior, $\hat{\ell} = \ell - \log\pi$ additionally incorporates regularization. Changing the support geometry yields corresponding structured or anisotropic variants of SAM. The Gaussian candidate underlying the lower CBO optimizer replaces the hard neighbourhood with a quadratic penalty. In the isotropic case, whenever the inner supremum is finite,

$$-\underline{\mathrm{CBO}}(g_\eta) = \sup_{\theta}\left\{\hat{\ell}(\theta) - \frac{\vartheta}{2}|\theta - \eta|^2\right\}. \tag{32}$$

Thus, the lower CBO update rule in Proposition 3 yields a local second-order update for a quadratically penalized SAM objective. For neural-network losses, the global supremum need not be finite, so the practical update is understood locally.

6 Experiments

In this section, we evaluate the efficacy of the CBO update rule derived in Proposition 3. We consider three variants: (1) **u**pper **CBO** **Opt**imization, denoted by `uCBOpt` (Alg. 1); (2) `uCBOpt` with adaptive curvature, denoted by `uCBOpt-adapt` (Alg. 2); and (3) **l**ower **CBO** **Opt**imization with adaptive curvature, denoted by `lCBOpt-adapt` (Alg. 3). In `uCBOpt`, we use a fixed curvature parameter ϑ. In `uCBOpt-adapt` and `lCBOpt-adapt`, we instead make the curvature parameter adaptive using

the diagonal precision $S_t = \mathrm{diag}(\gamma c_t)$, where c_t tracks a one-sided envelope of the data-plus-prior curvature $h_t + \delta$. We provide additional details on the algorithms in Appendix G.

We evaluate our methods by training neural networks on image classification tasks, closely following a subset of the experiments in Shen et al. (2024). We perform extensive hyperparameter tuning on Fashion-MNIST and tune the `CBOpt` learning rate and fixed curvature on CIFAR-10; other larger-dataset settings follow Shen et al. (2024). Additional training details are provided in Appendix H. For the classification experiments, we run each method with three random seeds and report the mean and standard deviation. We present the main results in t , while the full results, including additional metrics, final-epoch performance, and training and calibration plots, are presented in Appendix I. Ablation studies are also included in Appendix J.

Baselines. We evaluate our methods against six baseline approaches: SGD, AdamW (Loshchilov and Hutter, 2019), IVON (Shen et al., 2024), MC Dropout (MC-D) (Gal and Ghahramani, 2016), linearized last-layer Laplace (referred to as Laplace) (Daxberger et al., 2021), and SWAG (Maddox et al., 2019). AdamW and SGD serve as deterministic optimization baselines, whereas IVON, MC-D, Laplace and SWAG serve as uncertainty-aware baselines; these are discussed in Appendix A. For IVON, we additionally report predictions obtained from the posterior mean parameters, denoted by IVON@mean. For CIFAR-100, we consider only four baselines, as our primary objective is to demonstrate that our optimizers can scale effectively to larger models and datasets.

Algorithm 1 **u**pper **CBO** **Opt**imisation (`uCBOpt`)

Require: Learning rates $\{\rho_t\}$, weight decay $\delta > 0$, curvature parameter $\vartheta \in (0, \delta)$, momentum parameters $\beta_1, \beta_2 \in [0, 1)$, and Hessian init $h_0 > 0$.
Init: $\eta \leftarrow$ (NN-weights), $h \leftarrow h_0$, $g \leftarrow 0$.
for $t = 1, 2, \ldots$ **do**
  $\hat{g} \leftarrow \widehat{\nabla}\, \ell(\eta)$ # Stochastic gradient
  $\hat{h} \leftarrow \hat{g} \odot \hat{g}$
  $g \leftarrow \beta_1 g + (1 - \beta_1)\hat{g}$
  $h \leftarrow \beta_2 h + (1 - \beta_2)\hat{h}$
  $\bar{g} \leftarrow g/(1 - \beta_1^t)$
  $\eta \leftarrow \eta - \rho_t(\bar{g} + \delta\eta)/(h + \delta - \vartheta)$
end for
return η

Experiments. As in-domain tasks, we train LeNet on Fashion-MNIST, ResNet-20 on CIFAR-10, and DenseNet-121 on CIFAR-100. For the out-of-domain (OOD) tasks, we use EMNIST as the OOD dataset for LeNet trained on Fashion-MNIST, SVHN for ResNet-20 trained on CIFAR-10, and TinyImageNet for DenseNet-121 trained on CIFAR-100.

Results. In Table 1, the reported uCBOpt results correspond to $\vartheta = 8 \cdot 10^{-6}$ for Fashion-MNIST, $\vartheta \to 0^+$ for CIFAR-10, and $\vartheta = 5 \cdot 10^{-5}$ for CIFAR-100. Using validation-selected checkpoints, on Fashion-MNIST, the `CBOpt` variants achieve the strongest in-domain performance overall. In particular, `lCBOpt-adapt` obtains the best accuracy (0.913) and NLL (0.252), while `uCBOpt-adapt` performs comparably. For OOD detection on EMNIST, `uCBOpt` also matches MC Dropout for the lowest FPR@95. On CIFAR-10, the `CBOpt` variants remain competitive with the strongest baselines, reaching accuracies around 0.910, comparable to SGD and Laplace, and achieving OOD detection performance close to SWAG and better mean scores than MC Dropout. It is worth noting that MC Dropout uses additional dropout layers, which may provide extra regularization, and that IVON, MC Dropout and SWAG use multiple stochastic predictions at inference time. In contrast, our method does not require such sampling at test time. Nevertheless, as shown in Appendix K, combining `CBOpt` with MC Dropout preserves the in-domain performance of MC Dropout while improving OOD performance. The benefits of `CBOpt` become more pronounced on CIFAR-100. In this larger-scale setting, the `CBOpt` variants improve several in-domain and out-of-domain metrics, although IVON and MC dropout achieve lower ECE. These results suggest that `CBOpt` scales effectively to larger models and datasets. Overall, the `CBOpt` family is competitive with IVON, the most directly comparable variational baseline, improving predictive performance and OOD detection on several tasks, while IVON often achieves lower ECE.

7 Discussion

Possibilistic Bayesian inference is already an optimization problem, which seems to limit the interest in the development of possibilistic VI at first sight. On the contrary, we have shown that there is a surprising number of subtleties to be understood and of insights to be gained when exploring the

Table 1: Experimental results on in-domain and out-of-domain tasks over three runs. The best result and all results within one standard deviation of the best-performing method are bolded.

		In-domain			Out-of-domain		
Dataset	Optimizer	Acc. ↑	NLL ↓	ECE ↓	FPR@95 ↓	AUROC ↑	AUPR-Out ↑
In: Fashion-MNIST **Out:** EMNIST	AdamW	$0.900_{\pm 0.005}$	$0.294_{\pm 0.007}$	$0.026_{\pm 0.004}$	$0.827_{\pm 0.006}$	$0.745_{\pm 0.009}$	$0.834_{\pm 0.003}$
	SGD	$0.896_{\pm 0.005}$	$0.293_{\pm 0.005}$	$0.022_{\pm 0.004}$	$0.816_{\pm 0.013}$	$0.767_{\pm 0.022}$	$0.845_{\pm 0.012}$
	IVON@mean	$0.907_{\pm 0.001}$	$0.267_{\pm 0.005}$	$0.018_{\pm 0.008}$	$0.784_{\pm 0.036}$	$0.787_{\pm 0.029}$	$0.861_{\pm 0.019}$
	IVON	$\mathbf{0.911}_{\pm 0.003}$	$0.258_{\pm 0.001}$	$\mathbf{0.013}_{\pm 0.004}$	$\mathbf{0.753}_{\pm 0.040}$	$0.804_{\pm 0.030}$	$0.873_{\pm 0.020}$
	MC-D	$0.897_{\pm 0.003}$	$0.281_{\pm 0.007}$	$\mathbf{0.013}_{\pm 0.003}$	$\mathbf{0.730}_{\pm 0.024}$	$\mathbf{0.821}_{\pm 0.008}$	$\mathbf{0.885}_{\pm 0.007}$
	Laplace	$0.895_{\pm 0.004}$	$0.295_{\pm 0.009}$	$0.024_{\pm 0.003}$	$0.805_{\pm 0.056}$	$0.770_{\pm 0.039}$	$0.848_{\pm 0.031}$
	SWAG	$0.900_{\pm 0.003}$	$0.342_{\pm 0.009}$	$0.043_{\pm 0.003}$	$0.855_{\pm 0.008}$	$0.731_{\pm 0.005}$	$0.820_{\pm 0.002}$
	`uCBOpt`	$0.906_{\pm 0.003}$	$0.263_{\pm 0.006}$	$0.018_{\pm 0.006}$	$\mathbf{0.739}_{\pm 0.026}$	$0.810_{\pm 0.012}$	$\mathbf{0.879}_{\pm 0.010}$
	`uCBOpt-adapt`	$\mathbf{0.910}_{\pm 0.001}$	$\mathbf{0.256}_{\pm 0.003}$	$\mathbf{0.016}_{\pm 0.001}$	$0.783_{\pm 0.009}$	$0.773_{\pm 0.015}$	$0.857_{\pm 0.009}$
	`lCBOpt-adapt`	$\mathbf{0.913}_{\pm 0.002}$	$\mathbf{0.252}_{\pm 0.004}$	$0.020_{\pm 0.000}$	$0.794_{\pm 0.016}$	$0.784_{\pm 0.019}$	$0.859_{\pm 0.011}$
In: CIFAR-10 **Out:** SVHN	AdamW	$0.872_{\pm 0.001}$	$0.406_{\pm 0.002}$	$0.042_{\pm 0.003}$	$0.840_{\pm 0.047}$	$0.820_{\pm 0.031}$	$0.881_{\pm 0.021}$
	SGD	$0.909_{\pm 0.003}$	$0.308_{\pm 0.011}$	$0.039_{\pm 0.008}$	$\mathbf{0.732}_{\pm 0.042}$	$0.864_{\pm 0.013}$	$\mathbf{0.918}_{\pm 0.011}$
	IVON@mean	$0.892_{\pm 0.003}$	$0.358_{\pm 0.011}$	$0.045_{\pm 0.008}$	$0.827_{\pm 0.025}$	$0.830_{\pm 0.011}$	$0.890_{\pm 0.010}$
	IVON	$0.899_{\pm 0.004}$	$0.317_{\pm 0.004}$	$0.028_{\pm 0.005}$	$0.812_{\pm 0.042}$	$0.836_{\pm 0.014}$	$0.894_{\pm 0.012}$
	MC-D	$\mathbf{0.923}_{\pm 0.002}$	$\mathbf{0.230}_{\pm 0.006}$	$\mathbf{0.010}_{\pm 0.002}$	$0.795_{\pm 0.053}$	$0.847_{\pm 0.024}$	$0.902_{\pm 0.017}$
	Laplace	$0.909_{\pm 0.001}$	$0.303_{\pm 0.008}$	$0.035_{\pm 0.006}$	$\mathbf{0.759}_{\pm 0.047}$	$0.852_{\pm 0.019}$	$0.909_{\pm 0.013}$
	SWAG	$0.918_{\pm 0.001}$	$0.251_{\pm 0.004}$	$\mathbf{0.009}_{\pm 0.000}$	$\mathbf{0.728}_{\pm 0.035}$	$\mathbf{0.879}_{\pm 0.007}$	$\mathbf{0.924}_{\pm 0.007}$
	`uCBOpt`	$0.910_{\pm 0.004}$	$0.296_{\pm 0.009}$	$0.038_{\pm 0.001}$	$\mathbf{0.745}_{\pm 0.044}$	$0.869_{\pm 0.012}$	$\mathbf{0.918}_{\pm 0.010}$
	`uCBOpt-adapt`	$0.909_{\pm 0.005}$	$0.300_{\pm 0.016}$	$0.038_{\pm 0.004}$	$\mathbf{0.759}_{\pm 0.028}$	$0.866_{\pm 0.010}$	$0.915_{\pm 0.007}$
	`lCBOpt-adapt`	$0.909_{\pm 0.005}$	$0.299_{\pm 0.002}$	$0.037_{\pm 0.002}$	$\mathbf{0.760}_{\pm 0.016}$	$0.861_{\pm 0.006}$	$0.913_{\pm 0.003}$
In: CIFAR-100 **Out:** TinyImageNet	SGD	$0.649_{\pm 0.004}$	$1.334_{\pm 0.006}$	$0.096_{\pm 0.006}$	$0.848_{\pm 0.003}$	$0.720_{\pm 0.008}$	$0.684_{\pm 0.006}$
	IVON@mean	$0.599_{\pm 0.001}$	$1.521_{\pm 0.008}$	$0.092_{\pm 0.006}$	$0.865_{\pm 0.006}$	$0.704_{\pm 0.005}$	$0.666_{\pm 0.005}$
	IVON	$0.607_{\pm 0.002}$	$1.458_{\pm 0.012}$	$\mathbf{0.071}_{\pm 0.004}$	$0.868_{\pm 0.008}$	$0.709_{\pm 0.007}$	$0.672_{\pm 0.007}$
	MC-D	$0.649_{\pm 0.016}$	$1.298_{\pm 0.016}$	$\mathbf{0.074}_{\pm 0.020}$	$0.858_{\pm 0.008}$	$0.718_{\pm 0.008}$	$0.681_{\pm 0.009}$
	`uCBOpt`	$\mathbf{0.671}_{\pm 0.014}$	$\mathbf{1.231}_{\pm 0.005}$	$0.095_{\pm 0.014}$	$0.831_{\pm 0.002}$	$\mathbf{0.727}_{\pm 0.003}$	$0.693_{\pm 0.003}$
	`uCBOpt-adapt`	$0.655_{\pm 0.011}$	$1.298_{\pm 0.011}$	$0.096_{\pm 0.021}$	$\mathbf{0.828}_{\pm 0.002}$	$\mathbf{0.729}_{\pm 0.002}$	$\mathbf{0.696}_{\pm 0.001}$
	`lCBOpt-adapt`	$\mathbf{0.674}_{\pm 0.007}$	$1.268_{\pm 0.018}$	$0.110_{\pm 0.002}$	$0.836_{\pm 0.009}$	$\mathbf{0.729}_{\pm 0.005}$	$0.692_{\pm 0.006}$

properties of possibilistic VI in general as well as specifically through the lens of a possibilistic version of exponential families and conjugate priors. And indeed, even when closely following existing treatments of VI and its applications, many differences emerge, such as regularities in conjugate families and a new route leading to novel optimization methods.

In particular, the `CBOpt` family of optimizers developed from possibilistic VI demonstrates that this framework is not merely a reformulation, but can lead to practically useful algorithms. From a practical viewpoint, the scalar parameter ϑ can be interpreted as modifying the effective curvature or regularization strength appearing in the denominator of the update. In the adaptive variants, this contribution changes throughout training. Our theory therefore separates this effect from the standard weight-decay term, providing additional flexibility in the update. Empirically, this flexibility leads to competitive performance across in-domain and out-of-domain tasks, suggesting that possibilistic VI offers not only a conceptually distinct perspective on uncertainty representation, but also a practical foundation for developing scalable uncertainty-aware learning algorithms.

Limitations. The Newton-like update of Proposition 3 is based on a second-order Taylor expansion of the inner extremum problem and on the curvature condition $H_s \succ 0$, which couples the admissible range of the candidate precision ϑ to the local geometry of $\hat{\ell}$. Local curvature conditions do not ensure a finite global inner extremum: deep-network losses can outgrow the quadratic penalty in the lower CBO. The practical algorithms are therefore local approximate updates. Our exponential-family treatment further assumes that A is proper, closed, and convex with $A^{\dagger\dagger} = A$, and that $\mathbb{E}^{\star}_{q_\lambda}[\theta]$ is a singleton, excluding the fully uninformative limit. Our classification experiments use isotropic or diagonal Gaussian candidates with a squared-gradient curvature surrogate, leaving non-Gaussian candidates untested. The full-Hessian regression experiments provide complementary evidence, but do not isolate the effects of the diagonal and EMA approximations. In the reported lower-adaptive parameter regime, the update reduces to a rescaled zero-curvature update, as explained in Appendix J.

Acknowledgment

This research is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RS02/24), and by the Singapore Ministry of Digital Development and Information under the AI Visiting Professorship Programme (AIVP-2024-004). Jasraj Singh acknowledges support from the Independent Research Fund Denmark (Grant no. 5334-00122B).

References

Banerjee, A., Merugu, S., Dhillon, I. S., and Ghosh, J. (2005). Clustering with Bregman divergences. *Journal of machine learning research*, 6(Oct):1705–1749.

Bengs, V., Hüllermeier, E., and Waegeman, W. (2022). Pitfalls of epistemic uncertainty quantification through loss minimisation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France. PMLR.

Caprio, M., Dutta, S., Jang, K. J., Lin, V., Ivanov, R., Sokolsky, O., and Lee, I. (2024a). Credal Bayesian deep learning. *Transactions on Machine Learning Research*.

Caprio, M., Sultana, M., Elia, E. G., and Cuzzolin, F. (2024b). Credal learning theory. *Advances in Neural Information Processing Systems*, 37:38665–38694.

Catoni, O. (2004). Statistical learning theory and stochastic optimization. Saint-Flour summer school on probability theory 2001 (Jean Picard ed.). *Lecture Notes in Mathematics. Springer*, 2:10.

Cella, L. and Martin, R. (2025). Computationally efficient variational-like approximations of possibilistic inferential models. *International Journal of Approximate Reasoning*, 186:109506.

Charpentier, B., Borchert, O., Zügner, D., Geisler, S., and Günnemann, S. (2022). Natural posterior network: Deep Bayesian predictive uncertainty for exponential family distributions. In *International Conference on Learning Representations*.

Charpentier, B., Zügner, D., and Günnemann, S. (2020). Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in neural information processing systems*, 33:1356–1367.

Chau, S. L., Caprio, M., and Muandet, K. (2025). Integral imprecise probability metrics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. (2017). Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE.

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., and Hennig, P. (2021). Laplace redux - effortless Bayesian deep learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20089–20103. Curran Associates, Inc.

Dempster, A. P. (1968). A generalization of Bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(2):205–232.

Deng, D., Chen, G., Yu, Y., Liu, F., and Heng, P.-A. (2023). Uncertainty estimation by fisher information-based evidential deep learning. In *International conference on machine learning*, pages 7596–7616. PMLR.

Diaconis, P. and Ylvisaker, D. (1979). Conjugate priors for exponential families. *The Annals of statistics*, pages 269–281.

Dieng, A. B., Tran, D., Ranganath, R., Paisley, J., and Blei, D. (2017). Variational inference via χ upper bound minimization. *Advances in Neural Information Processing Systems*, 30.

Donsker, M. D. and Varadhan, S. R. S. (1976). Asymptotic evaluation of certain Markov process expectations for large time. iii. *Communications on Pure and Applied Mathematics*, 28:389–461.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. (2020). Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Hieu, N. M., Houssineau, J., Chada, N. K., and Delande, E. (2025). Decoupling epistemic and aleatoric uncertainties with possibility theory. In *The 28th International Conference on Artificial Intelligence and Statistics*, pages 2899–2907. ML Research Press.

Houssineau, J. (2021). A linear algorithm for multi-target tracking in the context of possibility theory. *IEEE Transactions on Signal Processing*, 69:2740–2751.

Houssineau, J. and Nott, D. J. (2022). Robust Bayesian inference in complex models with possibility theory. *arXiv preprint arXiv:2204.06911*.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708.

Jøsang, A. (2016). *Subjective logic*, volume 3. Springer.

Khan, M., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. (2018). Fast and scalable Bayesian deep learning by weight-perturbation in Adam. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2611–2620. PMLR.

Khan, M. E. and Rue, H. (2023). The Bayesian learning rule. *Journal of Machine Learning Research*, 24(281):1–46.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Kingma, D. P., Salimans, T., and Welling, M. (2015). Variational dropout and the local reparameterization trick. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational Bayes. In *International Conference on Learning Representations*.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Krueger, D., Huang, C.-W., Islam, R., Turner, R., Lacoste, A., and Courville, A. (2017). Bayesian hypernetworks. *arXiv preprint arXiv:1710.04759*.

Kullback, S. (1959). *Information Theory and Statistics*. John Wiley & Sons.

Kullback, S. and Leibler, R. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Le, Y. and Yang, X. (2015). Tiny imagenet visual recognition challenge. Stanford CS231N.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, Y. and Turner, R. E. (2016). Rényi divergence variational inference. *Advances in neural information processing systems*, 29.

Litvinov, G. L., Maslov, V. P., and Shpiz, G. B. (2001). Idempotent functional analysis: an algebraic approach. *Mathematical Notes*, 69(5):696–729.

Löhr, T., Hofman, P., Mohr, F., and Hüllermeier, E. (2025). Credal prediction based on relative likelihood. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2019). A simple baseline for Bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pages 13153–13164.

Malinin, A. and Gales, M. (2018). Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31.

Martin, R. (2022). Valid and efficient imprecise-probabilistic inference with partial priors, I. First results. *arXiv preprint arXiv:2203.06703*.

Martin, R. (2026). Possibilistic inferential models: A review. *Journal of the American Statistical Association*, 121(553):807–826.

Martin, R. and Liu, C. (2013). Inferential models: A framework for prior-free posterior probabilistic inference. *Journal of the American Statistical Association*, 108(501):301–313.

Mobiny, A., Yuan, P., Moulik, S. K., Garg, N., Wu, C. C., and Van Nguyen, H. (2021). Dropconnect is effective in modeling uncertainty of Bayesian deep networks. *Scientific reports*, 11(1):5458.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Ni, Y., Houssineau, J., Ong, Y. S., and Koniusz, P. (2026). Possibilistic predictive uncertainty for deep learning. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR. To appear.

Ristic, B., Houssineau, J., and Arulampalam, S. (2020). Target tracking in the framework of possibility theory: The possibilistic Bernoulli filter. *Information Fusion*, 62:81–88.

Rudner, T. G., Chen, Z., Teh, Y. W., and Gal, Y. (2022). Tractable function-space variational inference in Bayesian neural networks. *Advances in Neural Information Processing Systems*, 35:22686–22698.

Salimans, T., Kingma, D., and Welling, M. (2015). Markov chain Monte Carlo and variational inference: Bridging the gap. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1218–1226, Lille, France. PMLR.

Sensoy, M., Kaplan, L., and Kandemir, M. (2018). Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, IL.

Shen, Y., Daheim, N., Cong, B., Nickl, P., Marconi, G. M., Raoul, B. C. E. M., Yokota, R., Gurevych, I., Cremers, D., Khan, M. E., et al. (2024). Variational learning is effective for large deep networks. In *International Conference on Machine Learning*, pages 44665–44686. PMLR.

Verdoja, F. and Kyrki, V. (2020). Notes on the behavior of MC dropout. *CoRR*, abs/2008.02627.

Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305.

Wang, K., Cuzzolin, F., Manchingal, S. K., Shariatmadar, K., Moens, D., and Hallez, H. (2024). Credal deep ensembles for uncertainty quantification. *Advances in Neural Information Processing Systems*, 37:79540–79572.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xue, Z., Cai, H., Houssineau, J., and Zhang, J. (2025). Orbit-attitude coupled control for multitarget tracking based on partition pattern search. *IEEE Transactions on Aerospace and Electronic Systems*, 61(4):10855–10867.

Yoon, T. and Kim, H. (2025). Uncertainty estimation by flexible evidential deep learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3–28.

Zellner, A. (1988). Optimal information processing and Bayes's theorem. *The American Statistician*, 42(4):278–280.

Zhang, S., Ni, Y., Du, J., Xue, Y., Torr, P., Koniusz, P., and Van Den Hengel, A. (2025). Open-world objectness modeling unifies novel object detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30332–30342.

Supplementary Material

Table of Contents

A Related Work **16**

B Notations **17**

C More on Possibility Theory **17**

D Proofs **19**

D.1 Proof of Theorem 2 19

D.2 Proofs of Results in Section 5 22

E Discussion on Max-Relative Entropy **24**

E.1 Relationship with Generalized VI 24

E.2 Intuition Through Visualization 24

E.3 Example: Bayesian Logistic Regression 25

F Possibilistic Exponential Family **26**

G Algorithms **27**

H Experimental Details **28**

H.1 Computational Efficiency 28

H.2 Architectures and Datasets 29

H.3 Hyperparameters 30

H.4 Performance Metrics 31

I Full Experimental Results **32**

J Ablation Studies **36**

J.1 Varying ϑ in `uCBOpt` 36

J.2 Varying γ and β_3 in `uCBOpt-adapt` 37

J.3 Varying γ and β_3 in `lCBOpt-adapt` 37

J.4 Full-Hessian `CBOpt` for Regression Tasks 38

K Additional Experiment: Combining `CBOpt` with MC Dropout **40**

A Related Work

In this section, we position our work with respect to current literature.

Possibilistic Variational Inference. Our work is closest to Cella and Martin (2025), which proposes a "variational-like" Monte-Carlo–driven strategy for approximating the α-cut of an Inferential Model's (Martin and Liu, 2013) possibility contour with a $100(1-\alpha)\%$ credible set of a candidate family; see Martin (2026) for a review on IMs. Their framework ensures that the resulting approximations inherit the finite-sample validity properties of IMs (Martin, 2022). In contrast, we take a more standard approach, framing approximate inference as minimizing divergence from the Bayes posterior. Specifically, building on the established supremum-based operations in max-plus and idempotent analysis (Litvinov et al., 2001), we derive a maxitive analogue of the Donsker-Varadhan formulation, which enables direct optimization over a chosen candidate family, analogous to probabilistic VI. Importantly, the optimization problem is well-suited for modern autograd engines designed for large-scale compositional models, like neural networks.

Epistemic Uncertainty in Deep Learning. Neural networks are notoriously susceptible to overconfidence, presenting a significant hurdle for their deployment in safety-critical applications. Standard models often assign high predictive certainty to incorrect outputs, particularly when encountering out-of-distribution (OOD) data (Zhang et al., 2025). Accordingly, *Epistemic Uncertainty* (EU) quantification, which seeks to represent the model's lack of knowledge rather than the inherent noise in the data, has garnered attention from the research community. While Bayesian neural networks offer a theoretically grounded approach to this problem (Blundell et al., 2015; Gal and Ghahramani, 2016; Krueger et al., 2017; Lakshminarayanan et al., 2017; Mobiny et al., 2021; Rudner et al., 2022), their high computational complexity has spurred the development of more efficient second-order predictive models. These methods, such as Prior Networks (Malinin and Gales, 2018), PostNet (Charpentier et al., 2020), and Natural Posterior Networks (Charpentier et al., 2022), bypass parameter-space inference by directly modelling distributions over predictions.

A prominent subset of these second-order approaches is Evidential Deep Learning (EDL) (Sensoy et al., 2018). EDL draws upon Subjective Logic (Jøsang, 2016) and the Dempster-Shafer Theory (Dempster, 1968; Shafer, 1976) to treat predictions as "opinions" supported by evidence. While various iterations have sought to refine this framework (Deng et al., 2023; Yoon and Kim, 2025), many EDL variants rely on heuristic objectives that lack a rigorous probabilistic derivation (Bengs et al., 2022). Consequently, establishing a principled connection between these evidential objectives and formal epistemic uncertainty remains an active area of research.

This gap has motivated recent efforts to ground second-order prediction in possibility theory rather than in subjective logic or belief functions. The work of Ni et al. (2026) takes this direction by using the maxitive pseudo-divergence and the maxitive Bayes posterior under the uninformative possibilistic prior, both used here, to derive a Dirichlet-based loss for image classification. Their formulation specializes the present framework to the probability simplex, projecting parameter-space possibility onto prediction space via the supremum-based change of variable, and yields a closed-form inner maximizer that is competitive with state-of-the-art EDL methods. Their work and ours are complementary: where they take selected components and apply them to a specific second-order prediction problem, we develop the underlying variational principle, including the maxitive Donsker-Varadhan duality, the dual CBO formulations, and the exponential-family treatment, and study its consequences for general possibilistic inference and learning.

Imprecise probability. There exists a formal correspondence between possibility measures and credal sets: a given possibility measure characterizes a unique, closed, and convex set of probability measures that it upper bounds. This establishes possibilistic modelling as a mathematically rigorous instance of imprecise probability theory, a framework that has recently attracted growing interest in machine learning for principled epistemic uncertainty quantification, including credal generalization bounds (Caprio et al., 2024b), credal-set neural networks for out-of-distribution detection (Wang et al., 2024), imprecise Bayesian deep learning (Caprio et al., 2024a), likelihood-based credal prediction (Löhr et al., 2025), and metrics over imprecise probabilistic objects (Chau et al., 2025). Accordingly, our maxitive-VI formulation (Theorem 2) opens up a complementary line of research within this broader programme, providing a variational principle through which the comprehensive set of tools from imprecise probability theory can be applied to epistemic uncertainty quantification in deep learning.

Computational Efficiency. Bayesian deep learning provides a principled framework for uncertainty quantification, yet its practical application is often bottlenecked by the high computational cost of marginalizing over the parameters. Exact inference remains intractable for modern neural networks, necessitating the use of approximations that balance statistical rigour with hardware constraints. Here, we discuss a few popular strategies that we benchmark against:

- Monte-Carlo Dropout (MC-D) (Gal and Ghahramani, 2016) interprets the use of dropout at test time as a form of VI, simulating a model ensemble. It uses standard dropout during training and multiple stochastic forward passes at test time, but has been known to provide unreliable uncertainty estimates (Verdoja and Kyrki, 2020).
- Last-layer Laplace approximation (Laplace) (Daxberger et al., 2021) fits a Gaussian approximation to the posterior over the final-layer weights after maximum *a posteriori* (MAP) training. It is computationally efficient and often improves uncertainty estimation, but its posterior uncertainty is restricted to the last layer.
- Stochastic Weight Averaging Gaussian (SWAG) (Maddox et al., 2019) fits a low-rank Gaussian distribution to the trajectory of iterates generated by SGD. This is computationally efficient because it captures weight-space statistics during a standard training run.
- Improved Variational Online Newton (IVON) (Shen et al., 2024) is a second-order variational optimizer that approximates a Gaussian posterior. It is computationally efficient, matching the complexity of first-order optimizers, while providing well-calibrated uncertainty estimates.

In comparison to these approaches, our `CBOpt` models the network weights using a possibility function. Similar to the diagonal Gaussian approximations used in SWAG-diagonal and IVON, we adopt a mean-field approximation for performing VI, which is computationally efficient and enables scaling our methodology to large-scale learning problems.

B Notations

In Table 2, we present some of the notations used in this work.

Table 2: Summary of key notations

Notation	Description
$\mathbb{E}_f^\star$	Possibilistic expectation (mode) of f
$\mathcal{I}_f$	Possibilistic precision of f
$A^\dagger$	Legendre transform of A
$\hat{\ell}$	Regularized loss: $\ell(\theta) - \log \pi(\theta)$
$\overline{\mathrm{CBO}}$	Upper consistency bound
$\underline{\mathrm{CBO}}$	Lower consistency bound
$\theta^\star(\lambda)$	Mode / expected value of g_λ, that is $\mathbb{E}_{g_\lambda}^\star[\boldsymbol{\theta}]$

C More on Possibility Theory

To define epistemic uncertainty formally, we consider a sample space Ω whose elements characterise all the possible values of the relevant unknown quantities. Instead of equipping Ω with a probabilistic structure, we simply describe an unknown parameter θ_0 in a set Θ via a (deterministic) *uncertain variable* $\boldsymbol{\theta} : \Omega \to \Theta$. If an element $\omega \in \Omega$ were the correct one, then $\boldsymbol{\theta}(\omega)$ would be the true value of the parameter θ_0. To *describe* the available information about $\boldsymbol{\theta}$, we define a *possibility function* $f_{\boldsymbol{\theta}}$ (a.k.a. possibility distribution) on Θ as a non-negative function that verifies $\sup_{\theta \in \Theta} f_{\boldsymbol{\theta}}(\theta) = 1$. The possibility of an event $\boldsymbol{\theta} \in A$ for some $A \subseteq \Theta$ is then $\sup_{\theta \in A} f_{\boldsymbol{\theta}}(\theta)$.

Values and support. Since $f_{\boldsymbol{\theta}}$ is non-negative and $\sup_{\theta\in\Theta} f_{\boldsymbol{\theta}}(\theta) = 1$, the values of any possibility function lie in $[0, 1]$, although the supremum need not be attained. The support of f is the set $\operatorname{supp}(f_{\boldsymbol{\theta}}) \doteq \{\theta \in \Theta : f_{\boldsymbol{\theta}}(\theta) > 0\}$ of values that are not deemed impossible. Strict positivity is sometimes convenient; zeros are treated using the support conventions stated in (13).

Necessity. Each possibility function f_θ comes equipped with a dual necessity measure, defined for any $A \subseteq \Theta$ by

$$N_{\boldsymbol{\theta}}(A) \doteq 1 - \sup_{\theta\in A^c} f_{\boldsymbol{\theta}}(\theta). \tag{33}$$

Whereas $\sup_{\theta\in A} f_{\boldsymbol{\theta}}(\theta)$ quantifies how plausible the event $\theta \in A$ is, $N_{\boldsymbol{\theta}}(A)$ quantifies how strongly the available information rules out its complement. An event with low necessity is not necessarily implausible, only insufficiently supported by the available information, a key distinction for the conservative treatment of epistemic uncertainty in possibility theory.

Independence. Two uncertain variables $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ on Θ and Ψ are said to be independent (or independently described) when their joint possibility function factorises as

$$f_{\boldsymbol{\theta},\boldsymbol{\psi}}(\theta, \psi) = f_{\boldsymbol{\theta}}(\theta) f_{\boldsymbol{\psi}}(\psi), \qquad \forall (\theta, \psi) \in \Theta \times \Psi. \tag{34}$$

This convention mirrors the probabilistic notion of independence and is the one adopted throughout the paper. A weaker, more conservative alternative occasionally found in the literature replaces the product by a minimum but is not used here.

Marginalization and conditioning. Possibility functions behave similarly to probability mass functions, except that summation is replaced by a maximum. For instance, if $\boldsymbol{\theta}$ and another uncertain variable $\boldsymbol{\psi}$ on a set Ψ are jointly described by a possibility function $f_{\boldsymbol{\theta},\boldsymbol{\psi}}$, then the *marginal* possibility function describing $\boldsymbol{\theta}$ is characterised by

$$f_{\boldsymbol{\theta}}(\theta) = \sup_{\psi\in\Psi} f_{\boldsymbol{\theta},\boldsymbol{\psi}}(\theta, \psi), \qquad \forall \theta \in \Theta. \tag{35}$$

Similarly, for a fixed $\psi \in \Psi$ satisfying $f_{\boldsymbol{\psi}}(\psi) > 0$, the *conditional* possibility function of $\boldsymbol{\theta}$ given $\boldsymbol{\psi} = \psi$ is characterised by

$$f_{\boldsymbol{\theta}}(\theta \mid \boldsymbol{\psi} = \psi) = \frac{f_{\boldsymbol{\theta},\boldsymbol{\psi}}(\theta, \psi)}{f_{\boldsymbol{\psi}}(\psi)}, \qquad \forall \theta \in \Theta. \tag{36}$$

As is standard with probability distributions, we will often omit which uncertain variable is being described by a possibility function and simply write, e.g. f instead of $f_{\boldsymbol{\theta}}$.

Bayesian inference. Given an observed likelihood $L(\theta) \propto p(x \mid \theta)$ and a prior possibility function π, likelihood-based possibilistic updating multiplies the prior by the likelihood and normalises, yielding, when the denominator is positive and finite,

$$g^\star_{\max}(\theta) = \frac{p(x \mid \theta)\, \pi(\theta)}{\sup_{\theta'} p(x \mid \theta')\, \pi(\theta')}, \tag{37}$$

which is the posterior used as the central object of (6). The denominator plays the role of a marginal likelihood and is referred to in the main text as the maxitive marginal likelihood $Z_{\max}$. For a likelihood normalised by $\sup_\theta p(x \mid \theta)$, the corresponding consistency degree is $Z_{\max} / \sup_\theta p(x \mid \theta)$, rather than an averaged model fit.

Uninformative priors. The function $\pi \equiv \mathbf{1}$ is itself a valid possibility function and represents the complete absence of prior information, in the sense that every value of θ is deemed fully plausible. More generally, any prior of the form $\pi(\theta) = \exp(-R(\theta))$ with $R \geq 0$ is automatically a possibility function once renormalised by $\sup_\theta \exp(-R(\theta))$, regardless of whether $\int \exp(-R(\theta))\, \mathrm{d}\theta$ is finite. This contrasts with the probabilistic setting, where standard regularizers such as a constant or the indicator of an unbounded set fail to integrate and yield improper priors.

Connection to imprecise probability. A possibility function f can also be interpreted as an upper envelope over a family of probability distributions, namely the credal set $\{P : P(A) \leq \sup_{\theta\in A} f(\theta)$ for all measurable $A\}$. This perspective is not developed here, and we refer the interested reader to Hieu et al. (2025) for further discussion.

D Proofs

In this section, we present the proofs for the main theoretical results in the main text.

D.1 Proof of Theorem 2

We prove each equation of the theorem one after the other, with the notation $h = -\ell$ for conciseness.

Proof of Equation 8a. Consider some fixed possibility function g.

Let us first assume that there exists some $\theta_g^\star \in \operatorname{argsup}_{\theta\in\Theta} g(\theta)$. We then have $g(\theta_g^\star) = 1$, so that:

$$\log \sup_{\theta\in\Theta} e^{h(\theta)}\pi(\theta) \geq \log e^{h(\theta_g^\star)}\pi(\theta_g^\star) \tag{38a}$$

$$= \log \frac{e^{h(\theta_g^\star)}\pi(\theta_g^\star)}{g(\theta_g^\star)} \tag{38b}$$

$$\geq \inf_{\theta\in\Theta} \log \frac{e^{h(\theta)}\pi(\theta)}{g(\theta)} \tag{38c}$$

$$= \inf_{\theta\in\Theta} \left\{ h(\theta) - \log \frac{g(\theta)}{\pi(\theta)} \right\}. \tag{38d}$$

Now, if $\operatorname{argsup}_{\theta\in\Theta} g(\theta) = \varnothing$, then one can still define a sequence $(\theta_{g,n}^\star)_{n=2}^\infty$ in Θ such that for any integer $n \geq 2$, $g(\theta_{g,n}^\star) \geq 1 - 1/n$. We can then write:

$$\log \sup_{\theta\in\Theta} e^{h(\theta)}\pi(\theta) \geq \log e^{h(\theta_{g,n}^\star)}\pi(\theta_{g,n}^\star) \tag{39a}$$

$$= \log \frac{e^{h(\theta_{g,n}^\star)}\pi(\theta_{g,n}^\star)}{1-\frac{1}{n}} + \log\left(1-\frac{1}{n}\right) \tag{39b}$$

$$\geq \log \frac{e^{h(\theta_{g,n}^\star)}\pi(\theta_{g,n}^\star)}{g(\theta_{g,n}^\star)} + \log\left(1-\frac{1}{n}\right) \tag{39c}$$

$$\geq \inf_{\theta\in\Theta} \log \frac{e^{h(\theta)}\pi(\theta)}{g(\theta)} + \log\left(1-\frac{1}{n}\right) \tag{39d}$$

$$= \inf_{\theta\in\Theta} \left\{ h(\theta) - \log \frac{g(\theta)}{\pi(\theta)} \right\} + \log\left(1-\frac{1}{n}\right), \tag{39e}$$

so by letting $n \to +\infty$, we have

$$\log \sup_{\theta\in\Theta} e^{h(\theta)}\pi(\theta) \geq \inf_{\theta\in\Theta} \left\{ h(\theta) - \log \frac{g(\theta)}{\pi(\theta)} \right\}. \tag{40}$$

Consequently, the inequality above holds for any possibility function $g \in \mathcal{F}(\Theta)$, and by taking the supremum over $g \in \mathcal{F}(\Theta)$ in the right-hand side leads to:

$$\log \sup_{\theta\in\Theta} e^{h(\theta)}\pi(\theta) \geq \sup_{g\in\mathcal{F}(\Theta)} \inf_{\theta\in\Theta} \left\{ h(\theta) - \log \frac{g(\theta)}{\pi(\theta)} \right\}. \tag{41}$$

Furthermore, the choice of possibility function $g(\theta) = g_{\max}^\star(\theta) = e^{h(\theta)}\pi(\theta)/\sup_{\theta'} e^{h(\theta')}\pi(\theta')$ transforms the inequality into an equality, which finally gives Equation 8a:

$$\log \sup_{\theta\in\Theta} e^{h(\theta)}\pi(\theta) = \sup_{g\in\mathcal{F}(\Theta)} \inf_{\theta\in\Theta} \left\{ h(\theta) - \log \frac{g(\theta)}{\pi(\theta)} \right\}. \tag{42}$$

□

Proof of Equation 8b. Any possibility function g takes its values in $[0, 1]$, so that for any possibility function g and any parameter $\theta \in \Theta$, we have $\log g(\theta) \leq 0$. Thus, for any possibility function g and any parameter θ,

$$\log e^{h(\theta)}\pi(\theta) \leq h(\theta) - \log \frac{g(\theta)}{\pi(\theta)}, \tag{43}$$

so taking the supremum over θ in both sides leads for any g to:

$$\log \sup_{\theta \in \Theta} e^{h(\theta)} \pi(\theta) \leq \sup_{\theta \in \Theta} \left\{ h(\theta) - \log \frac{g(\theta)}{\pi(\theta)} \right\}, \tag{44}$$

and taking the infimum over possibility functions g in the right-hand side leads to

$$\log \sup_{\theta \in \Theta} e^{h(\theta)} \pi(\theta) \leq \inf_{g \in \mathcal{F}(\Theta)} \sup_{\theta \in \Theta} \left\{ h(\theta) - \log \frac{g(\theta)}{\pi(\theta)} \right\}, \tag{45}$$

Once again, $g = g^\star_{\max}$ transforms the inequality into an equality, which finally gives Equation 8b

$$\log \sup_{\theta \in \Theta} e^{h(\theta)} \pi(\theta) = \inf_{g \in \mathcal{F}(\Theta)} \sup_{\theta \in \Theta} \left\{ h(\theta) - \log \frac{g(\theta)}{\pi(\theta)} \right\}. \tag{46}$$

□

Proof of Equation 9. We have already mentioned in the proof of Equation 8a that

$$g^\star_{\max}(\theta) \doteq \frac{e^{h(\theta)} \pi(\theta)}{\sup_{\theta' \in \Theta} e^{h(\theta')} \pi(\theta')} \in \operatorname*{argmax}_{g \in \mathcal{F}(\Theta)} \inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g(\theta)}{\pi(\theta)} \right) \right\}, \tag{47}$$

so that

$$\inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g^\star_{\max}(\theta)}{\pi(\theta)} \right) \right\} = \sup_{g \in \mathcal{F}(\Theta)} \inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g(\theta)}{\pi(\theta)} \right) \right\}. \tag{48}$$

Furthermore, any possibility function $g' \in \mathcal{F}(\Theta)$ such that $g' \preceq g^\star_{\max}$ satisfies by monotonicity of the logarithmic function:

$$\inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g'(\theta)}{\pi(\theta)} \right) \right\} \geq \inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g^\star_{\max}(\theta)}{\pi(\theta)} \right) \right\}. \tag{49}$$

Hence, combining the two lines above, we get for any possibility function $g' \in \mathcal{F}(\Theta)$ such that $g' \preceq g^\star_{\max}$:

$$\inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g'(\theta)}{\pi(\theta)} \right) \right\} \geq \sup_{g \in \mathcal{F}(\Theta)} \inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g(\theta)}{\pi(\theta)} \right) \right\}. \tag{50}$$

Since by definition of the supremum, we have the reverse inequality:

$$\inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g'(\theta)}{\pi(\theta)} \right) \right\} \leq \sup_{g \in \mathcal{F}(\Theta)} \inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g(\theta)}{\pi(\theta)} \right) \right\}, \tag{51}$$

we finally have for any possibility function $g' \in \mathcal{F}(\Theta)$ such that $g' \preceq g^\star_{\max}$:

$$\inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g'(\theta)}{\pi(\theta)} \right) \right\} = \sup_{g \in \mathcal{F}(\Theta)} \inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g(\theta)}{\pi(\theta)} \right) \right\}. \tag{52}$$

This provides half of the proof, namely

$$\{ g \in \mathcal{F}(\Theta) : g \preceq g^\star_{\max} \} \subset \operatorname*{argmax}_{g \in \mathcal{F}(\Theta)} \inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g(\theta)}{\pi(\theta)} \right) \right\}. \tag{53}$$

To get the equality, we still have to show that any possibility function g not satisfying $g \preceq g^\star_{\max}$ does not belong to the argmax. To show this, let us consider some possibility function $g' \notin \{ g \in \mathcal{F}(\Theta) : g \preceq g^\star_{\max} \}$, and show that

$$\inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g'(\theta)}{\pi(\theta)} \right) \right\} < \sup_{g \in \mathcal{F}(\Theta)} \inf_{\theta \in \Theta} \left\{ h(\theta) - \log \left(\frac{g(\theta)}{\pi(\theta)} \right) \right\}. \tag{54}$$

To see this, recall that since $g' \not\preceq g^\star_{\max}$, there exists at least one $\theta' \in \Theta$ such that $g'(\theta') > g^\star_{\max}(\theta')$. Therefore,

$$h(\theta') - \log \left(\frac{g'(\theta')}{\pi(\theta')} \right) < h(\theta') - \log \left(\frac{g^\star_{\max}(\theta')}{\pi(\theta')} \right), \tag{55}$$

Using the definition of the infimum, we have

$$\inf_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g'(\theta)}{\pi(\theta)}\right)\right\}\leq h(\theta')-\log\left(\frac{g'(\theta')}{\pi(\theta')}\right)<h(\theta')-\log\left(\frac{g^\star_{\max}(\theta')}{\pi(\theta')}\right). \tag{56}$$

However, notice that by definition of $g^\star_{\max}$, the quantity in the right-hand side above does not depend on θ' since:

$$h(\theta')-\log\left(\frac{g^\star_{\max}(\theta')}{\pi(\theta')}\right)=h(\theta')-\log\left(\frac{\frac{e^{h(\theta')}\pi(\theta')}{\sup e^{h(\cdot)}\pi(\cdot)}}{\pi(\theta')}\right)=\log\sup_{\theta\in\Theta}e^{h(\theta)}\pi(\theta), \tag{57}$$

so combining the two lines above, we get

$$\inf_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g'(\theta)}{\pi(\theta)}\right)\right\}<\log\sup_{\theta\in\Theta}e^{h(\theta)}\pi(\theta), \tag{58}$$

By using Equation 8a, we can rewrite the quantity in the right-hand side:

$$\inf_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g'(\theta)}{\pi(\theta)}\right)\right\}<\sup_{g\in\mathcal{F}(\Theta)}\inf_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\}, \tag{59}$$

which is exactly what we wanted to show. Hence, g' cannot belong to the set of maximisers. This proves that

$$\operatorname*{argmax}_{g\in\mathcal{F}(\Theta)}\inf_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\}=\left\{g\in\mathcal{F}(\Theta):g\preceq g^\star_{\max}\right\}, \tag{60}$$

which concludes the proof. □

Proof of Equation 10. The proof of Equation 10 is very similar to the proof of Equation 9, and is only provided for the sake of completeness. Once again, we start from the following fact mentioned in the proof (8b):

$$g^\star_{\max}(\theta)\doteq\frac{e^{h(\theta)}\pi(\theta)}{\sup_{\theta'\in\Theta}e^{h(\theta')}\pi(\theta')}\in\operatorname*{argmin}_{g\in\mathcal{F}(\Theta)}\sup_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\}, \tag{61}$$

so that

$$\sup_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g^\star_{\max}(\theta)}{\pi(\theta)}\right)\right\}=\inf_{g\in\mathcal{F}(\Theta)}\sup_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\}. \tag{62}$$

Furthermore, any possibility function $g'\in\mathcal{F}(\Theta)$ such that $g^\star_{\max}\preceq g'$ satisfies by monotonicity of the logarithmic function:

$$\sup_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g'(\theta)}{\pi(\theta)}\right)\right\}\leq\sup_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g^\star_{\max}(\theta)}{\pi(\theta)}\right)\right\}. \tag{63}$$

Hence, combining the two lines above, we get for any possibility function $g'\in\mathcal{F}(\Theta)$ such that $g^\star_{\max}\preceq g'$:

$$\sup_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g'(\theta)}{\pi(\theta)}\right)\right\}\leq\inf_{g\in\mathcal{F}(\Theta)}\sup_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\}. \tag{64}$$

Since by definition of the infimum, we have the reverse inequality:

$$\sup_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g'(\theta)}{\pi(\theta)}\right)\right\}\geq\inf_{g\in\mathcal{F}(\Theta)}\sup_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\}, \tag{65}$$

we finally have for any possibility function $g'\in\mathcal{F}(\Theta)$ such that $g^\star_{\max}\preceq g'$:

$$\sup_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g'(\theta)}{\pi(\theta)}\right)\right\}=\inf_{g\in\mathcal{F}(\Theta)}\sup_{\theta\in\Theta}\left\{h(\theta)-\log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\}. \tag{66}$$

This provides half of the proof, namely

$$\{g \in \mathcal{F}(\Theta) : g^\star_{\max} \preceq g\} \subset \operatorname*{argmin}_{g\in\mathcal{F}(\Theta)} \sup_{\theta\in\Theta} \left\{ h(\theta) - \log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\}. \tag{67}$$

To get the equality, we now have to show that any possibility function g not satisfying $g^\star_{\max} \preceq g$ does not belong to the argmin. To show this, let us consider some possibility function $g' \notin \{g \in \mathcal{F}(\Theta) : g^\star_{\max} \preceq g\}$, and show that

$$\sup_{\theta\in\Theta} \left\{ h(\theta) - \log\left(\frac{g'(\theta)}{\pi(\theta)}\right)\right\} > \inf_{g\in\mathcal{F}(\Theta)} \sup_{\theta\in\Theta} \left\{ h(\theta) - \log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\}. \tag{68}$$

To see this, recall that since $g^\star_{\max} \not\preceq g'$, there exists at least one $\theta' \in \Theta$ such that $g'(\theta') < g^\star_{\max}(\theta')$. Therefore,

$$h(\theta') - \log\left(\frac{g'(\theta')}{\pi(\theta')}\right) > h(\theta') - \log\left(\frac{g^\star_{\max}(\theta')}{\pi(\theta')}\right), \tag{69}$$

Using the definition of the supremum, we now have

$$\sup_{\theta\in\Theta} \left\{ h(\theta) - \log\left(\frac{g'(\theta)}{\pi(\theta)}\right)\right\} \geq h(\theta') - \log\left(\frac{g'(\theta')}{\pi(\theta')}\right) > h(\theta') - \log\left(\frac{g^\star_{\max}(\theta')}{\pi(\theta')}\right). \tag{70}$$

Again, the quantity in the right-hand side above does not depend on θ':

$$h(\theta') - \log\left(\frac{g^\star_{\max}(\theta')}{\pi(\theta')}\right) = h(\theta') - \log\left(\frac{\dfrac{e^{h(\theta')}\pi(\theta')}{\sup e^{h(\cdot)}\pi(\cdot)}}{\pi(\theta')}\right) = \log\sup_{\theta\in\Theta} e^{h(\theta)}\pi(\theta), \tag{71}$$

so combining the two lines above leads to

$$\sup_{\theta\in\Theta} \left\{ h(\theta) - \log\left(\frac{g'(\theta)}{\pi(\theta)}\right)\right\} > \log\sup_{\theta\in\Theta} e^{h(\theta)}\pi(\theta), \tag{72}$$

Now using Equation 8b, we can rewrite the quantity in the right-hand side:

$$\sup_{\theta\in\Theta} \left\{ h(\theta) - \log\left(\frac{g'(\theta)}{\pi(\theta)}\right)\right\} > \inf_{g\in\mathcal{F}(\Theta)} \sup_{\theta\in\Theta} \left\{ h(\theta) - \log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\}, \tag{73}$$

which is precisely what we wanted to show. Hence, g' cannot belong to the set of minimisers. This proves that

$$\operatorname*{argmin}_{g\in\mathcal{F}(\Theta)} \sup_{\theta\in\Theta} \left\{ h(\theta) - \log\left(\frac{g(\theta)}{\pi(\theta)}\right)\right\} = \{g \in \mathcal{F}(\Theta) : g^\star_{\max} \preceq g\}, \tag{74}$$

which concludes the proof. □

D.2 Proofs of Results in Section 5

Proof of Proposition 1. Given the probabilistic likelihood $p(x\,|\,\theta) = \exp(\theta^\top T(x) - A(\theta) - B(x))$, under the uninformative prior on Θ, $\pi(\theta) = 1$, the posterior possibility is given by

$$\pi(\theta\,|\,x) = \frac{p(x\,|\,\theta)\pi(\theta)}{\max_{\theta'} p(x\,|\,\theta')\pi(\theta')} = \frac{p(x\,|\,\theta)}{\max_{\theta'} p(x\,|\,\theta')} \tag{75a}$$

$$= \exp\left((\theta-\theta^\star)^\top T(x) - (A(\theta) - A(\theta^\star))\right), \tag{75b}$$

where we denote the MLE by $\theta^\star \doteq \operatorname{argmax}_\theta p(x\,|\,\theta) = \operatorname{argmax}_\theta \theta^\top T(x) - A(\theta)$. The first-order condition for the maximum tells us that $T(x) = \nabla_\theta A(\theta^\star)$. Plugging this in the equation above, we get the result in (22b):

$$\pi(\theta\,|\,x) = \exp\left((\theta-\theta^\star)^\top \nabla_\theta A(\theta^\star) - (A(\theta) - A(\theta^\star))\right) \tag{76a}$$

$$= \exp\left(-D_A(\theta\|\theta^\star)\right). \tag{76b}$$

Next, we get the result in (22a) by rewriting the posterior as

$$\pi(\theta \mid x) = \exp\left(\theta^\top T(x) - A(\theta) - \left(\theta^{\star\top} T(x) - A(\theta^\star)\right)\right) \tag{77a}$$

$$= \exp\left(\theta^\top \lambda - A(\theta) - A^\dagger(\lambda)\right), \tag{77b}$$

where $\lambda \doteq T(x)$ and $A^\dagger = \max_\theta \theta^\top T(x) - A(\theta) = \theta^{\star\top} T(x) - A(\theta^\star)$ is the Legendre transform of A. The resulting posterior is indeed of the same form as g_λ in (20). □

Proof of Proposition 2. First, we show that $f_\theta : \lambda \to g_\lambda(\theta)$ is in $\mathcal{G}_{A^\dagger}(\Lambda) \doteq \left\{\lambda \mapsto \exp(\theta^\top \lambda - A^{\dagger\dagger}(\theta) - A^\dagger(\lambda)) : \theta \in \Theta\right\}$. Since $A^{\dagger\dagger} = A$ (assuming A is proper, closed, and convex), we have

$$g_\lambda(\theta) = \exp\left(\lambda^\top \theta - A^\dagger(\lambda) - A(\theta)\right) \tag{78a}$$

$$= \exp\left(\theta^\top \lambda - A^{\dagger\dagger}(\theta) - A^\dagger(\lambda)\right). \tag{78b}$$

Therefore, the mapping $f_\theta : \lambda \to g_\lambda(\theta)$, which can be written as

$$f_\theta : \lambda \to \exp\left(\theta^\top \lambda - A^{\dagger\dagger}(\theta) - A^\dagger(\lambda)\right), \tag{79}$$

is in $\mathcal{G}_{A^\dagger}(\Lambda)$.

For $\nu > 0$, we show that $g_\lambda^\nu \in \mathcal{G}_{\nu A}(\Theta)$; the case $\nu = 0$ gives $g_\lambda^0 = \mathbf{1}$. It holds that

$$g_\lambda^\nu(\theta) = \exp\left(\nu\lambda^\top \theta - \nu A^\dagger(\lambda) - \nu A(\theta)\right) \tag{80a}$$

$$= \exp\left(\nu\lambda^\top \theta - (\nu A)^\dagger(\nu\lambda) - \nu A(\theta)\right), \tag{80b}$$

where the second equality follows from the following property of the convex conjugate:

$$\nu A^\dagger(\lambda) = \nu \sup_\theta \lambda^\top \theta - A(\theta) = \sup_\theta \nu\lambda^\top \theta - \nu A(\theta) = (\nu A)^\dagger(\nu\lambda). \tag{81}$$

We conclude that $g_\lambda^\nu \in \mathcal{G}_{\nu A}(\Theta)$, as required.

Finally, consider the likelihood $p(x \mid \theta) = \exp(\theta^\top T(x) - A(\theta) - B(x))$, and a choice of prior $g_{\lambda,\nu} \doteq g_\lambda^\nu$, with $g_\lambda \in \mathcal{G}_A(\Theta)$ defined as $g_\lambda(\theta) = \exp\left(\lambda^\top \theta - A^\dagger(\lambda) - A(\theta)\right)$, for some $\lambda \in \Lambda$. For x such that $(T(x) + \nu\lambda)/(\nu + 1) \in \Lambda$, the posterior possibility is given by

$$g_{\lambda,\nu}(\theta \mid x) = \frac{p(x \mid \theta) g_\lambda^\nu(\theta)}{\max_{\theta'} p(x \mid \theta') g_\lambda^\nu(\theta')} \tag{82a}$$

$$= \frac{\exp\left(\theta^\top (T(x) + \nu\lambda) - (\nu+1) A(\theta)\right)}{\max_{\theta'} \exp\left(\theta'^\top (T(x) + \nu\lambda) - (\nu+1) A(\theta')\right)} \tag{82b}$$

$$= \exp\left(\theta^\top (T(x) + \nu\lambda) - (\nu+1) A(\theta) - (\nu+1) \max_{\theta'} \left(\theta'^\top \left(\frac{T(x) + \nu\lambda}{\nu+1}\right) - A(\theta')\right)\right) \tag{82c}$$

$$= \exp\left(\theta^\top (T(x) + \nu\lambda) - (\nu+1) A(\theta) - (\nu+1) A^\dagger\left(\frac{T(x) + \nu\lambda}{\nu+1}\right)\right) \tag{82d}$$

$$= g_{\frac{T(x)+\nu\lambda}{\nu+1}, \nu+1}(\theta). \tag{82e}$$

Therefore, $g_{\lambda,\nu}$ is a valid conjugate prior for the likelihood $p(\cdot \mid \theta)$, where the hyperparameters are updated as

$$\lambda' = \frac{\nu\lambda + T(x)}{\nu + 1}, \quad \nu' = \nu + 1. \tag{83}$$

As a sanity check, note that with $\nu = 0$ and $T(x) \in \Lambda$, g is the uninformative prior, and we recover the posterior in Proposition 1.

For n i.i.d. observations $\{x_i\}_{i=1}^n$, the likelihood product is $\prod_{i=1}^n p(x_i \mid \theta) \propto \exp\left(\theta^\top \sum_{i=1}^n T(x_i) - nA(\theta)\right)$. In this case, following the same derivation as above, the posterior possibility is given by

$$g_{\lambda,\nu}(\theta \mid x_{1:n}) = g_{\frac{\sum_{i=1}^n T(x_i) + \nu\lambda}{\nu+n}, \nu+n}(\theta), \tag{84}$$

where the hyperparameters are now updated as

$$\lambda' = \frac{\nu\lambda + \sum_{i=1}^{n} T(x_i)}{\nu + n}, \quad \nu' = \nu + n. \tag{85}$$

□

Proof of Proposition 3. Recall that $\mathrm{CBO}_s(\eta) \doteq \mathrm{ext}_{\theta\in\Theta} -\hat{\ell}(\theta) - \log g_\eta(\theta)$, with ext = sup for $s = +1$ and ext = inf for $s = -1$, where the candidate is given by

$$g_\eta(\theta) = \exp\left(-D_A(\theta\|\eta)\right) = \exp\left(\nabla A(\eta)^\top(\theta-\eta) + A(\eta) - A(\theta)\right) \tag{86}$$

$$\implies -\hat{\ell}(\theta) - \log g_\eta(\theta) = -\hat{\ell}(\theta) - \nabla A(\eta)^\top(\theta-\eta) - A(\eta) + A(\theta) \tag{87}$$

Under the assumed finiteness, uniqueness and regularity conditions, Danskin's theorem gives us $\nabla\mathrm{CBO}_s(\eta) = -\nabla^2 A(\eta)\,(\theta_s - \eta)$, where $\theta_s = \mathrm{argext}_{\theta\in\Theta} -\hat{\ell}(\theta) - \log g_\eta(\theta)$. The 2nd-order Taylor approximation of the objective around η is given by

$$-\hat{\ell}(\theta) - \log g_\eta(\theta) \approx -\hat{\ell}(\eta) - (\theta-\eta)^\top \nabla\hat{\ell}(\eta) - \frac{1}{2}(\theta-\eta)^\top\left(\nabla^2\hat{\ell}(\eta) - \nabla^2 A(\eta)\right)(\theta-\eta) \tag{88}$$

Under this approximation, assuming $H_s(\eta) = s(\nabla^2\hat{\ell}(\eta) - \nabla^2 A(\eta)) \succ 0$, the unconstrained quadratic inner problem has the finite extremum

$$\theta_s \approx \eta - sH_s(\eta)^{-1}\nabla\hat{\ell}\,(\eta) \tag{89}$$

Substituting this approximate inner optimizer into the natural-gradient step for minimizing $s\,\mathrm{CBO}_s(\eta)$ yields a Newton-like update rule:

$$\eta_{t+1} = \eta_t - \rho_t s\left(\nabla^2 A\,(\eta_t)\right)^{-1}\nabla\mathrm{CBO}_s(\eta_t) \tag{90a}$$

$$\approx \eta_t - \rho_t H_s(\eta_t)^{-1}\nabla\hat{\ell}\,(\eta_t) \tag{90b}$$

where $\rho_t > 0$ is the step size. □

E Discussion on Max-Relative Entropy

E.1 Relationship with Generalized VI

In classical VI, the optimization is typically framed through the decomposition of the ELBO, which involves the reverse KL divergence, $\mathrm{KL}(q\|q^\star_{\mathrm{add}})$. This yields a lower bound on the model evidence and has well-known properties such as the mode-seeking effect. Yet, in the probabilistic literature, many alternative divergences and bound constructions have been explored, leading to a variety of upper and lower bounds on the marginal likelihood. Notable examples include the χ-divergence upper bounds (Dieng et al., 2017) and variational Rényi bounds (Li and Turner, 2016). In each of these cases, one can define dual optimization problems – minimisation to the left or to the right of the divergence – corresponding to lower or upper bounds providing a sandwiching of the true model evidence.

Remarkably, in the possibilistic framework, an analogous structure arises naturally, but with the max-relative entropy replacing the KL divergence. The lower and upper consistency bounds (CBOs) correspond to two dual optimization perspectives: maximizing the lower CBO or minimizing the upper CBO, whose unrestricted optimizers respectively lower- and upper-bound the posterior possibility function; within a restricted family, exact bounds require a zero approximation gap. This is conceptually analogous to probabilistic VI, where reverse or forward KL leads to mode-seeking versus mass-covering behaviours.

E.2 Intuition Through Visualization

Recall the definition of max-relative entropy in Equation 13: $D_{\max}(g\,\|\,f) \doteq \sup_{\theta\in\Theta}\log\frac{g(\theta)}{f(\theta)}$. When attained, the supremum is attained at points where $g \geq f$, and $D_{\max}(g\,\|\,f) = 0$ if and only if g lower bounds f. To illustrate, we present two sinusoidal possibility functions in Figure 1a, and visualize the *active region* for computing $D_{\max}(g\,\|\,f)$, which is marked in darker shades.

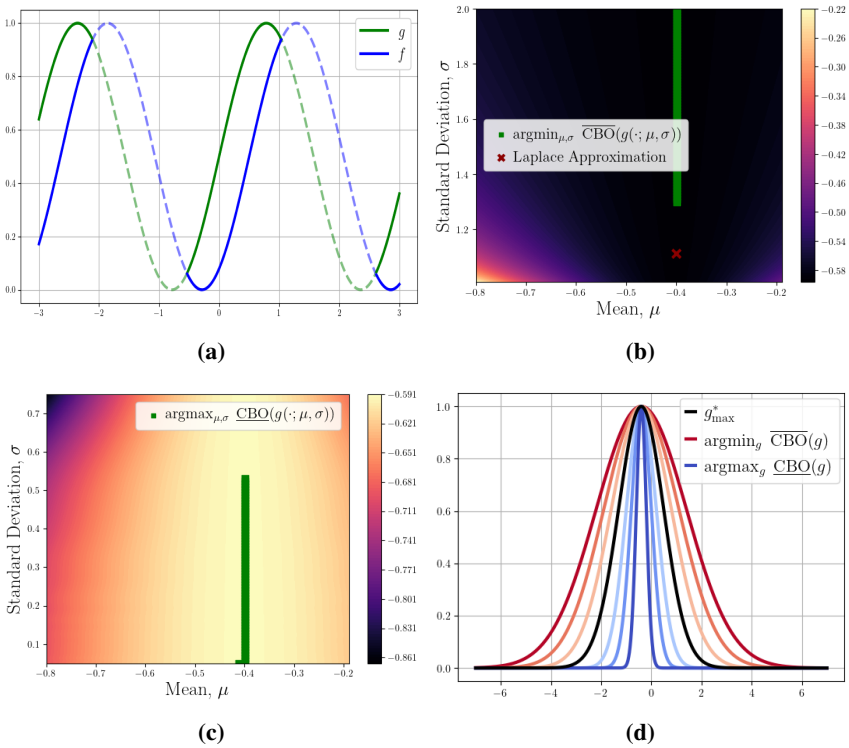


Figure 1: Building an intuition for Maxitive-VI. *Top Left:* Solid lines depict active regions in the supremum computation of $D_{\max}(g \,\|\, f)$ for two sinusoidal possibility functions; dashed lines depict inactive regions. *Top Right:* Minimizers of $\overline{\mathrm{CBO}}$ for the Bayesian LR problem described in Appendix E.3, along with the Laplace approximation, showing consistency in the mean computation while allowing for varying precision levels. *Bottom Left:* Maximizers of $\underline{\mathrm{CBO}}$. *Bottom Right:* Numerically computed maxitive posterior, $g^\star_{\max}$, and the CBO optimizers, showing how they bound $g^\star_{\max}$ (from either side, depending on the formulation).

Looking at Equation 12, we note that maximizing $\underline{\mathrm{CBO}}\,(g)$ amounts to minimizing $D_{\max}(g \,\|\, g^\star_{\max})$. Therefore, the set of maximizers consists of possibility functions lower bounding the true posterior, $g^\star_{\max}$. Similarly, minimizing $\overline{\mathrm{CBO}}\,(g)$ amounts to minimizing $D_{\max}(g^\star_{\max} \,\|\, g)$, which is achieved by possibility functions upper bounding the true posterior. This is visualized in Figure 1d for Gaussian candidates; see next subsection for a description of the inference problem.

E.3 Example: Bayesian Logistic Regression

We perform Bayesian logistic regression for a simple problem to visualize what the consistency bounds look like for different candidate possibility functions, and how the true posterior compares with the optimizers of upper/lower CBO.

Consider a binary random variable $\mathbf{y} \in \{-1, +1\}$ with $p(1 \mid f) = \sigma(f)$, where σ is the sigmoid function, $\sigma(x) = (1+\exp(-x))^{-1}$. We assume that $\boldsymbol{f}$ is a deterministic uncertain variable, and put a Gaussian possibilistic prior on it, i.e. $\boldsymbol{f}$ is described by $\overline{\mathrm{N}}(0,1)$. The maxitive posterior is then given by

$$g^\star_{\max}(f) = \frac{p(y|f)\pi(f)}{\sup_{f'} p(y|f')\pi(f')} = \frac{\sigma(yf)\exp\left(-\frac{1}{2}f^2\right)}{\sup_{f'} \sigma(yf')\exp\left(-\frac{1}{2}f'^2\right)} \tag{91}$$

The denominator is not analytically tractable, hence, we perform VI with the family of Gaussian possibility functions, $\mathcal{G} = \{\overline{\mathrm{N}}(\mu, \sigma^2) : \mu \in \mathbb{R}, \sigma \in \mathbb{R}^+\}$.

Say $y = -1$. In Figure 1b, we show the minimizers of $\overline{\mathrm{CBO}}$, along with the solution computed using Laplace approximation, i.e. values of μ and σ that match the first and second (possibilistic) moments

of $g^{\star}_{\max}$. We note that the minimizers match the mean of the Laplace approximation, while allowing arbitrarily large values for the variance of the approximate posterior, each of which yields a model consistent with the data. Similarly, in Figure 1c, we show the maximizers of $\underline{\mathrm{CBO}}$, which also match the mean, while allowing for the solution to be arbitrarily precise (lower σ). Finally, in Figure 1d, we show the true maxitive posterior, computed numerically, as well as the optimizers computed using the VI formulation, showing how the minimizers of $\overline{\mathrm{CBO}}$ upper bound the true posterior, while the maximizers of $\underline{\mathrm{CBO}}$ lower bound it.

F Possibilistic Exponential Family

We start by clarifying some properties of possibilistic exponential families:

Property 1: There is no general correspondence between the components T_+ and B_+ of a probabilistic (canonical) exponential family and their analogue in possibility theory.
Property 2: Although possibility functions are not densities and, therefore, do not require the definition of a reference measure, the carrier function $\exp(-B)$ of a possibilistic exponential family need not be an indicator function for the support of the family.
Property 3: The log-partition function A is simpler when compared to the expression of the log-partition A_+ for a probabilistic exponential family, that is

$$A_+(\lambda) = \log \int \exp\big(\lambda^\intercal T_+(\theta) - B_+(\theta)\big)\mathrm{d}\theta, \tag{92}$$

since $\sup$ and $\log$ can be exchanged when defining A.

These properties are illustrated in the following examples.

Example 1. Consider a Bernoulli-style possibility function: set $\Theta = \{0, 1\}$ and define the possibility of $\boldsymbol{\theta} = 0$ as α_0 and the possibility of $\boldsymbol{\theta} = 1$ as α_1, with $\alpha_0, \alpha_1 > 0$ and $\max\{\alpha_0, \alpha_1\} = 1$ by construction. The corresponding possibility function is $f(\theta \,|\, \alpha_0, \alpha_1) = \alpha_0^{1-\theta}\alpha_1^{\theta}$. This can be expressed in an exponential family form as

$$g_\lambda(\theta) = \exp\big(\lambda^\intercal T(\theta) - A(\lambda)\big), \tag{93}$$

with $T(\theta) = (1-\theta, \theta)$, $\lambda = (\lambda_1, \lambda_2) \in \Lambda = \mathbb{R}^2$ and

$$A(\lambda) = \sup_{\theta\in\{0,1\}} \lambda^\intercal T(\theta) = \max\{\lambda_1, \lambda_2\}. \tag{94}$$

These components differ significantly from those of a Bernoulli distribution. Although the parameters α_0 and α_1 are not directly related, the fact that their maximum is equal to 1 allows us to express this exponential family with a single variable: let $\eta = \lambda_2 - \lambda_1$, then another possible exponential-family form for this Bernoulli-style possibility function is

$$g_\eta(\theta) = \exp(\eta\theta - A(\eta)), \tag{95}$$

with $A(\eta) = \max\{0, \eta\}$. Despite its simplicity, the Bernoulli possibility function is useful to express, for instance, possibilities of detection in tracking problems, see e.g. Ristic et al. (2020).

Example 2. Consider a Poisson-style possibility function: set $\Theta = \mathbb{N}_0$ and consider a parameter $\alpha \in \mathbb{R}^+$. The corresponding possibility function is

$$f(\theta \,|\, \alpha) = \frac{\alpha^{\theta - \lfloor\alpha\rfloor}\lfloor\alpha\rfloor!}{\theta!}, \tag{96}$$

which can be expressed in an exponential family form as

$$g_\lambda(\theta) = \exp\big(\lambda^\intercal\theta - A(\lambda) - B(\theta)\big). \tag{97}$$

with $\lambda = \log\alpha$, $B(\theta) = \log(\theta!)$, and $A(\lambda) = \lfloor e^\lambda\rfloor\lambda - \log\big(\lfloor e^\lambda\rfloor!\big)$.

Example 3. Consider the univariate normal possibility function with known variance σ^2, $\overline{\mathrm{N}}(\theta; \mu, \sigma^2)$, for which we have $B(\theta) = \theta^2/(2\sigma^2)$, whereas the negative log carrier function of the corresponding univariate normal distribution is

$$\frac{\theta^2}{2\sigma^2} + \frac{1}{2}\log(2\pi\sigma^2). \tag{98}$$

Example 4. Consider the case where $\Theta = [0,1]$, $B(\theta) = 0$ and $T(\theta) = (T_1(\theta), T_2(\theta))$, with $T_1(\theta) = \theta$ and $T_2(\theta) = \mathbf{1}_0(\theta)$, i.e., the indicator of the point 0. The corresponding possibilistic exponential family is of the form

$$g_\lambda(\theta) = \exp(\lambda_1\theta + \lambda_2\mathbf{1}_0(\theta) - \max\{0, \lambda_1, \lambda_2\}), \tag{99}$$

which takes the component T_2 into account. If we considered the analogous probabilistic exponential family, the obtained density would be almost everywhere equal to

$$q_{\lambda_1}(\theta) \propto \exp\left(\lambda_1\theta - \log \int \exp(\lambda_1\theta)\mathrm{d}\theta\right), \tag{100}$$

since the term $\mathbf{1}_0(\theta)$ only affects a single point (of measure zero). This highlights how minimal representations of exponential families can differ between the probabilistic and possibilistic case, due to the respective behaviours of integration and optimization.

G Algorithms

In the main paper, we present the algorithm for `uCBOpt` (Alg. 1). Here, we present the algorithms for `uCBOpt-adapt` and `lCBOpt-adapt` in Algorithms 2 and 3, respectively (main differences are highlighted in blue). Both methods track a data-plus-prior curvature estimate $\tilde{h}_t = h_t + \delta$, where h_t is an exponential moving average of the squared stochastic gradients and δ is the weight-decay/prior curvature. The variable c_t then forms a one-sided candidate curvature envelope of $\tilde{h}_t$: `uCBOpt-adapt` uses a growing running minimum, while `lCBOpt-adapt` uses a decayed running maximum. The coordinatewise candidate precision $S_t = \mathrm{diag}(\gamma c_t)$ is held fixed when computing each local mean update.

In `uCBOpt-adapt`, c_t tracks the data-plus-prior curvature through $c_t = \min\{\beta_3 c_{t-1}, \tilde{h}_t\}$ with $\beta_3 > 1$. This behaves like a running minimum that can gradually increase over time when the observed curvature grows. The update divides by the residual curvature $\tilde{h}_t - \gamma c_t$, which can be interpreted as the part of the data-plus-prior sharpness that is not accounted for by the candidate curvature γc_t.

In `lCBOpt-adapt`, c_t is instead updated as $c_t = \max\{\beta_3 c_{t-1}, \tilde{h}_t\}$ with $\beta_3 \in [0,1)$. This behaves like a running maximum that can gradually decay over time when the observed curvature decreases. The update divides by $\gamma c_t - \tilde{h}_t$, with $\gamma > 1$, which measures the gap between the scaled upper candidate curvature and the current data-plus-prior curvature estimate. Thus, the two adaptive variants differ mainly in whether the curvature correction is constructed from a lower or upper envelope of $\tilde{h}_t$.

Below we provide some guidelines for choosing the hyperparameters in `uCBOpt`, `uCBOpt-adapt` and `lCBOpt-adapt`:

Learning rate schedule ρ_t. We found that the learning-rate schedule used for SGD and AdamW is also a good starting point for the `CBOpt` family. In our classification experiments with `CBOpt`, we use a short linear warmup followed by cosine annealing to zero. For `uCBOpt`, learning rates comparable to SGD can work well on larger datasets, while smaller learning rates are generally safer for the adaptive variants `uCBOpt-adapt` and `lCBOpt-adapt`.

Weight decay δ. The weight decay can be chosen similarly to SGD or AdamW, since it plays the role of a prior curvature term in the data-plus-prior curvature estimate. Weight-decay values that work well for IVON also tend to work well for the `CBOpt` family.

Gradient momentum β_1. We set $\beta_1 = 0.9$ throughout, following the standard momentum value used in SGD and Adam. We did not find it necessary to tune this parameter extensively.

Hessian momentum β_2. The Hessian momentum should be set close to one so that the curvature estimate changes smoothly. We use $\beta_2 = 0.99999$ in all classification experiments with `CBOpt`, following the setting commonly used in IVON. Smaller values may make the curvature estimate more responsive, but can also introduce instability, especially when combined with adaptive curvature tracking.

Algorithm 2: **u**pper **CBO** Optimization with adaptive curvature (`uCBOpt-adapt`)

Require: Learning rates $\{\rho_t\}$, weight decay $\delta > 0$, curvature parameters $\gamma \in [0, 1)$ and $\beta_3 > 1$, stability constant $\epsilon = 10^{-8}$, momentum parameters $\beta_1, \beta_2 \in [0, 1)$, and Hessian init $h_0 > 0$.
Init: $\eta \leftarrow$ (NN-weights), $m \leftarrow 0$, $h \leftarrow h_0$, $c \leftarrow +\infty$.
for $t = 1, 2, \ldots$ **do**
  $\hat{g} \leftarrow \widehat{\nabla}\ell(\eta)$
  $\hat{h} \leftarrow \hat{g} \odot \hat{g}$
  $m \leftarrow \beta_1 m + (1 - \beta_1)\hat{g}$
  $h \leftarrow \beta_2 h + (1 - \beta_2)\hat{h}$
  $\tilde{h} \leftarrow h + \delta$
  $c \leftarrow \min\{\beta_3 c, \tilde{h}\}$
  $\bar{m} \leftarrow m/(1 - \beta_1^t)$
  $\eta \leftarrow \eta - \rho_t(\bar{m} + \delta\eta)/(\tilde{h} - \gamma c + \epsilon)$
end for
return η

Algorithm 3: **l**ower **CBO** Optimization with adaptive curvature (`lCBOpt-adapt`)

Require: Learning rates $\{\rho_t\}$, weight decay $\delta > 0$, curvature parameters $\gamma > 1$ and $\beta_3 \in [0, 1)$, stability constant $\epsilon = 10^{-8}$, momentum parameters $\beta_1, \beta_2 \in [0, 1)$, and Hessian init $h_0 > 0$.
Init: $\eta \leftarrow$ (NN-weights), $m \leftarrow 0$, $h \leftarrow h_0$, $c \leftarrow h_0 + \delta$.
for $t = 1, 2, \ldots$ **do**
  $\hat{g} \leftarrow \widehat{\nabla}\ell(\eta)$
  $\hat{h} \leftarrow \hat{g} \odot \hat{g}$
  $m \leftarrow \beta_1 m + (1 - \beta_1)\hat{g}$
  $h \leftarrow \beta_2 h + (1 - \beta_2)\hat{h}$
  $\tilde{h} \leftarrow h + \delta$
  $c \leftarrow \max\{\beta_3 c, \tilde{h}\}$
  $\bar{m} \leftarrow m/(1 - \beta_1^t)$
  $\eta \leftarrow \eta - \rho_t(\bar{m} + \delta\eta)/(\gamma c - \tilde{h} + \epsilon)$
end for
return η

Curvature parameters $\vartheta, \gamma, \beta_3$. For `uCBOpt`, the curvature parameter ϑ controls the fixed candidate curvature. A small value is usually preferable, often one to three orders of magnitude smaller than the weight decay. If ϑ is too small, the method behaves similarly to the base curvature update, whereas overly large values can excessively distort the effective preconditioner and degrade performance. For the adaptive variants, γ controls the strength of the curvature correction, while β_3 controls the evolution of the candidate curvature. In `uCBOpt-adapt`, $\beta_3 > 1$ controls the growth rate of the running-minimum candidate; in `lCBOpt-adapt`, $\beta_3 < 1$ controls the decay rate of the running-maximum candidate. We find that γ values close to one work well, using $\gamma = 0.9$ for `uCBOpt-adapt` and $\gamma = 1.02$ for `lCBOpt-adapt`. We find $\beta_3 = 1.001$ for `uCBOpt-adapt` and $\beta_3 = 0.999$ for `lCBOpt-adapt` to be good default choices.

Hessian initialization h_0. The Hessian initialization controls the initial scale of the curvature estimate. Smaller values allow more adaptive behaviour early in training, while larger values reduce the initial update scale and can stabilize training. In our experiments, we find that $h_0 \in \{0.05, 0.1\}$ tends to work well for the `CBOpt` family.

Stability term ϵ. In our implementation of `uCBOpt-adapt` and `lCBOpt-adapt`, we add a small numerical stability constant $\epsilon = 10^{-8}$ to the denominator, following Adam, to prevent division by near-zero curvature estimates.

Batch size, training epochs. Batch sizes and training budgets that work well for SGD and AdamW generally also work well for the `CBOpt` family. For the adaptive variants, since curvature estimates are accumulated during training, longer training can improve their stability, especially for larger models and datasets.

H Experimental Details

H.1 Computational Efficiency

In Table 3, we report the training runtime and the peak allocated and reserved GPU memory. All classification experiments are conducted on a single NVIDIA H100 GPU, with three random seeds executed concurrently in separate processes. Consequently, the reported runtimes may be affected by resource contention, while GPU memory measurements are recorded per process. We evaluate `uCBOpt`, whose runtime and memory requirements are comparable to those of existing optimizers

such as SGD and AdamW. This is expected, as `uCBOpt` only requires maintaining per-parameter gradient and scalar curvature estimates, similar to the moment estimates used in Adam.

Table 3: First-epoch training runtime and peak-per-process GPU memory for classification experiments.

Dataset / Model	Metric	AdamW	SGD	IVON	`CBOpt` variants
Fashion-MNIST LeNet	Runtime (seconds)	3.26	3.52	3.30	3.28
	Memory allocated (GiB)	0.07	0.07	0.07	0.07
	Memory reserved (GiB)	0.09	0.09	0.09	0.09
CIFAR-10 ResNet-20	Runtime (seconds)	13.7	13.7	13.8	13.7
	Memory allocated (GiB)	0.60	0.59	0.60	0.60
	Memory reserved (GiB)	0.83	0.82	0.82	0.83
CIFAR-100 DenseNet-121	Runtime (seconds)	84.5	83.2	83.5	86.9
	Memory allocated (GiB)	4.52	4.52	4.53	4.52
	Memory reserved (GiB)	4.95	4.95	4.96	4.95

H.2 Architectures and Datasets

In-domain datasets. For in-domain tasks, we consider three standard image classification benchmarks: **Fashion-MNIST** (Xiao et al., 2017), consisting of grayscale images from 10 clothing categories; **CIFAR-10** (Krizhevsky, 2009), consisting of RGB natural images from 10 classes; and **CIFAR-100** (Krizhevsky, 2009), a more fine-grained RGB natural image dataset with 100 classes.

Out-of-domain datasets. For out-of-domain evaluation, we consider three datasets: **EMNIST** (Cohen et al., 2017), consisting of grayscale images of handwritten English letters (26 classes); **SVHN** (Netzer et al., 2011), consisting of RGB images of real-world house numbers cropped from Google Street View (10 classes); and **TinyImageNet** (Le and Yang, 2015), consisting of RGB natural images from 200 object classes. We use the EMNIST Letters test split (20,800 images), the SVHN test split (26,032 images), and the TinyImageNet validation split (10,000 images). These are evaluated against the full test sets of Fashion-MNIST, CIFAR-10, and CIFAR-100, respectively, each containing 10,000 images. No subsampling or balancing of in-domain and OOD samples is performed.

Table 4: Channel-wise mean and standard deviations used for normalizing datasets.

Dataset	Mean	Standard Deviation
SVHN	0.438, 0.444, 0.473	0.198, 0.201, 0.197
FashionMNIST	0.286	0.353
EMNIST	0.174	0.332
CIFAR-10	0.485, 0.456, 0.406	0.229, 0.224, 0.225
CIFAR-100	0.485, 0.456, 0.406	0.229, 0.224, 0.225
TinyImageNet	0.480, 0.448, 0.398	0.230, 0.227, 0.226

Data preprocessing. All images are normalized channel-wise using standard normalization constants listed in Table 4, following the preprocessing choices of Shen et al. (2024) where applicable. In particular, CIFAR-10 and CIFAR-100 use ImageNet normalization constants. During training, CIFAR-10 and CIFAR-100 images are augmented with random horizontal flips and random crops with padding, while Fashion-MNIST uses no data augmentation. For OOD evaluation, EMNIST and SVHN images undergo only tensor conversion and normalization, without additional orientation correction for EMNIST. To match the sizes in other datasets, TinyImageNet images are resized from the original 64×64 resolution to 32×32 using resizing and center cropping, followed by tensor conversion and normalization.

Architectures. We consider three image classification models: LeNet, ResNet-20 and DenseNet-121. On Fashion-MNIST, we train a LeNet; on CIFAR-10 we train a ResNet-20; and on CIFAR-100 we train a DenseNet-121.

- **LeNet** (LeCun et al., 1998). Consists of two convolutional layers with ReLU activations and max-pooling, followed by three fully connected layers. The model has 44,426 trainable parameters for Fashion-MNIST.
- **ResNet-20** (He et al., 2016). Consists of an initial convolutional layer followed by three residual stages with channel widths 16, 32, and 64, using downsampling at the beginning of the latter two stages. Global average pooling is applied before the final linear classifier. We use Filter Response Normalization (FRN) in the convolutional blocks, which normalizes each filter response using its spatial second moment and applies learned affine and threshold parameters. The model has 274,042 trainable parameters for CIFAR-10.
- **DenseNet-121** (Huang et al., 2017). Consists of an initial convolutional layer followed by four dense blocks with 6, 12, 24, and 16 bottleneck layers, respectively, using transition layers between blocks for downsampling and channel compression. Each bottleneck uses FRN-normalized 1×1 and 3×3 convolutions, and concatenates its output with previous feature maps to promote feature reuse. Global average pooling is applied before the final linear classifier. We use growth rate 12 and compression factor 0.5. The model has 1,050,928 trainable parameters for CIFAR-100.

For MC Dropout variants, we use the same backbone architectures but keep dropout active at inference time to obtain stochastic predictions. In LeNet, dropout is applied after the first two fully connected hidden layers. In ResNet-20, it is applied within the residual blocks after the FRN-normalized convolutional responses and residual addition. In DenseNet-121, dropout is applied inside the dense bottleneck layers after the FRN-normalized convolutional transformations and before concatenating the newly produced feature maps with the accumulated dense-block features.

H.3 Hyperparameters

Across the classification experiments, we use a 90/10 train-validation split. Each model is trained for a fixed number of epochs, and test performance is reported using the checkpoint with the lowest validation loss. We use a batch size of 128 for training and 256 for validation and testing. Except for the SWAG schedule described below, all methods use a linear warmup for the first 5 epochs, followed by cosine learning-rate annealing to zero. For the baselines, hyperparameters are tuned only on Fashion-MNIST, while those for the remaining datasets follow Shen et al. (2024). For the `CBOpt` family, we extensively tune the curvature parameters only on Fashion-MNIST. For CIFAR-10, we tune the learning rate and, for `uCBOpt`, the curvature parameter ϑ. For `uCBOpt-adapt` and `lCBOpt-adapt`, the curvature settings are listed below. For CIFAR-100, we use the same hyperparameter settings as CIFAR-10.

Fashion-MNIST. We train all models for 100 epochs. Hyperparameters are selected based on the lowest validation loss over 30 epochs. Details of the hyperparameter selection are provided below.

- **AdamW.** We tune the learning rate ρ over a grid from 10^{-3} to 10^{-2} and weight decay δ from 10^{-5} to 10^{-2}. We use $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The chosen hyperparameters are $\rho = 10^{-3}$, $\delta = 10^{-2}$.
- **SGD.** We tune the learning rate ρ over a grid from 10^{-3} to 10^{-1} and weight decay δ from 10^{-5} to $5 \cdot 10^{-3}$. We use momentum 0.9. The chosen hyperparameters are $\rho = 5 \cdot 10^{-3}$ and $\delta = 10^{-5}$.
- **IVON.** We tune the learning rate ρ over a grid from 10^{-3} to $2 \cdot 10^{-1}$, weight decay δ from 10^{-5} to $5 \cdot 10^{-3}$, and initial Hessian scale over $h_0 \in \{0.05, 0.1, 0.5\}$. We use $\beta_1 = 0.9$, $\beta_2 = 0.99999$, effective sample size 54000 (equivalent to the number of training samples) and one MC sample during training. The chosen hyperparameters are $\rho = 2 \cdot 10^{-1}$, $\delta = 2 \cdot 10^{-3}$, and $h_0 = 0.5$. We also apply the learning-rate rescaling procedure from Shen et al. (2024).
- **MC-Dropout.** We use the LeNet MC-Dropout variant with dropout probability $p = 0.05$, keeping dropout active at test time. The model is trained with SGD using $\rho = 5 \cdot 10^{-3}$ and $\delta = 10^{-5}$.
- **Laplace.** We first train a MAP LeNet model using SGD with momentum 0.9, $\rho = 5 \cdot 10^{-3}$, and $\delta = 10^{-5}$. We then fit a post-hoc last-layer Laplace approximation with a Kronecker-factored Hessian structure. For in-domain evaluation, we use the analytic GLM predictive approximation with a probit link. For OOD evaluation, we use the GLM Monte Carlo approximation with 64 samples drawn from the Gaussian distribution over logits. The prior precision is initialized to 0.54 (corresponding to the same prior setup as SGD) and optimized by marginal likelihood.

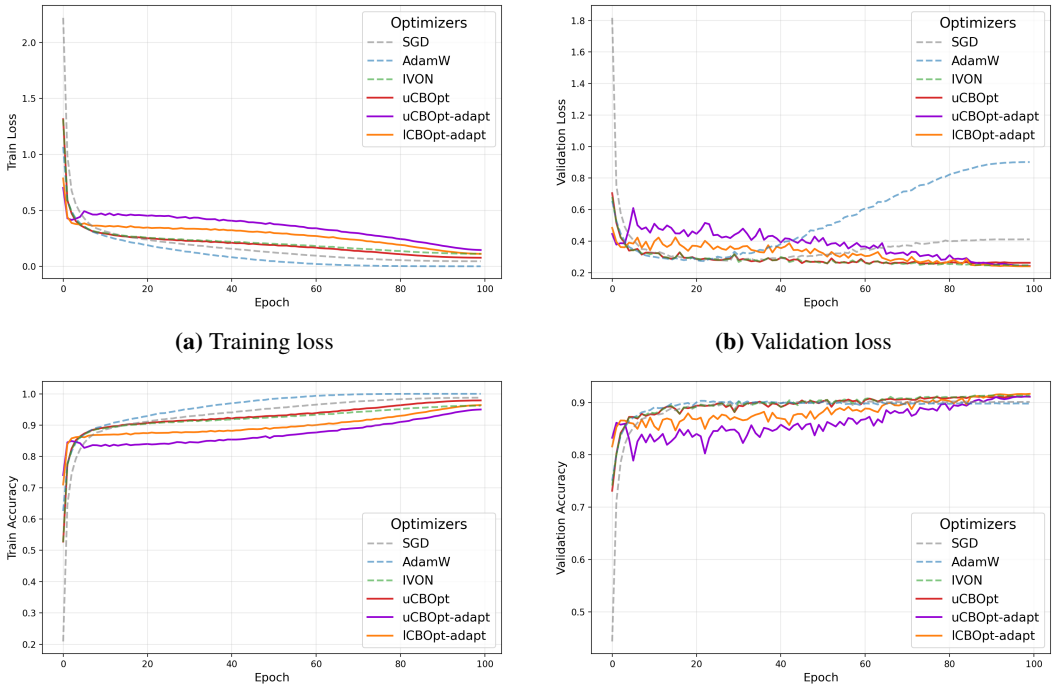
- **SWAG.** We first perform standard SGD training with momentum 0.9 using cosine annealing from an initial learning rate of $5 \cdot 10^{-3}$ to a final learning rate of 10^{-3} over the first 60 epochs. We then run 40 additional SWAG epochs with a constant learning rate of $5 \cdot 10^{-3}$, collecting SWAG statistics every epoch and maintaining a low-rank approximation of the posterior. We use weight decay $\delta = 10^{-5}$.
- `uCBOpt`**.** We tune the learning rate ρ over a grid from 10^{-3} to 10^{-1}, weight decay δ from 10^{-5} to $5 \cdot 10^{-3}$, initial Hessian scale over $h_0 \in \{0.05, 0.1, 0.5, 1.0\}$, and curvature parameter ϑ over a fine-grained grid from 0 to 10^{-4}. We use $\beta_1 = 0.9$ and $\beta_2 = 0.99999$. The chosen hyperparameters are $\rho = 10^{-2}$, $\delta = 2 \cdot 10^{-3}$, $h_0 = 0.05$, and $\vartheta = 8 \cdot 10^{-6}$.
- `uCBOpt-adapt`**.** We tune the learning rate ρ over a grid from 10^{-3} to 10^{-1}, weight decay δ from 10^{-5} to $5 \cdot 10^{-3}$, initial Hessian scale over $h_0 \in \{0.05, 0.1, 0.2\}$, curvature scaling parameter γ from 0.0 to 0.95, and curvature tracking parameter β_3 over $\{1.00001, 1.0001, 1.001, 1.01\}$. We use $\beta_1 = 0.9$ and $\beta_2 = 0.99999$. The chosen hyperparameters are $\rho = 10^{-2}$, $\delta = 2 \cdot 10^{-3}$, $h_0 = 0.05$, $\gamma = 0.9$, and $\beta_3 = 1.001$.
- `lCBOpt-adapt`**.** We tune the learning rate ρ over a grid from 10^{-3} to 10^{-1}, weight decay δ from 10^{-5} to $5 \cdot 10^{-3}$, initial Hessian scale over $h_0 \in \{0.05, 0.1, 0.2\}$, curvature scaling parameter γ from 1.0001 to 2.0, and curvature tracking parameter β_3 over $\{0.9, 0.99, 0.999, 0.9999, 0.99999\}$. We use $\beta_1 = 0.9$ and $\beta_2 = 0.99999$. The chosen hyperparameters are $\rho = 2 \cdot 10^{-3}$, $\delta = 2 \cdot 10^{-3}$, $h_0 = 0.1$, $\gamma = 1.02$, and $\beta_3 = 0.999$.

CIFAR-10 & CIFAR-100. We train all models for 200 epochs. Details of the hyperparameters are provided below.

- **AdamW.** We use $\rho = 2 \cdot 10^{-3}$ and $\delta = 2 \cdot 10^{-4}$, with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.
- **SGD.** We use $\rho = 10^{-1}$, momentum 0.9, and $\delta = 2 \cdot 10^{-4}$.
- **IVON.** We use $\rho = 0.2$, $\delta = 2 \cdot 10^{-4}$, $h_0 = 0.5$, $\beta_1 = 0.9$, $\beta_2 = 0.99999$, effective sample size 45000 (equivalent to the number of training samples) and one MC sample during training. We also apply the learning-rate rescaling procedure from Shen et al. (2024).
- **MC-Dropout.** We use the ResNet-20 (for CIFAR-10) and DenseNet-121 (for CIFAR-100) MC-Dropout variant with dropout probability $p = 0.05$, keeping dropout active at test time. The model is trained with SGD using momentum 0.9, $\rho = 10^{-1}$, and $\delta = 2 \times 10^{-4}$.
- **Laplace.** We first train MAP ResNet-20 (for CIFAR-10) and DenseNet-121 (for CIFAR-100) models using SGD with momentum 0.9, $\rho = 10^{-1}$, and weight decay $\delta = 2 \cdot 10^{-4}$. We then fit a post-hoc last-layer Laplace approximation with a Kronecker-factored Hessian structure. For in-domain evaluation, we use the analytic GLM predictive approximation with a probit link. For OOD evaluation, we use the GLM Monte Carlo approximation with 64 samples drawn from the Gaussian distribution over logits. The prior precision is initialized to 9 (corresponding to the same prior setup as SGD) and optimized by marginal likelihood.
- **SWAG.** We first perform standard SGD training with momentum 0.9 using cosine annealing from an initial learning rate of $5 \cdot 10^{-2}$ to a final learning rate of 10^{-2} over the first 160 epochs. We then run 40 additional SWAG epochs with a constant learning rate of $5 \cdot 10^{-3}$, collecting SWAG statistics every epoch and maintaining a low-rank approximation of the posterior. We use weight decay $\delta = 2 \cdot 10^{-4}$.
- `uCBOpt`**.** We use $\rho = 10^{-1}$, $\delta = 2 \cdot 10^{-4}$, $h_0 = 0.05$, $\vartheta = 5 \cdot 10^{-5}$, $\beta_1 = 0.9$, and $\beta_2 = 0.99999$.
- `uCBOpt-adapt`**.** We use $\rho = 5 \cdot 10^{-3}$, $\delta = 2 \cdot 10^{-4}$, $h_0 = 0.05$, $\gamma = 0.9$, $\beta_1 = 0.9$, $\beta_2 = 0.99999$, and $\beta_3 = 1.001$.
- `lCBOpt-adapt`**.** We use $\rho = 5 \cdot 10^{-3}$, $\delta = 2 \cdot 10^{-4}$, $h_0 = 0.1$, $\gamma = 1.02$, $\beta_1 = 0.9$, $\beta_2 = 0.99999$, and $\beta_3 = 0.999$.

H.4 Performance Metrics

In-domain tasks. For the in-domain tasks, we report metrics that capture classification performance, predictive uncertainty, and calibration quality. Classification performance is measured using top-1 (Acc.) and top-5 accuracy (Top-5 Acc.). We also report negative log likelihood (NLL) and Brier score to evaluate the quality of probabilistic predictions. Calibration is assessed using expected calibration error (ECE), computed over 20 equally spaced confidence bins, weighted by the proportion of samples in each bin. The multiclass Brier score is computed by summing the squared differences between predicted probabilities and one-hot labels across classes, averaged over samples. AUROC is computed using predictive confidence to measure how well the model separates correct from incorrect predictions. For `CBOpt`, predictive probabilities are obtained by applying softmax to the



(a) Training loss

(b) Validation loss

(c) Training accuracy

(d) Validation accuracy

Figure 2: Training and validation loss and accuracy curves for all methods on the Fashion-MNIST/LeNet experiment.

network logits at the learned parameters. Consequently, ECE and Brier score assess the quality of the resulting predictive probabilities, rather than the epistemic uncertainty represented by the possibility function. Although top-5 accuracy is less selective for 10-class datasets, we include it for consistency across datasets.

Out-of-domain tasks. For the out-of-domain tasks, we report metrics that evaluate how well the model separates in-domain from out-of-domain examples using uncertainty scores. We use the maximum predictive class probability, $s(x) = \max_k \hat{p}(y = k \mid x)$, as the in-domain confidence score, treating in-domain examples as the positive class. We use the false positive rate at 95% true positive rate (FPR@95) and detection error (Det. Err.) to assess threshold-based OOD detection performance. FPR@95 is computed at the first ROC threshold achieving a true positive rate of at least 95%. Detection error is defined as $\mathrm{DetErr} = \min_\tau [\mathrm{FPR}(\tau) + \mathrm{FNR}(\tau)]/2$, assuming equal prior probabilities for in-domain and out-of-domain examples, where the threshold τ is optimized independently of FPR@95. We also report the area under the receiver operating characteristic curve (AUROC), which measures how well the uncertainty scores distinguish between in-domain and out-of-domain examples across all detection thresholds. Lastly, we report the area under the precision–recall curve with in-domain examples treated as positive (AUPR-In) and with out-of-domain examples treated as positive (AUPR-Out). These metrics measure how well the model identifies in-domain and out-of-domain examples, respectively, under precision–recall trade-offs.

I Full Experimental Results

The complete results corresponding to Table 1 are reported in Table 5 for the in-domain tasks and in Table 6 for the out-of-domain tasks. For completeness, we also report the final-epoch model performance in Table 7 and Table 8. We note that our final-epoch results differ from those reported by Shen et al. (2024), mainly due to two differences in the experimental setup: (1) we use a training batch size of 128, whereas they use 50; and (2) we train on 90% of the training set and reserve the remaining 10% for validation, whereas they train on the full training set.

Across the experiments, CBOpt is most beneficial for improving in-domain predictive performance while maintaining competitive uncertainty estimates. The adaptive variants are particularly effective on Fashion-MNIST, where uCBOpt-adapt and lCBOpt-adapt obtain the best or near-best accuracy, NLL, Brier score, and AUROC. On CIFAR-10, all CBOpt variants remain competitive with SGD, Laplace, and IVON, although they do not outperform stronger stochastic baselines such as SWAG and MC dropout. On CIFAR-100, the advantages become clearer: fixed-ϑ uCBOpt and lCBOpt-adapt improve over the considered baselines on several in-domain metrics, suggesting that the proposed updates scale well to more challenging classification tasks.

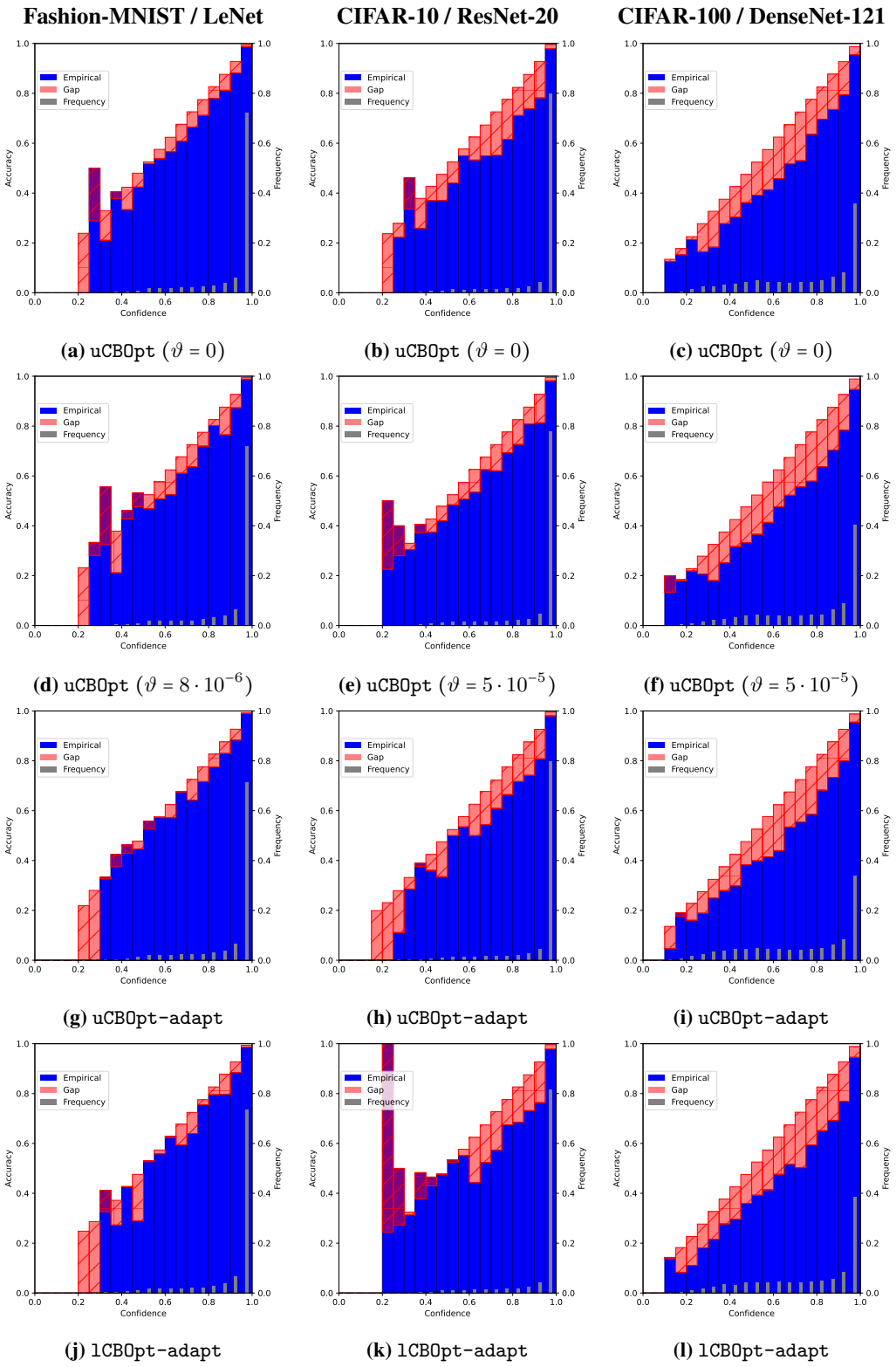
For OOD detection, uCBOpt gives the most consistent behaviour across datasets. It is competitive on Fashion-MNIST/EMNIST and CIFAR-10/SVHN, and achieves strong results on CIFAR-100/TinyImageNet. The adaptive variants are useful in some settings, particularly uCBOpt-adapt on CIFAR-100/TinyImageNet, but they are not uniformly better for OOD detection. Overall, the adaptive variants appear most helpful for in-domain performance, while fixed-ϑ uCBOpt provides the most stable OOD performance.

We also present the training and validation loss and accuracy curves for LeNet trained on Fashion-MNIST in Figure 2. The results show that uCBOpt exhibits training dynamics similar to IVON, achieving stable optimization and strong validation performance throughout training. In contrast, uCBOpt-adapt and lCBOpt-adapt converge more slowly, with higher training and validation losses in the early and middle stages. However, both adaptive variants continue to improve steadily and eventually reach validation accuracies comparable to the strongest baselines. AdamW achieves the lowest training loss and highest training accuracy, but its validation loss increases in later epochs, indicating stronger overfitting. By comparison, the CBOpt variants maintain more stable validation behaviour, suggesting better generalization.

We present the calibration plots of the CBOpt family of optimizers in Figure 3. Calibration is strongest on Fashion-MNIST; the CIFAR-100 plots show appreciable overconfidence, consistent with the higher ECE values.

Table 5: Experimental results on in-domain tasks over three runs (best validation model).

Dataset	Optimizer	Acc. ↑	Top-5 Acc. ↑	NLL ↓	ECE ↓	Brier ↓	AUROC ↑
Fashion-MNIST	AdamW	$0.900_{\pm 0.005}$	$0.998_{\pm 0.000}$	$0.294_{\pm 0.007}$	$0.026_{\pm 0.004}$	$0.148_{\pm 0.004}$	$0.901_{\pm 0.005}$
	SGD	$0.896_{\pm 0.005}$	$0.999_{\pm 0.000}$	$0.293_{\pm 0.005}$	$0.022_{\pm 0.004}$	$0.151_{\pm 0.004}$	$0.904_{\pm 0.002}$
	IVON@mean	$0.907_{\pm 0.001}$	$0.999_{\pm 0.000}$	$0.267_{\pm 0.005}$	$0.018_{\pm 0.008}$	$0.136_{\pm 0.001}$	$0.906_{\pm 0.002}$
	IVON	$0.911_{\pm 0.003}$	$0.999_{\pm 0.000}$	$0.258_{\pm 0.001}$	$0.013_{\pm 0.004}$	$0.132_{\pm 0.002}$	$0.904_{\pm 0.003}$
	MC-D	$0.897_{\pm 0.003}$	$0.999_{\pm 0.000}$	$0.281_{\pm 0.007}$	$0.013_{\pm 0.003}$	$0.146_{\pm 0.004}$	$0.906_{\pm 0.003}$
	Laplace	$0.895_{\pm 0.004}$	$0.999_{\pm 0.001}$	$0.295_{\pm 0.009}$	$0.024_{\pm 0.003}$	$0.152_{\pm 0.005}$	$0.902_{\pm 0.003}$
	SWAG	$0.900_{\pm 0.003}$	$0.999_{\pm 0.000}$	$0.342_{\pm 0.009}$	$0.043_{\pm 0.003}$	$0.153_{\pm 0.005}$	$0.899_{\pm 0.004}$
	uCBOpt ($\vartheta \to 0^+$)	$0.905_{\pm 0.002}$	$0.998_{\pm 0.000}$	$0.269_{\pm 0.004}$	$0.020_{\pm 0.006}$	$0.138_{\pm 0.003}$	$0.908_{\pm 0.003}$
	uCBOpt ($\vartheta = 8 \cdot 10^{-6}$)	$0.907_{\pm 0.002}$	$0.999_{\pm 0.000}$	$0.270_{\pm 0.002}$	$0.023_{\pm 0.000}$	$0.137_{\pm 0.003}$	$0.905_{\pm 0.003}$
	uCBOpt-adapt	$0.910_{\pm 0.001}$	$0.999_{\pm 0.000}$	$0.256_{\pm 0.003}$	$0.016_{\pm 0.001}$	$0.132_{\pm 0.001}$	$0.910_{\pm 0.002}$
	lCBOpt-adapt	$0.913_{\pm 0.002}$	$0.998_{\pm 0.000}$	$0.252_{\pm 0.004}$	$0.020_{\pm 0.000}$	$0.129_{\pm 0.001}$	$0.910_{\pm 0.003}$
CIFAR-10	AdamW	$0.872_{\pm 0.001}$	$0.995_{\pm 0.000}$	$0.406_{\pm 0.002}$	$0.042_{\pm 0.003}$	$0.189_{\pm 0.002}$	$0.893_{\pm 0.001}$
	SGD	$0.909_{\pm 0.003}$	$0.997_{\pm 0.001}$	$0.308_{\pm 0.011}$	$0.039_{\pm 0.008}$	$0.139_{\pm 0.002}$	$0.914_{\pm 0.004}$
	IVON@mean	$0.892_{\pm 0.003}$	$0.996_{\pm 0.001}$	$0.358_{\pm 0.011}$	$0.045_{\pm 0.008}$	$0.162_{\pm 0.003}$	$0.908_{\pm 0.002}$
	IVON	$0.899_{\pm 0.004}$	$0.996_{\pm 0.001}$	$0.317_{\pm 0.004}$	$0.028_{\pm 0.005}$	$0.149_{\pm 0.005}$	$0.910_{\pm 0.006}$
	MC-D	$0.923_{\pm 0.002}$	$0.998_{\pm 0.001}$	$0.230_{\pm 0.006}$	$0.010_{\pm 0.002}$	$0.113_{\pm 0.002}$	$0.923_{\pm 0.002}$
	Laplace	$0.909_{\pm 0.001}$	$0.997_{\pm 0.000}$	$0.303_{\pm 0.008}$	$0.035_{\pm 0.006}$	$0.139_{\pm 0.001}$	$0.911_{\pm 0.002}$
	SWAG	$0.918_{\pm 0.001}$	$0.997_{\pm 0.001}$	$0.251_{\pm 0.004}$	$0.009_{\pm 0.000}$	$0.121_{\pm 0.002}$	$0.920_{\pm 0.002}$
	uCBOpt ($\vartheta \to 0^+$)	$0.910_{\pm 0.004}$	$0.996_{\pm 0.000}$	$0.296_{\pm 0.009}$	$0.038_{\pm 0.001}$	$0.135_{\pm 0.005}$	$0.918_{\pm 0.003}$
	uCBOpt ($\vartheta = 5 \cdot 10^{-5}$)	$0.907_{\pm 0.003}$	$0.997_{\pm 0.000}$	$0.302_{\pm 0.004}$	$0.033_{\pm 0.004}$	$0.140_{\pm 0.004}$	$0.914_{\pm 0.000}$
	uCBOpt-adapt	$0.909_{\pm 0.005}$	$0.997_{\pm 0.001}$	$0.300_{\pm 0.016}$	$0.038_{\pm 0.004}$	$0.138_{\pm 0.008}$	$0.917_{\pm 0.001}$
	lCBOpt-adapt	$0.909_{\pm 0.005}$	$0.997_{\pm 0.000}$	$0.299_{\pm 0.002}$	$0.037_{\pm 0.002}$	$0.138_{\pm 0.004}$	$0.915_{\pm 0.003}$
CIFAR-100	SGD	$0.649_{\pm 0.004}$	$0.891_{\pm 0.005}$	$1.334_{\pm 0.006}$	$0.096_{\pm 0.006}$	$0.483_{\pm 0.002}$	$0.838_{\pm 0.002}$
	IVON@mean	$0.599_{\pm 0.001}$	$0.864_{\pm 0.002}$	$1.521_{\pm 0.008}$	$0.092_{\pm 0.006}$	$0.538_{\pm 0.002}$	$0.828_{\pm 0.001}$
	IVON	$0.607_{\pm 0.002}$	$0.870_{\pm 0.003}$	$1.458_{\pm 0.012}$	$0.071_{\pm 0.004}$	$0.523_{\pm 0.002}$	$0.830_{\pm 0.003}$
	MC-D	$0.649_{\pm 0.016}$	$0.894_{\pm 0.003}$	$1.298_{\pm 0.016}$	$0.074_{\pm 0.020}$	$0.476_{\pm 0.013}$	$0.838_{\pm 0.006}$
	uCBOpt ($\vartheta \to 0^+$)	$0.667_{\pm 0.007}$	$0.905_{\pm 0.001}$	$1.247_{\pm 0.001}$	$0.094_{\pm 0.007}$	$0.457_{\pm 0.005}$	$0.845_{\pm 0.001}$
	uCBOpt ($\vartheta = 5 \cdot 10^{-5}$)	$0.671_{\pm 0.014}$	$0.906_{\pm 0.004}$	$1.231_{\pm 0.005}$	$0.095_{\pm 0.014}$	$0.453_{\pm 0.009}$	$0.847_{\pm 0.002}$
	uCBOpt-adapt	$0.655_{\pm 0.011}$	$0.897_{\pm 0.006}$	$1.298_{\pm 0.011}$	$0.096_{\pm 0.021}$	$0.475_{\pm 0.008}$	$0.841_{\pm 0.003}$
	lCBOpt-adapt	$0.674_{\pm 0.007}$	$0.907_{\pm 0.003}$	$1.268_{\pm 0.018}$	$0.110_{\pm 0.002}$	$0.457_{\pm 0.008}$	$0.845_{\pm 0.005}$



(a) uCBOpt $(\vartheta = 0)$ **(b)** uCBOpt $(\vartheta = 0)$ **(c)** uCBOpt $(\vartheta = 0)$

(d) uCBOpt $(\vartheta = 8 \cdot 10^{-6})$ **(e)** uCBOpt $(\vartheta = 5 \cdot 10^{-5})$ **(f)** uCBOpt $(\vartheta = 5 \cdot 10^{-5})$

(g) uCBOpt-adapt **(h)** uCBOpt-adapt **(i)** uCBOpt-adapt

(j) lCBOpt-adapt **(k)** lCBOpt-adapt **(l)** lCBOpt-adapt

Figure 3: Reliability diagrams for the CBOpt family.

Table 6: Experimental results on out-of-domain tasks over three runs (best validation model).

Dataset	Optimizer	FPR@95% ↓	Det. Err. ↓	AUROC ↑	AUPR-In ↑	AUPR-Out ↑
In: Fashion-MNIST **Out:** EMNIST	AdamW	$0.827_{\pm 0.006}$	$0.312_{\pm 0.009}$	$0.745_{\pm 0.009}$	$0.627_{\pm 0.022}$	$0.834_{\pm 0.003}$
	SGD	$0.816_{\pm 0.013}$	$0.293_{\pm 0.018}$	$0.767_{\pm 0.022}$	$0.676_{\pm 0.043}$	$0.845_{\pm 0.012}$
	IVON@mean	$0.784_{\pm 0.036}$	$0.279_{\pm 0.026}$	$0.787_{\pm 0.029}$	$0.694_{\pm 0.047}$	$0.861_{\pm 0.019}$
	IVON	$\mathbf{0.753}_{\pm 0.040}$	$0.265_{\pm 0.028}$	$0.804_{\pm 0.030}$	$0.719_{\pm 0.048}$	$0.873_{\pm 0.020}$
	MC-D	$\mathbf{0.730}_{\pm 0.024}$	$0.250_{\pm 0.007}$	$0.821_{\pm 0.008}$	$0.752_{\pm 0.017}$	$0.885_{\pm 0.007}$
	Laplace	$0.805_{\pm 0.056}$	$0.290_{\pm 0.030}$	$0.770_{\pm 0.039}$	$0.675_{\pm 0.046}$	$0.848_{\pm 0.031}$
	SWAG	$0.855_{\pm 0.008}$	$0.324_{\pm 0.001}$	$0.731_{\pm 0.005}$	$0.601_{\pm 0.027}$	$0.820_{\pm 0.002}$
	`uCBOpt` ($\vartheta \to 0^+$)	$0.776_{\pm 0.025}$	$0.270_{\pm 0.013}$	$0.796_{\pm 0.014}$	$0.708_{\pm 0.023}$	$0.867_{\pm 0.010}$
	`uCBOpt` ($\vartheta = 8 \cdot 10^{-6}$)	$\mathbf{0.739}_{\pm 0.026}$	$0.259_{\pm 0.007}$	$0.810_{\pm 0.012}$	$0.720_{\pm 0.037}$	$0.879_{\pm 0.010}$
	`uCBOpt-adapt`	$0.783_{\pm 0.009}$	$0.290_{\pm 0.015}$	$0.773_{\pm 0.015}$	$0.655_{\pm 0.016}$	$0.857_{\pm 0.009}$
	`lCBOpt-adapt`	$0.794_{\pm 0.016}$	$0.278_{\pm 0.016}$	$0.784_{\pm 0.019}$	$0.674_{\pm 0.049}$	$0.859_{\pm 0.011}$
In: CIFAR-10 **Out:** SVHN	AdamW	$0.840_{\pm 0.047}$	$0.227_{\pm 0.027}$	$0.820_{\pm 0.031}$	$0.769_{\pm 0.038}$	$0.881_{\pm 0.021}$
	SGD	$\mathbf{0.732}_{\pm 0.042}$	$0.195_{\pm 0.012}$	$0.864_{\pm 0.013}$	$0.808_{\pm 0.016}$	$0.918_{\pm 0.011}$
	IVON@mean	$0.827_{\pm 0.025}$	$0.221_{\pm 0.010}$	$0.830_{\pm 0.011}$	$0.779_{\pm 0.013}$	$0.890_{\pm 0.010}$
	IVON	$0.812_{\pm 0.042}$	$0.214_{\pm 0.011}$	$0.836_{\pm 0.014}$	$0.791_{\pm 0.016}$	$0.894_{\pm 0.012}$
	MC-D	$0.795_{\pm 0.053}$	$0.206_{\pm 0.021}$	$0.847_{\pm 0.024}$	$0.796_{\pm 0.027}$	$0.902_{\pm 0.017}$
	Laplace	$\mathbf{0.759}_{\pm 0.047}$	$0.204_{\pm 0.015}$	$0.852_{\pm 0.019}$	$0.797_{\pm 0.021}$	$0.909_{\pm 0.013}$
	SWAG	$\mathbf{0.728}_{\pm 0.035}$	$0.175_{\pm 0.004}$	$0.879_{\pm 0.007}$	$0.841_{\pm 0.006}$	$0.924_{\pm 0.007}$
	`uCBOpt` ($\vartheta \to 0^+$)	$\mathbf{0.745}_{\pm 0.044}$	$0.186_{\pm 0.008}$	$0.869_{\pm 0.012}$	$0.824_{\pm 0.013}$	$0.918_{\pm 0.010}$
	`uCBOpt` ($\vartheta = 5 \cdot 10^{-5}$)	$0.782_{\pm 0.026}$	$0.200_{\pm 0.001}$	$0.854_{\pm 0.004}$	$0.804_{\pm 0.001}$	$0.908_{\pm 0.006}$
	`uCBOpt-adapt`	$\mathbf{0.759}_{\pm 0.028}$	$0.187_{\pm 0.012}$	$0.866_{\pm 0.010}$	$0.821_{\pm 0.020}$	$0.915_{\pm 0.007}$
	`lCBOpt-adapt`	$\mathbf{0.760}_{\pm 0.016}$	$0.195_{\pm 0.006}$	$0.861_{\pm 0.006}$	$0.809_{\pm 0.007}$	$0.913_{\pm 0.003}$
In: CIFAR-100 **Out:** TinyImageNet	SGD	$0.848_{\pm 0.003}$	$0.337_{\pm 0.006}$	$0.720_{\pm 0.008}$	$0.744_{\pm 0.005}$	$0.684_{\pm 0.006}$
	IVON@mean	$0.868_{\pm 0.008}$	$0.354_{\pm 0.006}$	$0.702_{\pm 0.008}$	$0.726_{\pm 0.008}$	$0.663_{\pm 0.007}$
	IVON	$0.858_{\pm 0.008}$	$0.348_{\pm 0.004}$	$0.709_{\pm 0.007}$	$0.733_{\pm 0.007}$	$0.672_{\pm 0.007}$
	MC-D	$0.852_{\pm 0.010}$	$0.341_{\pm 0.005}$	$0.718_{\pm 0.008}$	$0.740_{\pm 0.007}$	$0.681_{\pm 0.009}$
	`uCBOpt` ($\vartheta \to 0^+$)	$0.846_{\pm 0.009}$	$0.330_{\pm 0.006}$	$0.727_{\pm 0.009}$	$0.751_{\pm 0.007}$	$0.686_{\pm 0.011}$
	`uCBOpt` ($\vartheta = 5 \cdot 10^{-5}$)	$0.831_{\pm 0.002}$	$0.331_{\pm 0.004}$	$0.727_{\pm 0.003}$	$0.749_{\pm 0.003}$	$0.693_{\pm 0.003}$
	`uCBOpt-adapt`	$\mathbf{0.828}_{\pm 0.002}$	$0.332_{\pm 0.003}$	$0.729_{\pm 0.002}$	$0.749_{\pm 0.005}$	$0.696_{\pm 0.001}$
	`lCBOpt-adapt`	$0.836_{\pm 0.009}$	$0.330_{\pm 0.004}$	$0.729_{\pm 0.005}$	$0.754_{\pm 0.005}$	$0.692_{\pm 0.006}$

Table 7: Experimental results on in-domain tasks over three runs (final epoch model).

Dataset	Optimizer	Acc. ↑	Top-5 Acc. ↑	NLL ↓	ECE ↓	Brier ↓	AUROC ↑
Fashion-MNIST	AdamW	$0.897_{\pm 0.000}$	$0.997_{\pm 0.001}$	$0.981_{\pm 0.091}$	$0.086_{\pm 0.002}$	$0.184_{\pm 0.001}$	$0.866_{\pm 0.010}$
	SGD	$0.899_{\pm 0.003}$	$0.998_{\pm 0.001}$	$0.441_{\pm 0.025}$	$0.061_{\pm 0.004}$	$0.163_{\pm 0.005}$	$0.893_{\pm 0.003}$
	IVON@mean	$0.908_{\pm 0.003}$	$0.999_{\pm 0.000}$	$0.271_{\pm 0.007}$	$0.026_{\pm 0.003}$	$0.137_{\pm 0.004}$	$0.905_{\pm 0.002}$
	IVON	$0.913_{\pm 0.001}$	$0.999_{\pm 0.000}$	$0.255_{\pm 0.005}$	$0.018_{\pm 0.002}$	$0.130_{\pm 0.002}$	$0.906_{\pm 0.001}$
	MC-D	$0.904_{\pm 0.003}$	$0.999_{\pm 0.000}$	$0.298_{\pm 0.007}$	$0.034_{\pm 0.001}$	$0.143_{\pm 0.003}$	$0.903_{\pm 0.001}$
	Laplace	$0.897_{\pm 0.005}$	$0.999_{\pm 0.000}$	$0.447_{\pm 0.010}$	$0.064_{\pm 0.003}$	$0.165_{\pm 0.007}$	$0.895_{\pm 0.003}$
	SWAG	$0.900_{\pm 0.003}$	$0.999_{\pm 0.000}$	$0.342_{\pm 0.009}$	$0.043_{\pm 0.003}$	$0.153_{\pm 0.005}$	$0.899_{\pm 0.004}$
	`uCBOpt` ($\vartheta \to 0^+$)	$0.909_{\pm 0.003}$	$0.998_{\pm 0.000}$	$0.279_{\pm 0.004}$	$0.031_{\pm 0.004}$	$0.136_{\pm 0.003}$	$0.905_{\pm 0.001}$
	`uCBOpt` ($\vartheta = 8 \cdot 10^{-6}$)	$0.910_{\pm 0.002}$	$0.998_{\pm 0.001}$	$0.278_{\pm 0.007}$	$0.032_{\pm 0.002}$	$0.135_{\pm 0.002}$	$0.905_{\pm 0.003}$
	`uCBOpt-adapt`	$0.910_{\pm 0.001}$	$0.999_{\pm 0.000}$	$0.256_{\pm 0.003}$	$0.016_{\pm 0.001}$	$0.132_{\pm 0.002}$	$0.910_{\pm 0.002}$
	`lCBOpt-adapt`	$0.913_{\pm 0.002}$	$0.998_{\pm 0.000}$	$0.252_{\pm 0.003}$	$0.021_{\pm 0.001}$	$0.128_{\pm 0.001}$	$0.911_{\pm 0.005}$
CIFAR-10	AdamW	$0.888_{\pm 0.001}$	$0.995_{\pm 0.000}$	$0.638_{\pm 0.013}$	$0.083_{\pm 0.002}$	$0.190_{\pm 0.001}$	$0.903_{\pm 0.003}$
	SGD	$0.917_{\pm 0.003}$	$0.997_{\pm 0.000}$	$0.311_{\pm 0.014}$	$0.044_{\pm 0.003}$	$0.130_{\pm 0.005}$	$0.916_{\pm 0.002}$
	IVON@mean	$0.901_{\pm 0.002}$	$0.995_{\pm 0.000}$	$0.390_{\pm 0.003}$	$0.159_{\pm 0.001}$	$0.055_{\pm 0.002}$	$0.905_{\pm 0.004}$
	IVON	$0.910_{\pm 0.001}$	$0.997_{\pm 0.000}$	$0.299_{\pm 0.004}$	$0.029_{\pm 0.002}$	$0.135_{\pm 0.002}$	$0.915_{\pm 0.002}$
	MC-D	$0.924_{\pm 0.001}$	$0.998_{\pm 0.001}$	$0.229_{\pm 0.005}$	$0.009_{\pm 0.005}$	$0.112_{\pm 0.002}$	$0.923_{\pm 0.002}$
	Laplace	$0.917_{\pm 0.002}$	$0.997_{\pm 0.000}$	$0.304_{\pm 0.009}$	$0.043_{\pm 0.002}$	$0.130_{\pm 0.004}$	$0.918_{\pm 0.004}$
	SWAG	$0.918_{\pm 0.001}$	$0.997_{\pm 0.001}$	$0.251_{\pm 0.004}$	$0.009_{\pm 0.000}$	$0.121_{\pm 0.002}$	$0.920_{\pm 0.002}$
	`uCBOpt` ($\vartheta \to 0^+$)	$0.915_{\pm 0.000}$	$0.997_{\pm 0.001}$	$0.294_{\pm 0.005}$	$0.040_{\pm 0.002}$	$0.131_{\pm 0.001}$	$0.919_{\pm 0.001}$
	`uCBOpt` ($\vartheta = 5 \cdot 10^{-5}$)	$0.916_{\pm 0.003}$	$0.997_{\pm 0.000}$	$0.299_{\pm 0.015}$	$0.040_{\pm 0.004}$	$0.131_{\pm 0.005}$	$0.919_{\pm 0.002}$
	`uCBOpt-adapt`	$0.917_{\pm 0.002}$	$0.997_{\pm 0.001}$	$0.310_{\pm 0.023}$	$0.045_{\pm 0.002}$	$0.131_{\pm 0.004}$	$0.920_{\pm 0.006}$
	`lCBOpt-adapt`	$0.911_{\pm 0.002}$	$0.997_{\pm 0.000}$	$0.301_{\pm 0.003}$	$0.039_{\pm 0.002}$	$0.136_{\pm 0.002}$	$0.916_{\pm 0.004}$
CIFAR-100	SGD	$0.697_{\pm 0.001}$	$0.907_{\pm 0.005}$	$1.365_{\pm 0.021}$	$0.150_{\pm 0.002}$	$0.448_{\pm 0.000}$	$0.856_{\pm 0.002}$
	IVON@mean	$0.647_{\pm 0.008}$	$0.878_{\pm 0.004}$	$1.684_{\pm 0.062}$	$0.182_{\pm 0.007}$	$0.527_{\pm 0.012}$	$0.836_{\pm 0.001}$
	IVON	$0.674_{\pm 0.008}$	$0.897_{\pm 0.007}$	$1.307_{\pm 0.048}$	$0.093_{\pm 0.004}$	$0.451_{\pm 0.010}$	$0.849_{\pm 0.003}$
	MC-D	$0.699_{\pm 0.006}$	$0.909_{\pm 0.003}$	$1.354_{\pm 0.016}$	$0.149_{\pm 0.003}$	$0.447_{\pm 0.007}$	$0.854_{\pm 0.002}$
	`uCBOpt` ($\vartheta \to 0^+$)	$0.701_{\pm 0.003}$	$0.914_{\pm 0.001}$	$1.278_{\pm 0.007}$	$0.133_{\pm 0.002}$	$0.438_{\pm 0.004}$	$0.848_{\pm 0.001}$
	`uCBOpt` ($\vartheta = 5 \cdot 10^{-5}$)	$0.696_{\pm 0.010}$	$0.912_{\pm 0.004}$	$1.276_{\pm 0.032}$	$0.136_{\pm 0.005}$	$0.441_{\pm 0.011}$	$0.854_{\pm 0.002}$
	`uCBOpt-adapt`	$0.705_{\pm 0.002}$	$0.913_{\pm 0.004}$	$1.315_{\pm 0.023}$	$0.145_{\pm 0.000}$	$0.438_{\pm 0.001}$	$0.854_{\pm 0.002}$
	`lCBOpt-adapt`	$0.689_{\pm 0.002}$	$0.912_{\pm 0.001}$	$1.277_{\pm 0.022}$	$0.127_{\pm 0.004}$	$0.446_{\pm 0.004}$	$0.850_{\pm 0.005}$

Table 8: Experimental results on out-of-domain tasks over three runs (final epoch model).

Dataset	Optimizer	FPR@95% ↓	Det. Err. ↓	AUROC ↑	AUPR-In ↑	AUPR-Out ↑
In: Fashion-MNIST **Out:** EMNIST	AdamW	$0.890_{\pm 0.005}$	$0.369_{\pm 0.008}$	$0.647_{\pm 0.007}$	$0.429_{\pm 0.008}$	$0.761_{\pm 0.004}$
	SGD	$0.898_{\pm 0.010}$	$0.368_{\pm 0.017}$	$0.674_{\pm 0.022}$	$0.508_{\pm 0.029}$	$0.780_{\pm 0.014}$
	IVON@mean	$0.784_{\pm 0.027}$	$0.268_{\pm 0.021}$	$0.796_{\pm 0.024}$	$0.708_{\pm 0.043}$	$0.866_{\pm 0.015}$
	IVON	$\mathbf{0.744}_{\pm 0.036}$	$0.249_{\pm 0.022}$	$0.820_{\pm 0.023}$	$0.743_{\pm 0.039}$	$0.883_{\pm 0.014}$
	MC-D	$\mathbf{0.773}_{\pm 0.010}$	$0.246_{\pm 0.003}$	$0.813_{\pm 0.002}$	$0.734_{\pm 0.013}$	$0.875_{\pm 0.002}$
	Laplace	$0.890_{\pm 0.007}$	$0.363_{\pm 0.012}$	$0.680_{\pm 0.015}$	$0.517_{\pm 0.014}$	$0.784_{\pm 0.011}$
	SWAG	$0.855_{\pm 0.008}$	$0.324_{\pm 0.001}$	$0.731_{\pm 0.005}$	$0.601_{\pm 0.027}$	$0.820_{\pm 0.002}$
	`uCBOpt` ($\vartheta \to 0^+$)	$0.791_{\pm 0.031}$	$0.267_{\pm 0.018}$	$0.796_{\pm 0.019}$	$0.700_{\pm 0.024}$	$0.865_{\pm 0.015}$
	`uCBOpt` ($\vartheta = 8 \cdot 10^{-6}$)	$0.784_{\pm 0.003}$	$0.271_{\pm 0.002}$	$0.794_{\pm 0.005}$	$0.694_{\pm 0.023}$	$0.865_{\pm 0.002}$
	`uCBOpt-adapt`	$0.783_{\pm 0.009}$	$0.290_{\pm 0.015}$	$0.773_{\pm 0.015}$	$0.655_{\pm 0.016}$	$0.857_{\pm 0.009}$
	`lCBOpt-adapt`	$0.797_{\pm 0.015}$	$0.278_{\pm 0.016}$	$0.784_{\pm 0.019}$	$0.674_{\pm 0.049}$	$0.859_{\pm 0.011}$
In: CIFAR-10 **Out:** SVHN	AdamW	$0.808_{\pm 0.036}$	$0.224_{\pm 0.024}$	$0.830_{\pm 0.027}$	$0.755_{\pm 0.042}$	$0.893_{\pm 0.017}$
	SGD	$\mathbf{0.757}_{\pm 0.024}$	$0.200_{\pm 0.003}$	$0.858_{\pm 0.005}$	$0.800_{\pm 0.007}$	$0.913_{\pm 0.005}$
	IVON@mean	$0.799_{\pm 0.021}$	$0.213_{\pm 0.011}$	$0.842_{\pm 0.013}$	$0.785_{\pm 0.018}$	$0.900_{\pm 0.008}$
	IVON	$0.772_{\pm 0.027}$	$0.193_{\pm 0.010}$	$0.860_{\pm 0.010}$	$0.816_{\pm 0.014}$	$0.911_{\pm 0.007}$
	MC-D	$0.797_{\pm 0.041}$	$0.210_{\pm 0.020}$	$0.844_{\pm 0.021}$	$0.792_{\pm 0.026}$	$0.900_{\pm 0.013}$
	Laplace	$\mathbf{0.746}_{\pm 0.030}$	$0.200_{\pm 0.016}$	$0.859_{\pm 0.015}$	$0.801_{\pm 0.022}$	$0.915_{\pm 0.010}$
	SWAG	$\mathbf{0.728}_{\pm 0.035}$	$0.175_{\pm 0.004}$	$0.879_{\pm 0.007}$	$0.841_{\pm 0.006}$	$0.924_{\pm 0.007}$
	`uCBOpt` ($\vartheta \to 0^+$)	$\mathbf{0.742}_{\pm 0.044}$	$0.187_{\pm 0.014}$	$0.868_{\pm 0.016}$	$0.820_{\pm 0.022}$	$0.918_{\pm 0.011}$
	`uCBOpt` ($\vartheta = 5 \cdot 10^{-5}$)	$\mathbf{0.756}_{\pm 0.007}$	$0.196_{\pm 0.004}$	$0.860_{\pm 0.004}$	$0.804_{\pm 0.006}$	$0.914_{\pm 0.003}$
	`uCBOpt-adapt`	$\mathbf{0.763}_{\pm 0.041}$	$0.204_{\pm 0.017}$	$0.854_{\pm 0.019}$	$0.791_{\pm 0.026}$	$0.911_{\pm 0.012}$
	`lCBOpt-adapt`	$0.771_{\pm 0.012}$	$0.196_{\pm 0.005}$	$0.859_{\pm 0.005}$	$0.806_{\pm 0.006}$	$0.912_{\pm 0.004}$
In: CIFAR-100 **Out:** TinyImageNet	SGD	$\mathbf{0.828}_{\pm 0.007}$	$0.315_{\pm 0.003}$	$0.744_{\pm 0.003}$	$0.767_{\pm 0.003}$	$0.705_{\pm 0.004}$
	IVON@mean	$0.855_{\pm 0.001}$	$0.336_{\pm 0.002}$	$0.718_{\pm 0.002}$	$0.744_{\pm 0.001}$	$0.677_{\pm 0.003}$
	IVON	$0.838_{\pm 0.007}$	$0.321_{\pm 0.003}$	$0.739_{\pm 0.005}$	$0.765_{\pm 0.003}$	$0.698_{\pm 0.005}$
	MC-D	$0.839_{\pm 0.005}$	$0.316_{\pm 0.005}$	$0.738_{\pm 0.005}$	$0.762_{\pm 0.006}$	$0.697_{\pm 0.004}$
	`uCBOpt` ($\vartheta \to 0^+$)	$\mathbf{0.834}_{\pm 0.009}$	$0.319_{\pm 0.004}$	$0.739_{\pm 0.004}$	$0.762_{\pm 0.003}$	$0.700_{\pm 0.004}$
	`uCBOpt` ($\vartheta = 5 \cdot 10^{-5}$)	$\mathbf{0.832}_{\pm 0.006}$	$0.318_{\pm 0.003}$	$0.739_{\pm 0.004}$	$0.762_{\pm 0.002}$	$0.701_{\pm 0.003}$
	`uCBOpt-adapt`	$\mathbf{0.828}_{\pm 0.002}$	$0.312_{\pm 0.004}$	$0.746_{\pm 0.002}$	$0.769_{\pm 0.005}$	$0.707_{\pm 0.002}$
	`lCBOpt-adapt`	$0.841_{\pm 0.002}$	$0.326_{\pm 0.003}$	$0.732_{\pm 0.002}$	$0.757_{\pm 0.001}$	$0.692_{\pm 0.001}$

J Ablation Studies

J.1 Varying ϑ in `uCBOpt`

We perform an ablation over the fixed candidate's curvature parameter, denoted by ϑ in `uCBOpt`. The $\vartheta \to 0^+$ configuration is practically implemented as $\vartheta = 0$. The model is a LeNet trained on Fashion-MNIST. The results are summarized in the following Table 9.

We observe that `uCBOpt` is relatively stable across a broad range of ϑ values, with only small variations in in-domain accuracy and calibration metrics. However, the choice of ϑ has a more noticeable effect on OOD detection performance. In particular, intermediate values around $\vartheta = 8 \cdot 10^{-6}$ achieve the strongest overall OOD performance, yielding the lowest FPR@95 and highest AUROC and AUPR-Out. Very small or larger values of ϑ tend to slightly degrade OOD detection quality, suggesting that an appropriate positive curvature floor can improve uncertainty estimation without significantly affecting in-domain predictive performance.

Table 9: Effect of the ϑ parameter in `uCBOpt` on performance averaged over three runs.

	In-domain			Out-of-domain		
ϑ	Acc. ↑	NLL ↓	ECE ↓	FPR@95 ↓	AUROC ↑	AUPR-Out ↑
0^+	$0.905_{\pm 0.002}$	$0.269_{\pm 0.004}$	$0.020_{\pm 0.006}$	$0.776_{\pm 0.025}$	$0.796_{\pm 0.014}$	$0.867_{\pm 0.010}$
$1 \cdot 10^{-7}$	$0.905_{\pm 0.002}$	$0.267_{\pm 0.001}$	$0.019_{\pm 0.006}$	$0.777_{\pm 0.019}$	$0.786_{\pm 0.011}$	$0.862_{\pm 0.006}$
$2 \cdot 10^{-7}$	$0.905_{\pm 0.002}$	$0.271_{\pm 0.002}$	$0.021_{\pm 0.007}$	$0.762_{\pm 0.006}$	$0.799_{\pm 0.009}$	$0.871_{\pm 0.005}$
$5 \cdot 10^{-7}$	$0.906_{\pm 0.005}$	$0.272_{\pm 0.006}$	$0.021_{\pm 0.004}$	$0.780_{\pm 0.041}$	$0.786_{\pm 0.027}$	$0.863_{\pm 0.020}$
$8 \cdot 10^{-7}$	$0.905_{\pm 0.001}$	$0.269_{\pm 0.005}$	$0.021_{\pm 0.002}$	$0.802_{\pm 0.039}$	$0.781_{\pm 0.024}$	$0.856_{\pm 0.019}$
$1 \cdot 10^{-6}$	$0.907_{\pm 0.002}$	$0.270_{\pm 0.002}$	$0.023_{\pm 0.000}$	$0.778_{\pm 0.020}$	$0.796_{\pm 0.015}$	$0.868_{\pm 0.010}$
$2 \cdot 10^{-6}$	$0.906_{\pm 0.003}$	$0.273_{\pm 0.003}$	$0.021_{\pm 0.002}$	$0.799_{\pm 0.030}$	$0.773_{\pm 0.031}$	$0.852_{\pm 0.020}$
$5 \cdot 10^{-6}$	$0.908_{\pm 0.004}$	$0.267_{\pm 0.002}$	$0.020_{\pm 0.006}$	$0.766_{\pm 0.019}$	$0.795_{\pm 0.012}$	$0.867_{\pm 0.009}$
$8 \cdot 10^{-6}$	$0.906_{\pm 0.003}$	$0.263_{\pm 0.006}$	$0.018_{\pm 0.006}$	$0.739_{\pm 0.026}$	$0.810_{\pm 0.012}$	$0.879_{\pm 0.010}$
$1 \cdot 10^{-5}$	$0.907_{\pm 0.003}$	$0.269_{\pm 0.002}$	$0.020_{\pm 0.005}$	$0.796_{\pm 0.014}$	$0.777_{\pm 0.009}$	$0.855_{\pm 0.007}$

J.2 Varying γ and β_3 in `uCBOpt-adapt`

We perform an ablation over the γ and β_3 parameters in `uCBOpt-adapt`. The model is a LeNet trained on Fashion-MNIST. The results are summarized in the following Table 10 and Table 11.

For the γ ablation, we observe that moderate-to-large values of γ generally improve in-domain performance, with $\gamma = 0.9$ achieving the best overall trade-off between accuracy, NLL, and calibration. However, when γ becomes too large ($\gamma = 0.95$), performance deteriorates substantially, indicating that overly aggressive curvature correction can destabilize optimization. We also observe that OOD performance is relatively stable across most γ values, though the best OOD metrics are obtained at smaller γ values.

For the β_3 ablation, the results are comparatively stable across a broad range of values, suggesting that `uCBOpt-adapt` is not highly sensitive to the precise choice of β_3. Nevertheless, $\beta_3 = 1.001$ achieves the strongest overall performance, providing the best balance between in-domain accuracy, calibration, and OOD detection metrics.

Table 10: Effect of the γ parameter in `uCBOpt-adapt` on performance averaged over three runs.

	In-domain			Out-of-domain		
γ	Acc. ↑	NLL ↓	ECE ↓	FPR@95 ↓	AUROC ↑	AUPR-Out ↑
0.0	$0.905_{\pm 0.002}$	$0.269_{\pm 0.004}$	$0.020_{\pm 0.006}$	$0.776_{\pm 0.025}$	$0.796_{\pm 0.014}$	$0.867_{\pm 0.010}$
0.05	$0.907_{\pm 0.002}$	$0.268_{\pm 0.004}$	$0.021_{\pm 0.003}$	$0.787_{\pm 0.013}$	$0.790_{\pm 0.013}$	$0.862_{\pm 0.008}$
0.1	$0.903_{\pm 0.002}$	$0.276_{\pm 0.004}$	$0.022_{\pm 0.001}$	$0.812_{\pm 0.014}$	$0.753_{\pm 0.003}$	$0.841_{\pm 0.004}$
0.3	$0.905_{\pm 0.001}$	$0.273_{\pm 0.007}$	$0.022_{\pm 0.004}$	$0.787_{\pm 0.011}$	$0.778_{\pm 0.009}$	$0.858_{\pm 0.006}$
0.5	$0.904_{\pm 0.001}$	$0.266_{\pm 0.004}$	$0.017_{\pm 0.003}$	$0.791_{\pm 0.012}$	$0.769_{\pm 0.010}$	$0.853_{\pm 0.007}$
0.7	$0.909_{\pm 0.002}$	$0.265_{\pm 0.004}$	$0.025_{\pm 0.002}$	$0.798_{\pm 0.008}$	$0.773_{\pm 0.016}$	$0.855_{\pm 0.009}$
0.9	$0.910_{\pm 0.001}$	$0.256_{\pm 0.003}$	$0.016_{\pm 0.001}$	$0.783_{\pm 0.009}$	$0.773_{\pm 0.015}$	$0.857_{\pm 0.009}$
0.95	$0.832_{\pm 0.005}$	$0.460_{\pm 0.011}$	$0.017_{\pm 0.009}$	$0.722_{\pm 0.134}$	$0.743_{\pm 0.092}$	$0.849_{\pm 0.056}$

Table 11: Effect of the β_3 parameter in `uCBOpt-adapt` on performance averaged over three runs.

	In-domain			Out-of-domain		
β_3	Acc. ↑	NLL ↓	ECE ↓	FPR@95 ↓	AUROC ↑	AUPR-Out ↑
1.00001	$0.908_{\pm 0.002}$	$0.256_{\pm 0.008}$	$0.017_{\pm 0.003}$	$0.803_{\pm 0.007}$	$0.767_{\pm 0.019}$	$0.851_{\pm 0.009}$
1.0001	$0.909_{\pm 0.002}$	$0.256_{\pm 0.001}$	$0.020_{\pm 0.001}$	$0.797_{\pm 0.001}$	$0.753_{\pm 0.006}$	$0.847_{\pm 0.003}$
1.001	$0.910_{\pm 0.001}$	$0.256_{\pm 0.003}$	$0.016_{\pm 0.001}$	$0.783_{\pm 0.009}$	$0.773_{\pm 0.015}$	$0.857_{\pm 0.009}$
1.01	$0.909_{\pm 0.002}$	$0.256_{\pm 0.003}$	$0.018_{\pm 0.003}$	$0.792_{\pm 0.012}$	$0.771_{\pm 0.013}$	$0.853_{\pm 0.007}$
1.1	$0.909_{\pm 0.002}$	$0.256_{\pm 0.003}$	$0.018_{\pm 0.003}$	$0.792_{\pm 0.012}$	$0.771_{\pm 0.013}$	$0.853_{\pm 0.007}$

J.3 Varying γ and β_3 in `lCBOpt-adapt`

We perform an ablation over the γ and β_3 parameters in `lCBOpt-adapt`. The model is a LeNet trained on Fashion-MNIST. The results are summarized in the following Table 12 and Table 13.

For the γ ablation, we observe that `lCBOpt-adapt` is sensitive to the choice of γ. Very small values close to 1 (e.g., $\gamma = 1.005$) lead to substantially degraded optimization performance, while moderate values between 1.02 and 1.10 provide consistently strong in-domain and OOD results. In particular, $\gamma = 1.02$ achieves the best overall in-domain performance, whereas slightly larger values such as $\gamma = 1.05$ and $\gamma = 1.10$ provide stronger OOD detection metrics. Larger γ values beyond this range gradually reduce performance.

For the β_3 ablation, all metrics remain identical because every tested value satisfies $\beta_3 \le \beta_2 = 0.99999$. Indeed, $\tilde{h}_t = \beta_2 \tilde{h}_{t-1} + (1-\beta_2)(\hat{g}_t^2 + \delta) \ge \beta_2 \tilde{h}_{t-1}$, so the initialization $c_0 = \tilde{h}_0$ and the recursion in Alg. 3 imply $c_t = \tilde{h}_t$ by induction. The denominator is therefore $(\gamma-1)\tilde{h}_t + \epsilon$. Holding other settings fixed, this is exactly the $\vartheta = 0$ `uCBOpt` update with learning rate $\rho_t/(\gamma-1)$ and an added stability constant $\epsilon/(\gamma-1)$. Thus, the γ ablation also changes the effective learning rate, and these experiments do not isolate a distinct adaptive-envelope effect. Nontrivial envelope memory requires $\beta_3 > \beta_2$ in this recursion.

Table 12: Effect of the γ parameter in `lCBOpt-adapt` on performance averaged over three runs.

	In-domain			Out-of-domain		
γ	Acc. ↑	NLL ↓	ECE ↓	FPR@95 ↓	AUROC ↑	AUPR-Out ↑
1.005	$0.838_{\pm 0.009}$	$0.460_{\pm 0.036}$	$0.020_{\pm 0.005}$	$0.736_{\pm 0.021}$	$0.744_{\pm 0.016}$	$0.850_{\pm 0.007}$
1.01	$0.909_{\pm 0.002}$	$0.253_{\pm 0.003}$	$0.017_{\pm 0.001}$	$0.804_{\pm 0.019}$	$0.767_{\pm 0.019}$	$0.849_{\pm 0.013}$
1.02	$0.913_{\pm 0.002}$	$0.252_{\pm 0.004}$	$0.020_{\pm 0.000}$	$0.794_{\pm 0.016}$	$0.784_{\pm 0.019}$	$0.859_{\pm 0.011}$
1.03	$0.911_{\pm 0.001}$	$0.262_{\pm 0.001}$	$0.024_{\pm 0.002}$	$0.795_{\pm 0.006}$	$0.780_{\pm 0.010}$	$0.857_{\pm 0.004}$
1.05	$0.908_{\pm 0.001}$	$0.267_{\pm 0.007}$	$0.026_{\pm 0.005}$	$0.786_{\pm 0.012}$	$0.796_{\pm 0.014}$	$0.865_{\pm 0.009}$
1.07	$0.907_{\pm 0.002}$	$0.272_{\pm 0.005}$	$0.022_{\pm 0.005}$	$0.800_{\pm 0.011}$	$0.769_{\pm 0.002}$	$0.851_{\pm 0.004}$
1.10	$0.906_{\pm 0.002}$	$0.268_{\pm 0.003}$	$0.023_{\pm 0.004}$	$0.771_{\pm 0.025}$	$0.795_{\pm 0.015}$	$0.868_{\pm 0.010}$
1.20	$0.907_{\pm 0.002}$	$0.272_{\pm 0.006}$	$0.020_{\pm 0.006}$	$0.793_{\pm 0.027}$	$0.790_{\pm 0.019}$	$0.862_{\pm 0.013}$
1.50	$0.904_{\pm 0.003}$	$0.274_{\pm 0.005}$	$0.019_{\pm 0.001}$	$0.810_{\pm 0.006}$	$0.775_{\pm 0.005}$	$0.852_{\pm 0.004}$
2.0	$0.901_{\pm 0.002}$	$0.282_{\pm 0.005}$	$0.014_{\pm 0.002}$	$0.804_{\pm 0.028}$	$0.763_{\pm 0.011}$	$0.847_{\pm 0.009}$

Table 13: Effect of the β_3 parameter in `lCBOpt-adapt` on performance averaged over three runs.

	In-domain			Out-of-domain		
β_3	Acc. ↑	NLL ↓	ECE ↓	FPR@95 ↓	AUROC ↑	AUPR-Out ↑
0.9	$0.913_{\pm 0.002}$	$0.252_{\pm 0.004}$	$0.020_{\pm 0.000}$	$0.794_{\pm 0.016}$	$0.784_{\pm 0.019}$	$0.859_{\pm 0.011}$
0.99	$0.913_{\pm 0.002}$	$0.252_{\pm 0.004}$	$0.020_{\pm 0.000}$	$0.794_{\pm 0.016}$	$0.784_{\pm 0.019}$	$0.859_{\pm 0.011}$
0.999	$0.913_{\pm 0.002}$	$0.252_{\pm 0.004}$	$0.020_{\pm 0.000}$	$0.794_{\pm 0.016}$	$0.784_{\pm 0.019}$	$0.859_{\pm 0.011}$
0.9999	$0.913_{\pm 0.002}$	$0.252_{\pm 0.004}$	$0.020_{\pm 0.000}$	$0.794_{\pm 0.016}$	$0.784_{\pm 0.019}$	$0.859_{\pm 0.011}$
0.99999	$0.913_{\pm 0.002}$	$0.252_{\pm 0.004}$	$0.020_{\pm 0.000}$	$0.794_{\pm 0.016}$	$0.784_{\pm 0.019}$	$0.859_{\pm 0.011}$

J.4 Full-Hessian `CBOpt` for Regression Tasks

To further evaluate the update rule derived in Proposition 3, we consider a full-Hessian implementation of CBOpt on 10 regression datasets. Unlike the practical implementation in Section 5.2, which uses diagonal squared-gradient and EMA approximations, this experiment computes the full Hessian of the regularized loss at each update.

We use a small neural network with one hidden layer of two neurons, taking five PCA components as input (15 trainable parameters). We minimize half the mean squared error with quadratic regularization of strength $\delta = 0.02$. Missing feature values are imputed using dataset-wide medians, and categorical features are one-hot encoded before splitting. For each run, we randomly split the data into 60% training and 40% testing. Feature standardization, PCA, and target standardization are fitted on the training partition and applied to the test partition. In the contamination experiment, we randomly select up to 80 training observations, then corrupt the targets of the 30% with the largest feature norms by adding randomly signed perturbations of six training-target standard deviations. In the under-representation experiment, we identify observations above the 80th percentile of the training targets, retain 20% of these observations, and select additional non-extreme observations up to a nominal training size of 80, prioritizing the retained extreme observations. The test sets remain unchanged. Results are averaged over 20 runs with different data splits, training subsets, and parameter initializations.

We compare `lCBOpt` and `uCBOpt` against Adam, Bayes by Backprop (BBB) (Blundell et al., 2015), VOGN, and Vadam (Khan et al., 2018). All methods share an Adam warm start with learning rate 0.015, using 100 steps for contamination and 20 steps for under-representation, followed by 60 method-specific updates. The learning rates for Adam, BBB, VOGN, and Vadam are 0.008, 0.008, 0.02, and 0.008, respectively, while both CBOpt variants use 0.2 for contamination and 0.3 for under-representation. BBB uses a mean-field Gaussian approximation with prior precision 0.02, 16 antithetic samples, and a KL term divided by the training-set size. VOGN uses a diagonal empirical-Fisher approximation with momentum 0.99, while Vadam uses $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$. For `CBOpt`, we compute the full Hessian H of the regularized loss and select candidate precisions $\vartheta_{\mathrm{u}} = \max\{10^{-4}, \lambda_{\min}(H) - 0.05\}$ and $\vartheta_{\mathrm{l}} = \lambda_{\max}(H) + 0.05$. The corresponding curvature matrices are symmetrized, shifted to ensure a minimum eigenvalue of at least 10^{-4}, and damped by $10^{-4}I$. Updates for `CBOpt`, VOGN, and Vadam are clipped to a maximum Euclidean norm of 0.12.

Targeted label contamination. We first evaluate robustness to targeted label contamination. We select 30% of the training samples with the largest feature norms and corrupt their target values by

Table 14: Test RMSE under 30% targeted label contamination (averaged over 20 runs). Mean ranks are computed across all runs and datasets.

Dataset	Adam	BBB	VOGN	Vadam	lCBOpt	uCBOpt
Concrete	$25.04_{\pm 3.61}$	$25.53_{\pm 3.80}$	$23.52_{\pm 3.05}$	$23.48_{\pm 3.08}$	$\mathbf{23.05}_{\pm 3.42}$	$28.40_{\pm 4.46}$
Airfoil	$9.37_{\pm 1.66}$	$9.63_{\pm 1.62}$	$8.87_{\pm 1.37}$	$8.75_{\pm 1.30}$	$\mathbf{8.54}_{\pm 1.19}$	$12.19_{\pm 2.07}$
Real Estate	$18.88_{\pm 4.02}$	$19.26_{\pm 4.00}$	$17.36_{\pm 3.68}$	$17.09_{\pm 3.59}$	$\mathbf{16.76}_{\pm 3.82}$	$22.51_{\pm 4.07}$
Power Plant	$18.23_{\pm 3.97}$	$19.00_{\pm 4.04}$	$16.86_{\pm 4.03}$	$16.74_{\pm 3.97}$	$\mathbf{15.95}_{\pm 3.68}$	$23.18_{\pm 3.47}$
Energy Heating	$13.25_{\pm 2.42}$	$13.61_{\pm 2.39}$	$12.14_{\pm 2.05}$	$12.08_{\pm 2.07}$	$\mathbf{11.25}_{\pm 2.25}$	$15.73_{\pm 3.79}$
Auto MPG	$9.84_{\pm 3.17}$	$10.13_{\pm 3.25}$	$8.95_{\pm 2.93}$	$8.83_{\pm 2.93}$	$\mathbf{8.18}_{\pm 2.66}$	$11.78_{\pm 4.19}$
Automobile	$1.96_{\pm 0.37}$	$1.99_{\pm 0.38}$	$1.81_{\pm 0.35}$	$1.76_{\pm 0.33}$	$\mathbf{1.67}_{\pm 0.33}$	$2.43_{\pm 0.62}$
Computer Hardware	$217.63_{\pm 58.41}$	$222.12_{\pm 58.42}$	$200.18_{\pm 50.74}$	$199.63_{\pm 49.17}$	$\mathbf{190.98}_{\pm 51.02}$	$262.46_{\pm 73.64}$
Wine Red	$1.21_{\pm 0.15}$	$1.24_{\pm 0.16}$	$1.14_{\pm 0.15}$	$1.14_{\pm 0.14}$	$\mathbf{1.11}_{\pm 0.15}$	$1.48_{\pm 0.26}$
Diabetes	$114.52_{\pm 17.74}$	$116.65_{\pm 17.16}$	$107.46_{\pm 16.80}$	$105.92_{\pm 15.84}$	$\mathbf{100.45}_{\pm 15.89}$	$133.64_{\pm 21.74}$
Mean Rank ↓	3.97	4.95	2.54	2.29	**1.48**	5.79

Table 15: Large-target MAE under under-representation of extreme cases (averaged over 20 runs). Mean ranks are computed across all runs and datasets.

Dataset	Adam	BBB	VOGN	Vadam	lCBOpt	uCBOpt
Concrete	$12.31_{\pm 1.93}$	$\mathbf{12.14}_{\pm 1.76}$	$14.48_{\pm 1.60}$	$16.12_{\pm 1.80}$	$12.94_{\pm 2.15}$	$12.74_{\pm 1.90}$
Airfoil	$3.85_{\pm 0.43}$	$3.82_{\pm 0.46}$	$4.65_{\pm 0.64}$	$4.99_{\pm 0.81}$	$4.13_{\pm 0.59}$	$\mathbf{3.62}_{\pm 0.43}$
Real Estate	$11.74_{\pm 1.69}$	$11.67_{\pm 1.81}$	$14.09_{\pm 2.02}$	$15.29_{\pm 2.75}$	$13.36_{\pm 1.83}$	$\mathbf{11.50}_{\pm 1.65}$
Power Plant	$3.97_{\pm 0.16}$	$4.07_{\pm 0.20}$	$3.95_{\pm 0.12}$	$5.71_{\pm 2.42}$	$4.22_{\pm 0.27}$	$\mathbf{3.70}_{\pm 0.87}$
Energy Heating	$4.08_{\pm 0.61}$	$\mathbf{3.85}_{\pm 0.54}$	$5.33_{\pm 0.71}$	$7.13_{\pm 1.44}$	$5.06_{\pm 0.83}$	$4.44_{\pm 1.16}$
Auto MPG	$4.19_{\pm 0.85}$	$\mathbf{3.93}_{\pm 0.78}$	$6.12_{\pm 0.82}$	$7.43_{\pm 1.12}$	$5.93_{\pm 0.78}$	$4.09_{\pm 0.89}$
Automobile	$1.47_{\pm 0.18}$	$1.47_{\pm 0.18}$	$1.67_{\pm 0.14}$	$1.74_{\pm 0.15}$	$1.64_{\pm 0.16}$	$\mathbf{1.47}_{\pm 0.19}$
Computer Hardware	$146.49_{\pm 42.23}$	$151.94_{\pm 43.89}$	$194.10_{\pm 47.86}$	$210.56_{\pm 52.00}$	$207.07_{\pm 41.17}$	$\mathbf{131.75}_{\pm 37.68}$
Wine Red	$0.36_{\pm 0.02}$	$0.37_{\pm 0.02}$	$0.37_{\pm 0.02}$	$0.37_{\pm 0.03}$	$0.36_{\pm 0.03}$	$\mathbf{0.36}_{\pm 0.02}$
Diabetes	$69.92_{\pm 9.80}$	$68.81_{\pm 10.19}$	$82.76_{\pm 9.13}$	$88.96_{\pm 9.90}$	$78.67_{\pm 10.44}$	$\mathbf{68.62}_{\pm 10.29}$
Mean Rank ↓	2.54	2.44	4.39	5.54	4.03	**2.06**

adding or subtracting six times the standard deviation of the training targets, with the sign chosen randomly for each sample. Evaluation is performed on the clean test set. As shown in Table 14, `lCBOpt` achieves the lowest test RMSE across all 10 datasets, outperforming Adam and the probabilistic variational baselines. It also obtains the best mean rank of 1.48 across all datasets and runs, suggesting that the lower CBO update can be effective under targeted label contamination.

Under-representation of extreme cases. Next, we investigate the effect of under-representing extreme target values during training. We define large-target observations as those whose targets exceed the 80th percentile of the original training targets, and retain only 20% of these observations. The test distribution remains unchanged. We evaluate large-target mean absolute error (LT-MAE), computed over test observations whose targets exceed the same threshold. As shown in Table 15, `uCBOpt` achieves the lowest average LT-MAE on 7 of the 10 datasets and obtains the best mean LT-MAE rank of 2.06 across all datasets and runs. These findings suggest that the upper CBO formulation can be beneficial when extreme target values are under-represented during training.

Overall, these experiments provide complementary empirical evidence for the effectiveness of full-Hessian `CBOpt` without diagonal or EMA curvature approximations. The results also highlight the different behaviours of the lower and upper CBO formulations under the two regression settings, motivating further investigation into whether these findings generalize to broader settings.

K Additional Experiment: Combining CBOpt with MC Dropout

We further evaluate whether CBOpt can be combined with MC Dropout. The experiment is conducted on ResNet-20 trained on CIFAR-10, with SVHN used as the out-of-domain dataset. The results are reported in Table 16.

Combining CBOpt with MC Dropout generally preserves the strong in-domain performance of MC Dropout while improving several OOD detection metrics (bolded results). In particular, uCBOpt-adapt + MC-D achieves the best in-domain accuracy, NLL, and ECE among the reported methods, suggesting that the adaptive update remains beneficial even when stochastic dropout inference is used. For OOD detection, the CBOpt + MC-D variants improve AUROC and AUPR-Out over standard MC Dropout, with lCBOpt-adapt + MC-D achieving the highest AUROC and AUPR-Out. The fixed-ϑ uCBOpt variant also substantially reduces FPR@95, although its AUROC improvement is more modest. Overall, these results suggest that CBOpt and MC Dropout are complementary: MC Dropout improves calibration via stochastic inference, whereas the CBOpt update can further improve OOD separability.

Table 16: Experimental results for ResNet-20/CIFAR-10 with CBOpt combined with MC dropout (best validation model).

	In-domain			Out-of-domain		
Optimizer	Acc. ↑	NLL ↓	ECE ↓	FPR@95 ↓	AUROC ↑	AUPR-Out ↑
MC-D	$0.923_{\pm 0.002}$	$0.230_{\pm 0.006}$	$0.010_{\pm 0.002}$	$0.795_{\pm 0.053}$	$0.847_{\pm 0.024}$	$0.902_{\pm 0.017}$
uCBOpt ($\vartheta \to 0^+$) + MC-D	$0.921_{\pm 0.003}$	$0.236_{\pm 0.005}$	$0.010_{\pm 0.002}$	$0.797_{\pm 0.043}$	$0.859_{\pm 0.017}$	$0.906_{\pm 0.013}$
uCBOpt ($\vartheta = 5 \cdot 10^{-5}$) + MC-D	$0.921_{\pm 0.005}$	$0.239_{\pm 0.007}$	$0.010_{\pm 0.003}$	$0.796_{\pm 0.043}$	$0.854_{\pm 0.023}$	$0.903_{\pm 0.015}$
uCBOpt-adapt + MC-D	$0.924_{\pm 0.002}$	$0.228_{\pm 0.003}$	$0.007_{\pm 0.001}$	$0.787_{\pm 0.025}$	$0.861_{\pm 0.009}$	$0.908_{\pm 0.008}$
lCBOpt-adapt + MC-D	$0.918_{\pm 0.001}$	$0.241_{\pm 0.002}$	$0.010_{\pm 0.001}$	$0.776_{\pm 0.027}$	$0.867_{\pm 0.007}$	$0.911_{\pm 0.007}$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The theoretical contributions claimed in the abstract and introduction are provided in Section 4 and Section 5, while the experimental results are provided in Section 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work, pertaining to the choice of candidate family, under Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide the assumptions for our theorems alongside their statements, and the proofs in Appendix D. Due to limited space available, we are unable to provide an intuition in the main text.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We present experimental details in Appendix H and Appendix J.4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide access to our code at github.com/shelvia-w/cbopt.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide experimental details in Appendix H and Appendix J.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the mean and standard deviation over independent repeats in all the experimental result tables.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss the hardware, runtime and memory usage for classification experiments in Appendix H.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We predict no direct harmful consequences of our work.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There are potential societal consequences of our work, none which we feel must be specifically highlighted.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all the data, models, and methods used in our experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We include documentation of our code in a Markdown file, in the `CBOpt` repository.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.