EVIDENTIAL TURING PROCESSES

Anonymous authors
Paper under double-blind review

ABSTRACT

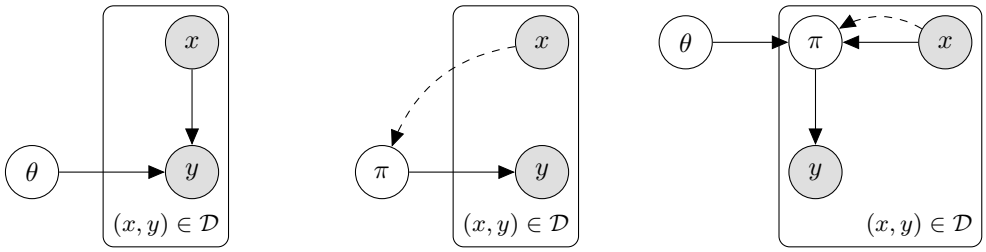
A probabilistic classifier with reliable predictive uncertainties i) fits successfully to the target domain data, ii) provides calibrated class probabilities in difficult regions of the target domain (e.g. class overlap), and iii) accurately identifies queries coming out of the target domain and reject them. We introduce an original combination of Evidential Deep Learning, Neural Processes, and Neural Turing Machines capable of providing all three essential properties mentioned above for total uncertainty quantification. We observe our method on three image classification benchmarks to consistently improve the in-domain uncertainty quantification, out-of-domain detection, and robustness against input data corruption with one single model. Our unified solution delivers an implementation-friendly and computationally efficient recipe for safety clearance and provides intellectual economy to an investigation of algorithmic roots of epistemic awareness in deep neural nets.

1 INTRODUCTION

Deep neural nets are the gold standard of modern machine learning in pattern classification. Despite their groundbreaking success in problems where there is abundant training data, the applicability of deep neural nets to safety-critical use cases such as autonomous driving or medical diagnostics is still a matter of fundamental research (Schwalbe & Schels, 2020). Key challenges in the development of total safety in deep learning are at least two-fold: i) reliable risk assessment in difficult regions of the target domain (e.g. class overlap) based on which safety-preserving fallback functions can be deployed, and ii) rejection of inputs that do not belong to the target domain, such as an image of a cat presented to a character recognition system.

The attention of the machine learning community on each of these critical safety elements is in steady increase (Naeini et al., 2015; Guo et al., 2017; Kuleshov et al., 2018). However, the developments follow isolated directions and the proposed solutions are largely fragmented. Calibration of neural net probabilities focuses primarily on post-hoc adjustment algorithms trained on validation sets from in-domain data (Guo et al., 2017), ignoring the out-of-domain detection aspect. On the other hand, recent advances in deep learning out-of-domain detection build on strong regularization mechanisms that strongly penalize divergence from the probability mass observed in the target domain (Sensoy et al., 2018; Malinin & Gales, 2018). The overlooked aspect here is the distortion of the class probabilities in the target domain as the models are encouraged to focus on detecting in- vs. out-of-domain data at the cost of proper in-distribution calibration. Such fragmentation of best practices not only hinders their accessibility by real-world applications but also complicates the scientific inquiry of the underlying reasons of sub-optimality of neural net uncertainties.

We aim to identify the guiding principles for Bayesian modeling that deliver the most complete set of uncertainty quantification capabilities in one single model. We first characterize Bayesian models with respect to the relationship of their global and local variables: (a) *Parametric Bayesian Models* comprise a likelihood function that maps inputs to outputs via probabilistic global parameters, (b) *Evidential Bayesian Models* apply an uninformative prior distribution on the parameters of the likelihood function and infer the prior hyperparameters by amortizing on an input observation. Investigating the decomposition of predictive distribution variance, we find out that the uncertainty characteristics of these two approaches exhibit complementary strengths. We introduce a third approach that employs a prior on the likelihood parameters that both conditions on the input observations on the true model and amortizes on them during inference. The mapping from the input observations to the prior hyperparameters is governed by a random set of global parameters. We refer to the resulting approach as (c) *Complete Bayesian Models*, as we find out that its predictive distribution variance combines the



(a) Parametric Bayesian Model (b) Evidential Bayesian Model (c) Complete Bayesian Model

Figure 1: Plate diagrams of three main Bayesian modeling approaches. Shaded nodes are observed and unshaded ones are latent. The solid direct arrows denote conditional dependencies in the true model, while the bent dashed arrows denote amortization in variational inference.

favorable properties of Parametric and Evidential Bayesian Models. Figure 1 gives an overview of the relationship between these three Bayesian modeling approaches.

The advantage of the Complete Bayesian Models come with the challenge of choosing an input-dependent prior, being the main technical difficulty we tackle in this work. We introduce a new stochastic process construction that addresses this problem by accumulating observations on a context set during training time into the parameters of a global hyperprior variable by an explicitly designed update rule. We design the input-specific prior on the likelihood parameters by conditioning it on both the hyperprior and the input observation. As conditioning via explicit parameter updates amounts to maintaining an external memory that can be queried by the prior distribution, we refer to the eventual stochastic process as a *Turing Process* with inspiration from the earlier work on Neural Turing Machines (Graves et al., 2014) and Neural Processes (Garnelo et al., 2018b). We arrive at our target Bayesian design that is equipped with the complete set of uncertainty quantification capabilities by incorporating the Turing Process design into a Complete Bayesian Model. We refer to the resulting approach as an *Evidential Turing Process*. We observe on three real-world image classification tasks that the Evidential Turing Process is the only model that is consistently either superior to or on par with the best-performing alternative approach in prediction error, model fit, in-domain uncertainty calibration, out-of-domain detection, and robustness to input data corruption.

2 Parametric, Evidential, and Complete Bayesian Models

We are given a set of samples $(x, y) \in \mathcal{D}$ assumed to be observed independently from an unknown distribution $(X, Y) \sim \mathcal{P}_{XY}$. We are interested in learning a probabilistic model $p(y|x)$ that can accurately predict the outputs y from inputs x with a calibrated prediction probability score

$$\mathbb{P}(Y = y \,|p(y|x)) = p(y|x), \qquad \forall\, p(y|x)., \tag{1}$$

deliver good fit on test data, and accurately distinguish between in-domain and out-of-domain samples based on its predictive entropy. We refer to performing all these tasks with a single model as complete uncertainty quantification.

The *Parametric Bayesian Model (PBM)* follows the data generating process

$$p(\mathcal{D}, \theta) = p(\theta) \prod_{(x,y)\in\mathcal{D}} p(y|\theta, x) \tag{2}$$

for a global parameter set θ and a likelihood function $p(y|\theta, x)$ that links the input space x to the output space y, a trivial-looking aspect that will become decisive below. The newly introduced *Evidential Bayesian Model (EBM)* (Sensoy et al., 2018) approach follows a data generating process

$$p(\mathcal{D}, \pi) = p(\pi) \prod_{(x,y)\in\mathcal{D}} p(y|\pi) \tag{3}$$

for a global latent variable π that governs the sufficient statistics of the likelihood function $p(y|\pi)$, which is *not a function of the input observations* x.

It is customary to infer the intractable posterior by minimizing the variational free energy subject to an approximate distribution. For the Parametric Bayesian, the true posterior $p(\theta|\mathcal{D})$ is approximated with $q_\lambda(\theta)$ by minimizing the functional

$$\mathcal{F}_{PBM}(\lambda) = \mathbb{E}_{q_\lambda \theta)} \left[-\sum_{(x,y)\in\mathcal{D}} \log p(y|\theta, x) + \log \frac{q_\lambda(\theta)}{p(\theta)} \right], \tag{4}$$

with respect to the variational parameters λ. The Evidential Bayesian Model establishes the link between the inputs and outputs via an amortized approximate posterior $q_\lambda(\pi|x)$, leading to the variational free energy functional below

$$\mathcal{F}_{EBM}(\lambda) = \sum_{(x,y)\in\mathcal{D}} \mathbb{E}_{q_\lambda(\pi|x)} \left[-\log p(y|\pi) + \log \frac{q_\lambda(\pi|x)}{p(\pi)} \right]. \tag{5}$$

While Eq. 4 averages the model fit and divergence from a prior guess over the whole sample set, Eq. 5 averages over an observation-specific distribution $q_\lambda(\pi|x)$.

The label of a new observation x_* is then predicted as $p(y_*|\mathcal{D}, x_*) \approx \int p(y_*|\theta, x_*) q_\lambda(\theta) d\theta$. The variance of this predictive distribution decomposes according to the law of total variance for the Parametric Bayesian Model as

$$\mathbb{V}_{PBM}\left[y_*|\mathcal{D}, x_*\right] = \mathbb{V}_{q_\lambda(\theta)}\left[\mathbb{E}\left[y_*|\theta, x_*\right]\right] + \mathbb{E}_{q_\lambda(\theta)}\left[\mathbb{V}\left[y_*|\theta, x_*\right]\right] \tag{6}$$

and for the Evidential Bayesian Model as

$$\mathbb{V}_{EBM}\left[y_*|\mathcal{D}, x_*\right] = \mathbb{V}_{q_\lambda(\pi_*|x_*)}\left[\mathbb{E}\left[y_*|\pi_*, x_*\right]\right] + \mathbb{E}_{q_\lambda(\pi_*|x_*)}\left[\mathbb{V}\left[y_*|\pi_*, x_*\right]\right]. \tag{7}$$

It is common to expect from this decomposition to provide two quantities: *i) Model uncertainty:* an expression on the limits of the knowledge of the modeler on the learning problem as a whole, comprising the effects of the discrepancy between assumed data generating process model and the true data distribution, as well as the error made in parameter estimation due to the limited sample size. *ii) Data uncertainty:* an expression on the observation-specific uncertainty caused by class overlap, measurement noise, or samples coming from another distribution.

Key observations. We base our subsequent modeling assumptions on the following observations:

i) The most expressive term for model uncertainty is $\mathbb{V}_{q_\lambda(\theta)}\left[\mathbb{E}\left[y_*|\theta, x_*\right]\right]$ in Eq. 6, as it quantifies the effects of global unknowns on the point prediction of the model.

ii) The most expressive term for data uncertainty is $\mathbb{E}_{q_\lambda(\pi_*|x_*)}\left[\mathbb{V}\left[y_*|\pi_*, x_*\right]\right]$ in Eq. 7, as it quantifies the effect of *only* the observation-specific unknowns on the observation-specific noise estimation of the model. This is in contrast to $\mathbb{E}_{q_\lambda(\theta)}\left[\mathbb{V}\left[y_*|\theta, x_*\right]\right]$, which does not provide an isolated view of data uncertainty from the global uncertainty effects.

iii) $\mathcal{F}_{PBM}$ permits an explicit modeler control on the model uncertainty by means of choosing the global expectation with respect to $q_\lambda(\theta)$ as well as the global regularizer $p(\theta)$.

iv) $\mathcal{F}_{EBM}$ permits an explicit modeler control on the data uncertainty by means of an observation-specific expectation with respect to $q_\lambda(\pi|x)$ as well as an observation-specific regularizer $p(\pi)$.

We introduce the *Complete Bayesian Model (CBM)* that inherits only the favorable properties of both parametric and evidential modeling practices:

$$p(y, \mathcal{D}, \pi, \theta) = p(\theta) \prod_{(x,y)\in\mathcal{D}} p(y|\pi) p(\pi|\theta, x). \tag{8}$$

For a structured (non-mean-field) and amortized approximate posterior $q_\lambda(\theta, \pi|x) = q_\lambda(\pi|\theta) q_\lambda(\theta)$, the variational free energy reads

$$\mathcal{F}_{CBM} = \mathbb{E}_{q_\lambda(\theta)} \left[-\sum_{(x,y)\in\mathcal{D}} \mathbb{E}_{q_\lambda(\pi|\theta,x)} \left[\log p(y|\pi) + \log \frac{q_\lambda(\pi|\theta, x)}{p(\pi|\theta, x)} \right] + \log \frac{q_\lambda(\theta)}{p(\theta)} \right]. \tag{9}$$

The posterior predictive distribution variance decomposes and nicely simplifies as

$$\begin{aligned}
&\mathbb{V}_{CBM}\left[y_*|\mathcal{D}, x_*\right] \\
&= \mathbb{V}_{q_\lambda(\theta)}\left[\mathbb{E}_{q_\lambda(\pi|\theta,x)}\left[\mathbb{E}\left[y|\pi,\theta\right]\right]\right] + \mathbb{E}_{q_\lambda(\theta,\pi|x)}\left[\mathbb{V}\left[y|\pi\right]\right] + \mathbb{E}_{q(\theta)}\left[\mathbb{V}_{q_\lambda(\pi|\theta,x)}\left[\mathbb{E}\left[y|\pi\right]\right]\right] \\
&= \underbrace{\mathbb{V}_{q_\lambda(\theta)}\left[\mathbb{E}\left[y|\theta,x\right]\right]}_{\text{Model uncertainty}} + \underbrace{\mathbb{E}_{q_\lambda(\pi|x)}\left[\mathbb{V}\left[y|\pi\right]\right]}_{\text{Data uncertainty}} + \underbrace{\mathbb{E}_{q_\lambda(\theta)}\left[\mathbb{V}_{q_\lambda(\pi|\theta,x)}\left[\mathbb{E}\left[y|\pi\right]\right]\right]}_{\text{Interaction term}},
\end{aligned} \tag{10}$$

which maintains the model uncertainty term of Eq. 6 that accounts for only global effects, the data uncertainty term of Eq. 7 that accounts for only observation-specific effects on predictor variance, and a new third term that makes the interaction between the two uncertainties explicit.

Main Hypothesis. *A Bayesian model which is capable of quantifying both uncertainties in isolation, as well as their interaction explicitly is a guiding principle for predictors that can succeed in in-domain calibration and out-of-domain detection simultaneously.*

3 Evidential Turing Processes as Complete Bayesian Models

While the Complete Bayesian Model provides the decomposition of uncertainties we are interested in, it is not yet ready to use. Both the Parametric Bayesian Model and the Evidential Bayesian Model use their priors as independently chosen regularizers. The nice decomposition in Eq. 10 is possible only when the prior can sensibly depend on an input observation. Below we formulate this bottleneck as the main technical problem of this paper and summarize the essence of our solution.

Main Problem. *The Complete Bayesian Model assumes an input-specific prior $p(\pi|\theta, x)$ in Eq. 8. However, there does not exist an established method for choosing such a prior.*

Our Solution Proposal. *We conjecture that a prior can effectively depend on an input if it receives context information during training time, stores it into an external memory, and retrieves relevant parts of it using the conditioned input example as a query.*

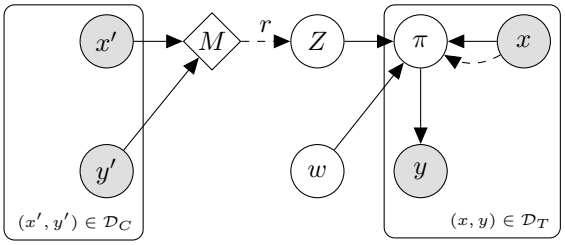
We initially build the mathematical framework that allows us to incorporate training-time observations into the prior hyperparameters on the fly. We start from the line of thought introduced for Neural Processes (Garnelo et al., 2018b) which designs a stochastic process by integrating out the global parameters of a Parametric Bayesian Model. Our ultimate target is to have a stand-alone prior design that *learns* to generate prior beliefs on individual samples at test time. This is possible if the parameters of the prior can be updated during the training time based on training observations but without introducing dependency on a context set during test time. We equip the Neural Process definition with a prior which can be conditioned on a context set on demand by updating its hyperparameters via an explicitly designed context-dependent update rule. Inspired from the notion of Neural Turing Machines (Graves et al., 2014), we call the resultant system of random variables a *Turing Process* and define it more formally as below.

Definition 1. *A Turing Process is a collection of random variables $Y = \{y_i | i \in \mathcal{I}\}$ for an index set $\mathcal{I} = \{1, \ldots, K\}$ with arbitrary $K \in \mathbb{N}^+$ that satisfy the two properties*

$$\textit{(i)}\ p_M(Y) = \int \prod_{y_i \in Y} p_M(y_i|\theta) p_M(\theta) d\theta, \quad \textit{(ii)}\ p_{M'}(Y|Y') = \int \prod_{y_i \in Y} p(y_i|\theta) p_{M'}(\theta|Y') d\theta,$$

for another random variable collection Y' of arbitrary size that lives in the same probability space as Y, a probability measure $p_M(\theta)$ with free parameters M and some function $M' = r(M, Y')$.

The Turing Process can express all stochastic processes since Property (i) is sufficient to satisfy the Kolmogorov's Extension Theorem (Øksendal, 1992) and choosing r simply to be an identity mapping would revert us back to the generic definition of a stochastic process. The Turing Process goes further by permitting to explicitly specify how information accumulates within the prior. This is a different approach from the Neural Process that performs conditioning $p(Y|Y')$ by applying an amortized posterior $q(\theta|Y')$ on a plain stochastic process that satisfies only Property (i), hence maintains a data-agnostic prior $p(\theta)$ and depends on a context set also at the prediction time.



(a) ETP Plate diagram

while *Model not converged* **do**
  for $\mathcal{D}_{batch} \subset \mathcal{D}$ **do**
    Choose $\mathcal{D}_C \subset \mathcal{D}_{batch}$
    $M \leftarrow$
    $\mathbb{E}_{Z \sim p_M(Z)} \left[r(Z, \mathcal{D}_C) \right]$
    $\lambda \leftarrow \lambda - \nabla_\lambda \mathcal{F}_{ETP}(\lambda)$
  end
end

(b) ETP Training

Figure 2: The Evidential Turing Process: *(left)* The essential design choice is how $p(\pi|Z, w, x)$ is constructed thanks to an external memory M that can be queried with respect to any input x without needing to store context data at test time. The dashed arrow indicates that Z can condition on a context $\mathcal{D}_C$ by updates its parameters M with an explicit function r. *(right)* The ETP training routine.

We address the main problem by incorporating a Turing Process design into a Complete Bayesian Model. Given a disjoint partitioning of the data $\mathcal{D} = \mathcal{D}_C \cup \mathcal{D}_T$ into a context set $(x', y') \in \mathcal{D}_C$ and a target set $(x, y) \in \mathcal{D}_T$, we propose the data generating process on the target set below

$$p(y, \mathcal{D}_T, \pi, \theta) = p(w) \underbrace{p_M(Z)}_{\text{Memory}} \prod_{(x,y) \in \mathcal{D}_T} \{ p(y|\pi) \underbrace{p(\pi|Z, w, x)}_{\substack{\text{Input-specific} \\ \text{prior}}} \}. \tag{11}$$

We decouple the global parameters $\theta = \{w, Z\}$ into w that parameterizes a mapping from the input to the hyperparameter space and an external memory $Z = \{z_1, \ldots, z_R\}$ governed by the distribution $p_M(Z)$ parameterized by M that stores a fixed-sized summary of the context data observed during training. The memory variable Z can update its belief conditioned on the context set $\mathcal{D}_C$ by updating its parameters with an explicit rule

$$p_M(Z|\mathcal{D}_C) \leftarrow p_{\mathbb{E}_{Z \sim p_M(Z)}[r(Z, \mathcal{D}_C)]}(Z). \tag{12}$$

The hyperpriors of $p(\pi|Z, w, x)$ are then determined by querying the memory Z for each input observation x using an attention mechanism, for instance a transformer network (Vaswani et al., 2017). Choosing the variational distribution as $q_\lambda(\theta, \pi|x) = q_\lambda(w) p_M(Z) q(\pi|Z, w, x)$ and plugging Eq. 11 into Eq. 9, we attain the variational free energy formula for our target model

$$\mathcal{F}_{ETP}(\lambda) = \tag{13}$$
$$\mathbb{E}_{q_\lambda(w)} \left[-\sum_{(x,y) \in \mathcal{D}} \mathbb{E}_{q_\lambda(\pi|Z,w,x)} \left[\mathbb{E}_{p_M(Z)} \left[\log p(y|\pi) + \log \frac{q_\lambda(\pi|Z, w, x)}{p(\pi|Z, w, x)} \right] \right] + \log \frac{q_\lambda(w)}{p(w)} \right].$$

The learning procedure will then alternate between updating the variational parameters λ via gradient-descent on $\mathcal{F}_{ETP}(\lambda)$ and updating the parameters of the memory variable Z via the update rule in Eq. 12 as summarized in the algorithm in Figure 2b. An Evidential Turing Process trained on a data set $\mathcal{D}$ can predict a test sample (x_*, y_*) by only querying the input x_* on the memory Z and evaluating the prior $p(\pi_*|Z, w, x_*)$ without need for any context set as

$$p(y_*|\mathcal{D}, x_*) \approx \iint p(y_*|\pi_*) p(\pi_*|Z, w, x_*) p_M(Z) p(w) dZ dw. \tag{14}$$

4 Most Important Special Cases of the Evidential Turing Process

Bayesian Neural Net (BNN) (Neal, 1995; MacKay, 1992; 1995) is the most characteristic example for Parametric Bayesian Models in the machine learning domain. Its state-of-the-art variants (Molchanov et al., 2017) are also in commonplace use. The BNN parameterizes its likelihood $p(y|\theta, x)$ with a neural network $f_\theta(\cdot)$, whose weights follow a distribution $\theta \sim p(\theta)$. We adopt the following data generating process for classification

$$\theta \sim \mathcal{N}(\theta|0, \beta^{-1} I), \qquad y|\theta, x \sim \mathcal{C}at\big(y|f_\theta(x)\big) \quad \forall\, (x, y) \in \mathcal{D}. \tag{15}$$

and infer its posterior by mean-field variational posterior $q_\lambda(\theta)$ applying the reparameterization trick (Kingma et al., 2015; Molchanov et al., 2017).

Evidential Deep Learning (EDL) (Sensoy et al., 2018) is the quintessential example of the family of Evidential Bayesian Models. EDL employs an uninformative Dirichlet prior on the class probabilities, which then set the mean of a Normal distribution on the one-hot-coded class labels

$$\pi \sim \mathcal{D}ir(\pi|1, \dots, 1), \qquad y|\pi \sim \mathcal{N}(y|\pi, 0.5 I_K) \qquad \forall\ (x, y) \in \mathcal{D}, . \tag{16}$$

EDL links the input observations to the distribution of class probabilities by performing amortized variational inference with the approximate distribution $q_\lambda(\pi; x) = \mathcal{D}ir(\pi|\alpha_\lambda(x))$, which gives the eventual training objective

$$\mathcal{F}_{EDL}(\lambda) = \mathbb{E}_{q_\lambda(\pi|x)} \left[\sum_{(x,y)\in\mathcal{D}} ||y - \pi||_2^2 + \beta \log \frac{q(\pi|x)}{p(\pi)} \right], \tag{17}$$

up to an additive constant due to the normalization factor of the Gaussian distribution[1] and assuming $\beta = 1$ (see Appendix A.3 for the derivation details). Relying on a $\beta \neq 1$ is similar to the modification of the ELBO studied e.g. in the β-VAE model (Higgins et al., 2017).

Neural Processes (NP) (Garnelo et al., 2018a;b) are Parametric Bayesian Models used to generate stochastic processes from neural networks by integrating out the global parameters θ as in Property (i) of Definition 1:

$$p(Y|X) = \int \prod_{(x_i,y_i)\in\mathcal{D}} p(y_i|\theta, x_i) p(\theta) d\theta. \tag{18}$$

NPs differ from Parametric Bayesian Models by amortizing an arbitrary sized context set $\mathcal{D}_C$ on the global parameters $q(\theta|\mathcal{D}_C)$ via an aggregator network during inference. NPs can be viewed as Turing Processes with an identity map r. The training objective of an NP reads

$$\mathcal{F}_{NP}(\lambda) = \mathbb{E}_{q_\lambda(\theta|\mathcal{D})} \left[-\sum_{(x,y)\in\mathcal{D}\setminus\mathcal{D}_C} \log p(y|\theta, x) + \log \frac{q_\lambda(\theta|\mathcal{D})}{q_\lambda(\theta|\mathcal{D}_C)} \right]. \tag{19}$$

An NP shares with the proposed ETP the reliance on a context set during training time, but has the limitation of requiring it also at test time, while the ETP goes beyond this restriction thanks to its memory structure. NPs have since been extended to allow for attention mechanisms (Kim et al., 2019), translation equivariance (Gordon et al., 2020; Foong et al., 2020), or the application area of temporal data (Singh et al., 2019; Yoon et al., 2020). All of these NP variants assume prediction-time access to context, making them suitable for interpolation tasks such as image inpainting but not for generic prediction problems.

Evidential Neural Processes (ENPs) can be formulated as the Evidential Bayesian Model version of a neural process without an external memory. Adopting the categorical likelihood with class probabilities governed by a Dirichlet prior with hyperparameters obtained from a global hyperprior that aggregates a context set $\mathcal{D}_C$, we arrive at the data generating process below that brings together best practices of EDL and NPs to make a strong enough baseline to challenge our ETP

$$Z|\mathcal{D}_C \sim \mathcal{N}\big(Z|(\mu, \sigma^2) = r(\{h_\theta(x_j, y_j) \mid (x_j, y_j) \in \mathcal{D}_C\})\big), \tag{20}$$

$$\pi|Z \sim \mathcal{D}ir\big(\pi|\alpha_\theta(Z)\big), \tag{21}$$

$$y|\pi \sim \mathcal{C}at(y|\pi) \qquad \forall\ (x, y) \in \mathcal{D}, \tag{22}$$

with an aggregation rule $r(\cdot)$, an encoder net $h_\theta(\cdot, \cdot)$, and a neural net $\alpha_\theta(\cdot)$ mapping the global variable Z to the probability simplex. where we also apply this time to a categorical distribution for the y. To prevent the ENP from underfitting to its context set, we further increase the flexibility of the model by following the attentive NP approach (Kim et al., 2019) and replace the usually fixed aggregation rule r with an attention mechanism, allowing the model to switch focus on different parts of the context sets depending on the current target. At the prediction time, we use $Z \sim \mathcal{N}(Z|\mathbf{1}, \kappa^2 I)$ for some small κ to induce an uninformative prior on π in the absence of a context set.

[1]EDL uses a slight modification of the KL term in (17) that only regularizes the output of the non-target dimensions by using $\tilde{\alpha} = y + (1 - y) \circ \alpha_\theta(x)$, where $\circ$ refers to elementwise multiplication.

5 CASE STUDY: CLASSIFICATION WITH EVIDENTIAL TURING PROCESSES

Evidential Turing Processes introduce a generic set of design principles for a random variable system. We demonstrate a realization of an Evidential Turing Process to a classification problem below

$$Z|M \sim \prod_{r=1}^{R} \mathcal{N}(z_r|m_r, \kappa^2 I), \tag{23}$$

$$\pi|Z; x \sim \mathcal{D}ir\big(\pi|\exp(a(v_w(x); Z))\big), \tag{24}$$

$$y|\pi \sim \mathcal{C}at(y|\pi) \quad \forall\, (x, y) \in \mathcal{D}. \tag{25}$$

The Normal distribution over the Z is parameterized by the memory $M = (m_1, \ldots, m_R)$ consisting of R K-dimensional cells $m_r \in \mathbb{R}^K$.[2] The input data x is mapped to the same space via an encoding neural net $v_w(\cdot)$ parameterizing a Dirichlet distribution over the class probabilities π via an attention mechanism $a(v_w, z) = \sum_{z' \in Z} \phi(v_w(x), z')z$, where

$$\phi(v_w(x), z) \triangleq \frac{\exp(k_\psi(z)^\top v_w(x)/\sqrt{K})}{\sum_{z' \in Z} \exp\left(k_\psi(z')^\top v_w(x)/\sqrt{K}\right)}. \tag{26}$$

The function $k_\psi(\cdot)$ is the key generator network operating on the embedding space. We restrict ourselves to the scaled dot product as the similarity metric within this work to keep the presentation simple. Alternatives such as Laplace and cosine similarities, or making use of multi-head attention (Vaswani et al., 2017) would be directly applicable in practice. We update the memory cells as a weighted average between remembering and updating

$$m \leftarrow \mathbb{E}_{Z \sim p(Z|M)} \left[\tanh\Big(\gamma m + (1-\gamma) \sum_{(x,y) \in \mathcal{D}_C} \phi(v_w(x), z)\big(\text{onehot}(y) + \text{soft}(v_w(x))\big)\Big) \right], \tag{27}$$

where $\gamma \in (0, 1)$ is a fixed scaling factor controlling the relative importance. The second term adds new information to the cell as a weighted sum over all pairs (x, y), taking both the true label (via onehot(y)) as well as the uncertainty in the prediction (via soft$(v_\theta(x))$) into account. The final $\tanh(\cdot)$ transformation ensures that the memory content remains in a fixed range across updates, as practised in LSTMs (Hochreiter & Schmidhuber, 1997).

6 EXPERIMENTS

If our *Main Hypothesis* in Page 4 is correct, then ETP should outperform the alternative Bayesian modeling approaches introduced in Section 4 in multiple data sets with respect to multiple uncertainty quantification tasks as a single stand-alone model. We devise an experiment setup to test this hypothesis. We first illustrate the qualitative properties of ETP in a toy example and then benchmark it in three image classification tasks against BNN to represent the best practice of Parametric Bayesian Models, EDL as the state of the art of Evidential Bayesian Models, and ENP as a straightforward adaptation of Neural Processes into Evidential Bayesian Models. We refer to the ETP and ENP with deterministic neural net parameters as ETP-D and ENP-D, respectively. We perform all the image classification experiments with LeNet5-sized architectures, leaving a rigorous neural architecture search to future work. See the Appendix B for full details on the experimental setup in each case and for the hyperparameters used throughout. We also provide a reference implementation of the proposed model and the experimental pipeline.[3]

Illustrative Example. We illustrate the qualitative behaviour of ETP via two experiments with a reduced version of the model that uses identity mappings for $k(\cdot)$ and $h(\cdot)$ and rescales the memory to improve transparency. The first illustrates the ETP behaviour on a simple synthetic 1d two-class classification task, while a second replicates this on a real world 2d classification task (see appendix section B.2 for the second experiment). The data consist of observations (20 per class) from two highly overlapping Gaussian distributions. The neural net, a simple one hidden layer architecture,

[2] In the experiments we fix $\kappa^2 = 0.1$. Preliminary results showed stability as well for larger/smaller values.

[3] *Hidden to keep anonymity*

Table 1: Image classification and calibration results. The results give mean $\pm$ one standard deviation over ten replications with the best marked in bold.

Dataset	Method	Err (%) (↓)	NLL (↓)	ECE (%) (↓)	AU-ROC (%) (↑)	TPE (↓)
FMNIST	BNN	**7.9**$\pm$**0.1**	0.65$\pm$0.01	6.7$\pm$0.1	75.9$\pm$2.3	**6.1**$\pm$**0.1**
	EDL	8.6$\pm$0.1	0.37$\pm$0.01	3.7$\pm$0.2	77.5$\pm$2.0	6.6$\pm$0.1
	ENP-D	8.8$\pm$0.1	0.30$\pm$0.00	4.1$\pm$0.2	74.5$\pm$2.5	10.1$\pm$0.1
	ENP	8.2$\pm$0.2	0.40$\pm$0.01	5.8$\pm$0.2	74.1$\pm$1.6	11.5$\pm$0.1
	ETP-D	8.0$\pm$0.1	**0.29**$\pm$**0.00**	2.7$\pm$0.1	**91.4**$\pm$**1.4**	7.8$\pm$0.1
	ETP	**7.9**$\pm$**0.2**	**0.29**$\pm$**0.01**	**2.6**$\pm$**0.2**	90.0$\pm$0.9	8.8$\pm$0.1
CIFAR10	BNN	15.3$\pm$0.3	0.50$\pm$0.01	5.5$\pm$0.3	84.1$\pm$1.3	**8.2**$\pm$**0.1**
	EDL	18.5$\pm$0.2	0.72$\pm$0.01	9.0$\pm$0.2	79.2$\pm$0.7	8.6$\pm$0.1
	ENP-D	17.4$\pm$0.1	0.52$\pm$0.01	3.3$\pm$0.1	81.3$\pm$1.1	23.7$\pm$0.2
	ENP	**15.1**$\pm$**0.2**	0.47$\pm$0.00	4.1$\pm$0.2	**84.3**$\pm$**0.9**	24.9$\pm$0.1
	ETP-D	16.6$\pm$0.3	0.49$\pm$0.01	**2.4**$\pm$**0.1**	81.4$\pm$1.4	9.9$\pm$0.1
	ETP	15.3$\pm$0.2	**0.46**$\pm$**0.00**	2.7$\pm$0.1	82.1$\pm$0.6	10.8$\pm$0.1
SVHN	BNN	7.9$\pm$0.1	0.71$\pm$0.01	6.5$\pm$0.1	86.2$\pm$0.5	**11.2**$\pm$**0.1**
	EDL	7.3$\pm$0.1	0.34$\pm$0.00	4.0$\pm$0.1	**90.9**$\pm$**0.3**	11.8$\pm$0.1
	ENP-D	7.7$\pm$0.1	0.36$\pm$0.01	4.5$\pm$0.1	86.7$\pm$0.4	34.4$\pm$0.2
	ENP	7.5$\pm$0.1	0.40$\pm$0.00	4.8$\pm$0.1	84.6$\pm$0.4	35.5$\pm$0.3
	ETP-D	7.9$\pm$0.1	0.29$\pm$0.00	**2.4**$\pm$**0.2**	85.9$\pm$0.8	13.7$\pm$0.3
	ETP	**6.9**$\pm$**0.1**	0.26$\pm$0.00	**2.6**$\pm$**0.1**	90.0$\pm$0.4	15.2$\pm$0.2

can learn the task reasonably well. Figure 3 shows the raw data and underlying distributions as well as the learned memory evidence. The model learns to place more evidence on the correct class further away from the decision boundary while also becoming more varied with increasing distance. Within the high-density region, the asymmetry in how the specific observations are spread out leads to an asymmetry in the memory evidence. Below zero, the blue points are well separated, leading to a quick emphasis on that class as we move towards the left. Above zero, however, some blue observations in the orange region prevent the model from rapidly developing over-confidence.

Image classification. We evaluate all models on three pairs of in-domain/out-of-domain image data sets: FMNIST/MNIST, CIFAR10/SVHN and SVHN/CIFAR10. In Table 1, we report the prediction error (Err), Expected Calibration Error (ECE) (Naeini et al., 2015), and Negative Log-Likelihout-of-domain (NLL), Area Under ROC Curve (AUROC) for out-of-domain detection, and the computational cost per-epoch (TPE) as wall clock time. We perform the out-of-domain detection task as in Malinin & Gales (2018) and classify the test split of the target domain and a data set from another domain based on the the entropy $\mathcal{H}[\mathbb{E}_{p(\pi|x;\theta)}\,[p(y|\pi)]]$ of the predictive distribution. ETP and its variant ETP-D are the only method that consistently rank among top performers in all three data sets with respect to all in-domain calibration scores without losing the ability of out-of-domain detection. We view this outcome as a strong evidence in favor of our *Main Hypothesis:* Devising a model that can express all the three terms in Eq. 10 is essential for complete uncertainty quantification. The computational cost of ETP remains comparable to BNN and EDL.

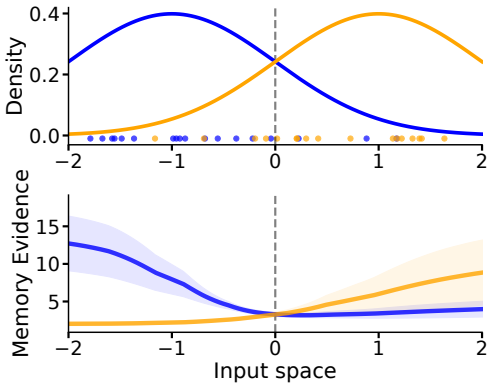


Figure 3: **1D Classification Task.** The upper plot shows the underlying distributions of each of the two classes, as well as the observed data. The lower shows the regularizing evidence the generative model places on each of the two classes depending on location in space, that is, mean $\pm$ one standard deviation over ten samples from $p(Z|M)$.

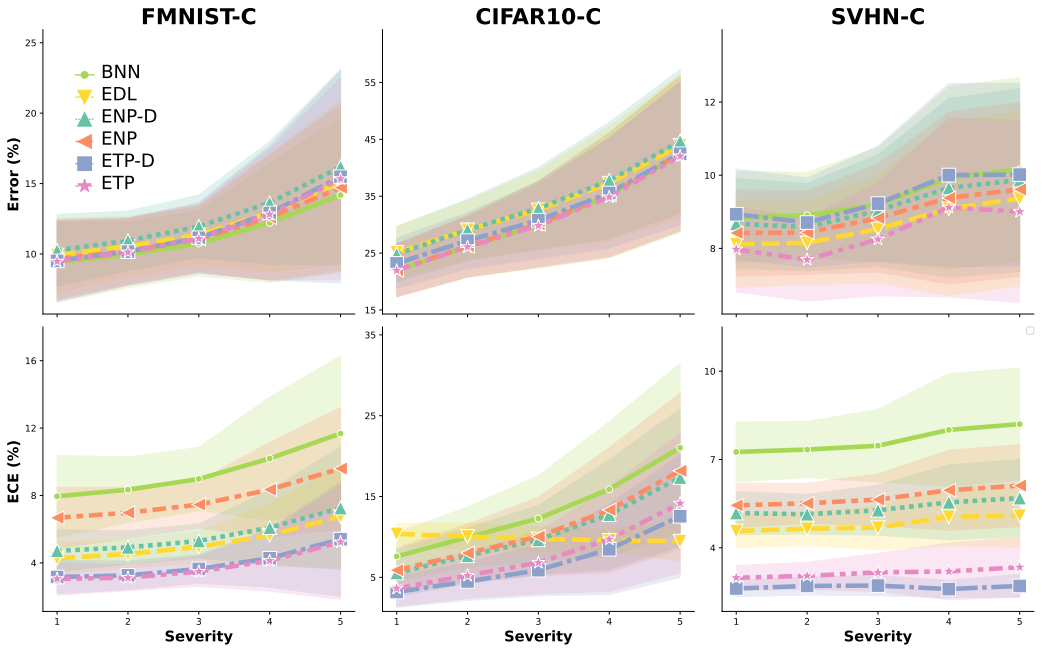


Figure 4: Benchmarking results for robustness on three data sets: FMNIST-C, CIFAR10-C, and SVHN-C. *(top row)* prediction error. *(bottom row)* Expected Calibration Error (ECE). The results are the average and one standard deviation error bars over ten replications (see Appendix B.3 Table 3 for a tabular version of these results).

Robustness against corruption. We evaluate the robustness of the models with respect to 19 types of in-distribution training data corruption as in Hendrycks & Dietterich (2019) at five different severity levels. Figure 4 depicts the performance of models averaged across all corruptions. ETP gives the best performance nearly in all data sets and distortion severities (see Appendix B.3 Table 3 for a tabular version of these results).

7 Broad Impact and Limitations

We view the empirical outcome of this work as a proof-of-concept demonstrating that epistemic awareness of a Bayesian model is indeed a capability learnable only from in-domain data, as appears in biological intelligence via closed-loop interactions of neuronal systems with stimuli. The implications of our work could follow up in interdisciplinary venues, with a particular focus on the relation of associative memory and attention to the neuroscientific and learning-theoretic roots, as well as algorithmic explanations of epistemic awareness (Lorenz, 1978).

Bayesian inference by perturbing neural net weights provides a strong alternative to ETP. While we observed ETP outperform BNN variants in our experiments, whether BNNs could bridge the gap after further improvements in the approximate inference remains to be researched. The attention mechanism of ETP could also be improved using more advanced attention methods. The explainability of the memory content of ETP is an open issue, which deserves further investigation. The generalizeability of our results to very deep architectures, such as ResNet 152 or DenseNet 161 could be a topic of a separate large-scale empirical study.

ETP is meant mainly as a method to facilitate the safe use of Bayesian models. However, it does not improve the explainability and fairness of the decisions made by the underlying design choices, such as the architecture of the used neural nets in the pipeline. Potential negative societal impacts of deep neural net classifiers stemming from these two factors need to be circumvented separately before the real-world deployment of our work.

REFERENCES

Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions*. Dover, 1965.

Andrew Foong, Wessel Bruinsma, Jonathan Gordon, Yann Dubois, James Requeima, and Richard Turner. Meta-Learning Stationary Stochastic Process Prediction with Convolutional Neural Processes. In *Advances in Neural Information Processing Systems*, 2020.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional Neural Processes. In *International Conference on Machine Learning*, 2018a.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural Processes. *Theoretical Foundations and Applications of Deep Generative Models workshop*, 2018b.

Jonathan Gordon, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner. Convolutional Conditional Neural Processes. In *International Conference on Learning Representations*, 2020.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing Machines. *arXiv preprint arXiv:1410.5401*, 2014.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning*, 2017.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *Internationl Conference on Learning Representations*, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive Neural Processes. In *International Conference on Learning Representations*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P. Kingma, Tim Salimans, and Max Welling. Variational Dropout and the Local Reparameterization Trick. *Advances in Neural Information Processing Systems*, 2015.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate Uncertainties for Deep Learning using Calibrated Regression. In *International Conference on Machine Learning*, 2018.

Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Konrad Lorenz. *Die Rückseite des Spiegels: Versuch einer Naturgeschichte menschlichen Erkennens*. Piper, 1978.

David JC MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 1992.

David JC MacKay. Probable Networks and Plausible Predictions – A Review of Practical Bayesian Methods for Supervised Neural Networks. *Network: Computation in Neural Systems*, 1995.

Andrey Malinin and Mark Gales. Predictive Uncertainty Estimation via Prior Networks. In *Advances in Neural Information Processing Systems*, 2018.

Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational Dropout Sparsifies Deep Neural Networks. In *International Conference on Machine Learning*, 2017.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well Calibrated Probabilities using Bayesian Binning. In *AAAI Conference on Artificial Intelligence*, 2015.

Radford Neal. *Bayesian Learning for Neural Networks*. PhD thesis, 1995.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. 2011.

Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer-Verlag, 1992.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. 2019.

Gesina Schwalbe and Martin Schels. A Survey on Methods for the Safety Assurance of Machine Learning Based Systems. In *European Congress on Embedded Real Time Software and Systems*, 2020.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential Deep Learning to Quantify Classification Uncertainty. In *Advances in Neural Information Processing Systems*, 2018.

Gautam Singh, Jaesik Yoon, Youngsung Son, and Sungjin Ahn. Sequential Neural Processes. In *Advances in Neural Information Processing Systems*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, 2017.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. 2017.

Jaesik Yoon, Gautam Singh, and Sungjin Ahn. Robustifying Sequential Neural Processes. In *International Conference on Machine Learning*, 2020.

APPENDIX

A FURTHER THEORETICAL RESULTS AND DERIVATIONS

A.1 KULLBACK LEIBLER BETWEEN TWO DIRICHLET DISTRIBUTIONS

As our objective and the EDL baseline rely on the Kullback-Leibler divergence between two Dirichlet distributions, we repeat here for completeness the general form. Assuming two distributions over a K-dimensional probability π, $\mathcal{D}ir(\pi|\alpha)$ and $\mathcal{D}ir(\pi|\beta)$, parameterized by $\alpha, \beta \in \mathbb{R}_+^K$, we have

$$\text{KL}\left(\mathcal{D}ir(\pi|\alpha) \parallel \mathcal{D}ir(\pi|\beta)\right) = \log\left(\frac{\Gamma(\sum_k \alpha_k)\prod_k \Gamma(\beta_k)}{\Gamma(\sum_k \beta_k)\prod_k \Gamma(\alpha_k)}\right) + \sum_k (\alpha_k - \beta_k)\big(\psi(\alpha_k) - \psi(\textstyle\sum_k \alpha_k)\big),$$

where $\Gamma(\cdot)$ and $\psi(a) := \frac{d}{da}\log\Gamma(a)$ are the *gamma* and *digamma* functions (Abramowitz & Stegun, 1965).

A.2 THE EVIDENTIAL DEEP LEARNING OBJECTIVE

The analytical solutions for the Evidential Deep Learning (Sensoy et al., 2018) loss as defined in the main paper in (17) can be computed as follows. For a single K-dimensional observation y, dropping the n from the notation and instead using the index for the dimensionality throughout the following equations, we have that the expectation in the expectation terms are given as

$$\begin{aligned}
\mathbb{E}_{p(\pi|x)}\left[||y-\pi||^2\right] &= \mathbb{E}_{p(\pi|x)}\left[(y-\pi)^\top(y-\pi)\right] \\
&= \sum_{k=1}^{K} \mathbb{E}_{p(\pi|x)}\left[(y_k-\pi_k)^2\right] \\
&= \sum_{k=1}^{K} \mathbb{E}_{p(\pi|x)}\left[(y_k - \mathbb{E}_{p(\pi|x)}\left[\pi_k\right] + \mathbb{E}_{p(\pi|x)}\left[\pi_k\right] - \pi_k)^2\right] \\
&= \sum_{k=1}^{K} (y_k - \mathbb{E}_{p(\pi|x)}\left[\pi_k\right])^2 + \text{var}_{p(\pi|x)}\left[\pi_k\right],
\end{aligned}$$

with the tractable expectation and variance of a Dirichlet distributed π. The Kullback-Leibler term is between two Dirichlet distributions and given (see the general form in A.1) as

$$\begin{aligned}
&\text{KL}\left(\mathcal{D}ir(\pi|\alpha_\theta(x)) \parallel \mathcal{D}ir(\pi|1,\dots,1)\right) \\
&= \log\left(\frac{\Gamma\big(\sum_k \alpha_\theta(x)_k\big)}{\Gamma(K)\prod_k \Gamma(\alpha_\theta(x)_k)}\right) + \sum_{k=1}^{K}(\alpha_\theta(x)_k - 1)\Big(\psi(\alpha_\theta(x)_k) - \psi\big(\textstyle\sum_k \alpha_\theta(x)_k\big)\Big).
\end{aligned}$$

A.3 EVIDENTIAL DEEP LEARNING AS A LATENT VARIABLE MODEL

As discussed in the main paper for a set of N observations $\mathcal{D} = \{(x_1, y_1), \dots, (x_N, y_N)\}$, the objective EDL minimizes is given via (17) as

$$\mathcal{L}_{\text{EDL}} = \sum_{n=1}^{N} \mathbb{E}_{p(\pi_n|x_n)}\left[||y_n - \pi_n||_2^2\right] + \lambda \text{KL}\left(p(\pi_n|x_n) \parallel \mathcal{D}ir(\pi_n|1,\dots,1)\right),$$

where $p(\pi_n|x_n) = \mathcal{D}ir(\pi|\alpha_\theta(x_n))$. We can instead assume the following generative model

$$\begin{aligned}
\pi_n &\sim \mathcal{D}ir(\pi_n|1,\dots,1) \qquad \forall n \\
y_n &\sim \mathcal{N}(y_n|\pi_n, 0.5 I_K) \qquad \forall n,
\end{aligned}$$

i.e. latent variables π_n and K-dimensional observations y_n following a multivariate normal prior. Approximating the intractable posterior $p(\boldsymbol{\pi}|\mathbf{y})$, where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_n)$, $\mathbf{y} = (y_1, ..., y_n)$, with an

amortized variational posterior $q(\pi_n; x_n) = \mathcal{D}ir(\pi_n|\alpha_\theta(x_n))$, where $\alpha_\theta(\cdot)$ is the same architecture as in the EDL model, we have as the *evidence lower bound (ELBO)* to be maximized

$$
\begin{aligned}
& \sum_{n=1}^{N} \mathbb{E}_{q(\pi_n;x_n)} \left[\log p(y_n|\pi_n)\right] - \mathrm{KL}\left(q(\pi_n; x_n) \parallel p(\pi_n)\right) \\
= & \sum_{n=1}^{N} \mathbb{E}_{q(\pi_n;x_n)} \left[-\frac{1}{2}\log(2\pi) + \frac{K}{2}\log(2) - (y_n - \pi_n)^\top (y_n - \pi_n)\right] - \mathrm{KL}\left(q(\pi_n; x_n) \parallel p(\pi_n)\right) \\
= & \text{ const} - \sum_{n=1}^{N} \mathbb{E}_{q(\pi_n;x_n)} \left[(y_n - \pi_n)^\top (y_n - \pi_n)\right] - \mathrm{KL}\left(q(\pi_n; x_n) \parallel p(\pi_n)\right) \\
= & \text{ const} - \mathcal{L}_{\text{EDL}},
\end{aligned}
$$

that is the negative ELBO is equal to the EDL loss up to an additive constant.

A.4 Posterior Predictive

The posterior predictive for y^* given a new observation x^* can be computed as

$$
p(y^* = k|x^*, \mathcal{D}) \approx \mathbb{E}_{p(Z|M)} \left[\mathbb{E}_{q_\lambda(\pi|x^*,Z)} \left[p(y^* = k|\pi)\right]\right] = \mathbb{E}_{p(Z|M)} \left[h_\theta^k / \textstyle\sum_{k'} h_\theta^{k'}\right],
$$

where $h_\theta^k = h_\theta\left(v_\theta(x^*), a(v_\theta(x^*); Z)\right)_k$. That is, we can compute it analytically up to a sampling-based evaluation of the expectation over $p(Z|M)$.

B Experimental Details

B.1 Synthetic 1D two-class classification

The data set used in this experiment is a synthetic data set consisting of two classes with 20 observations each sampled from standard normal distributions centered at ± 1 respectively. The encoding function $v_\phi(\cdot)$ consists of a multi-layer perceptron with a single hidden layer consisting of 32 neurons and a ReLU activation function. It is optimized for 400 epochs with the Adam optimizer (Kingma & Ba, 2014) using the PyTorch (Paszke et al., 2019) default parameters and a learning rate of $1e-3$. In order to keep the model as simple as possible, the key generator function $k_\lambda(\cdot)$ is constrained to being the identity function, while $h_\xi\big(v(x), a(v(x); Z)\big) = v(x) + \tanh(a(v(x); Z)$. The memory update is simplified to dropping the $\tanh(\cdot)$ rescaling.

B.2 Iris 2D three-class classification

The data set used in this experiment is the classical Iris data set[4] consisting of 150 samples of three iris species (50 per class), with four features measured per sample. We first map the data to the first two principal components for visualization and also use this modified version as the training data. This gives an interpretable toy learning setup where one class is clearly separated from the other two overlapping classes. The encoding function $v_\phi(\cdot)$ consists of an MLP with two hidden layers of 32 neurons each and ReLU activation functions. It is optimized for 400 epochs with the Adam optimizer using the PyTorch default parameters and a learning rate of $1e-3$. In order to keep the model as simple as possible, the key generator function $k_\lambda(\cdot)$ is constrained to being the identity function, while $h_\xi\big(v(x), a(v(x); Z)\big) = v(x) + \tanh(a(v(x); Z)$. The memory update is simplified to dropping the $\tanh(\cdot)$ rescaling.

Figure 5 shows the varying importance the model assigns to different parts of the input space depending on the class under consideration. For the separate blue class, the model is a lot more confident in relative terms and absolute values. See Figure 6 for an enhanced set of plots showing the range for each of them. See Figure 7 for an enhanced plot of the horizontal cut through the input

[4] Originally due to Fisher (1936) and Anderson (1935). We rely on the version provided by the `scikit learn` library, see https://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets.

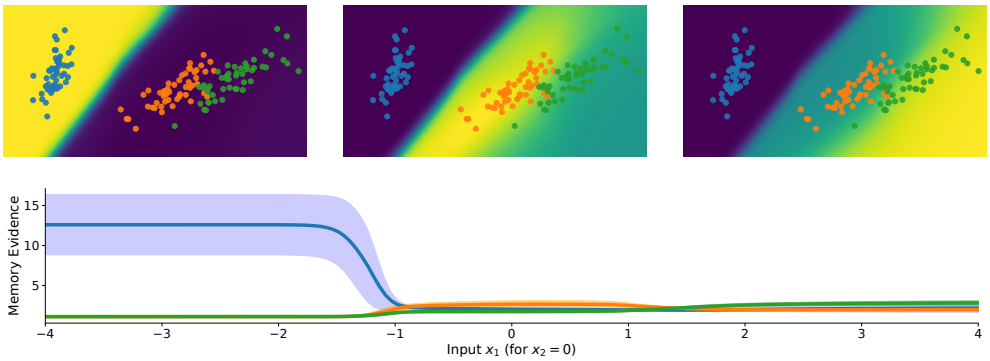


Figure 5: **2D Classification Task.** The three upper plots visualize the regularizing evidence that the model places on the location in space as the average over ten samples from $p(Z|M)$. The color scales are different across the three plots and vary in overall intensity. The lower plot visualizes a horizontal cut through the middle of these plots, putting them on a common scale. The solid lines in each color indicate the mean memory evidence, and the corresponding shaded areas indicate one standard deviation. Around the separable blue class, the memory strongly emphasizes the correct class with large confidence and suppresses the other two classes depicted in orange and green. While always preferring the correct class, the memory regularizes against overconfidence around the overlapping orange and green classes.

space at $x_2 = 0$, while varying the first input dimension. It shows the certainty (although with a large variance) assigned to the separate blue class and the significantly more conservative regularization in the overlapping region starting around $x_1 = -1$.

B.3 Image Classification Experiments

All experiments are implemented in PyTorch (Paszke et al., 2019) version 1.7.1 and trained on a TITAN RTX. They have been replicated ten times over random initial seeds. The experimental details are as follows:

Fashion MNIST (FMNIST) (Xiao et al., 2017). Uses a LeNet5-sized architecture (see Table 2). The out-of-distribution data is the MNIST (LeCun et al., 2010) data set. All data sets are z-score normalized given the in-domain mean and standard deviations. The training objective for each model is optimized via the Adam optimizer (Kingma & Ba, 2014) with a learning rate of $1e-3$. Each model is trained for 50 epochs.

CIFAR10 (C10) (Krizhevsky et al., 2009). Uses a LeNet5-sized architecture (see Table 2). The out-of-distribution data is the SVHN (Netzer et al., 2011). All data sets are z-score normalized given the in-domain mean and standard deviations. We further rely on data augmentation for the in domain data using random crops with 32 pixels and four pixel padding as well as horizontal flips. The training objective for each model is optimized via the Adam optimizer (Kingma & Ba, 2014) with a learning rate of $1e-4$. Each model is trained for 100 epochs.

SVHN (Netzer et al., 2011). Uses a LeNet5-sized architecture (see Table 2). The out-of-distribution data is the CIFAR10 (Krizhevsky et al., 2009) data set. All data sets are z-score normalized given the in-domain mean and standard deviations. The training objective for each model is optimized via the Adam optimizer (Kingma & Ba, 2014) with a learning rate of $1e-4$. Each model is trained for 100 epochs.

Shared Details and Observations. In all experiments, we train the BNN models with a cross-entropy loss, relying on the softmax for prediction. The EDL model is trained with the original loss (Sensoy et al., 2018). The prediction is made with the mean of the Dirichlet distribution like in the original work. In all models, entropy is used for detecting out-of-distribution instances.

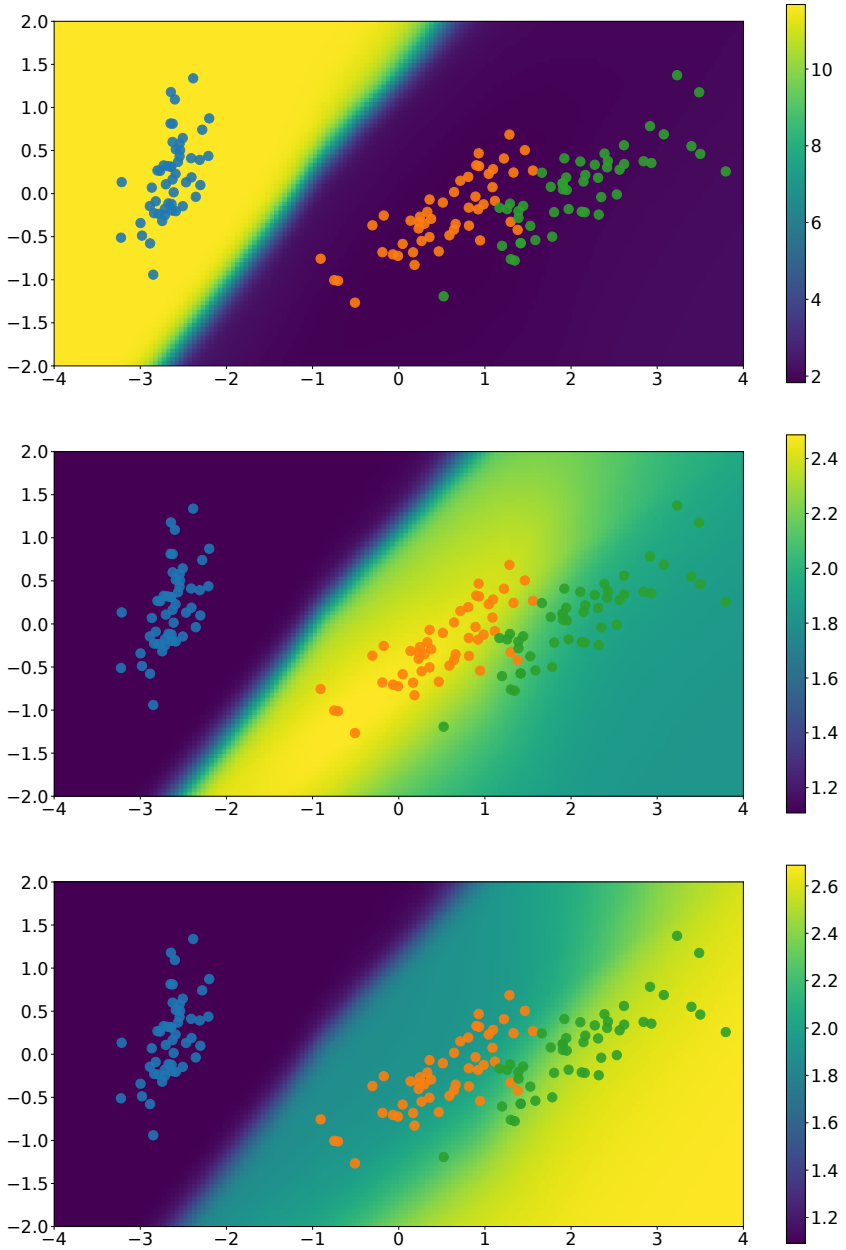


Figure 6: **2D Classification Enhanced.** The enlarged version of the visualization in Figure 3. While the range of values between the memory evidence belonging to the orange and the green class is roughly similar, for the well separated blue class the model assigns a much larger absolute value.

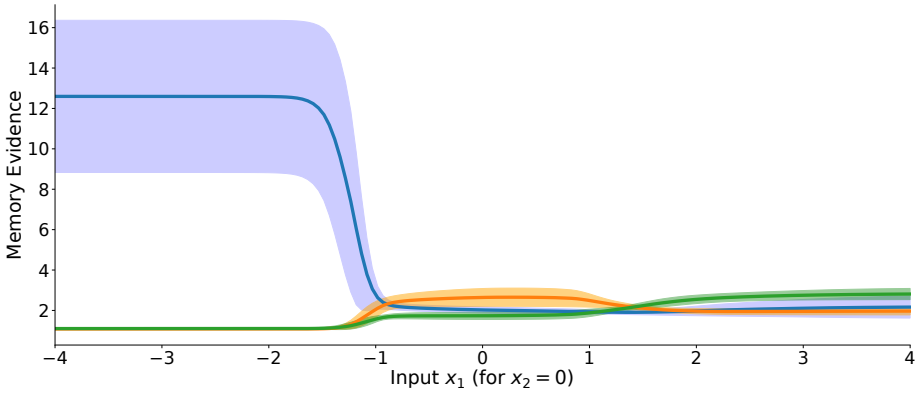


Figure 7: **2D Classification - Horizontal Cut.** An overlay of Figure 6 at $x_2 = 0$. It shows the mean memoevidence $\pm$ one standard deviation (over ten samples) for each of the three classes.

Table 2: The neural network architectures used for the FMNIST, CIFAR10 and SVHN data set with ReLU activations between the layers.

FMNIST	CIFAR10/SVHN
Convolution (5×5) with 20 channels	Convolution (5×5) with 192 channels
MaxPooling (2×2) with stride 2	
Convolution (5×5) with 50 channels	Convolution (5×5) with 192 channels
MaxPooling (2×2) with stride 2	
Linear with 500 neurons	Linear with 1000 neurons
Linear with 10 neurons	

Robustness against perturbations. The CIFAR10-C dataset is taken from the original work by Hendrycks & Dietterich (2019). As FMNIST-C and SVHN-C are not available, we created them are created following the same procedure as in the original work. For FMNIST the procedure, designed for three-channel RGB images, needs to be adapted to work with one-channel gray-scale images. To that end, we convert the gray-scale images to pseudo-RGB images by replicating the original gray channel three times. After applying the distortions, we transfer it again back into a one-channel gray-scale format. Table 3 gives the raw results used to create Figure 4 in the main paper.

Table 3: Quantitative results of models on the FMNIST-C, CIFAR10-C and SVHN-C test dataset using the LeNet5. The results shown mean $\pm$ one standard deviation.

		FMNIST-C		CIFAR10-C		SVHN-C	
Severity	Method	Err (%) (↓)	ECE (%) (↓)	Err (%)(↓)	ECE (%) (↓)	Err (%)(↓)	ECE (%) (↓)
1	BNN	$9.4_{\pm 2.8}$	$8.0_{\pm 2.4}$	$21.9_{\pm 4.5}$	$7.6_{\pm 2.9}$	$8.8_{\pm 1.2}$	$7.3_{\pm 1.0}$
	EDL	$9.9_{\pm 2.6}$	$4.3_{\pm 1.0}$	$25.1_{\pm 4.6}$	$10.4_{\pm 1.3}$	$8.1_{\pm 1.2}$	$4.6_{\pm 0.6}$
	ENP-D	$10.3_{\pm 2.6}$	$4.7_{\pm 1.2}$	$24.8_{\pm 4.8}$	$5.5_{\pm 2.9}$	$8.7_{\pm 1.2}$	$5.2_{\pm 0.7}$
	ENP	$9.6_{\pm 2.8}$	$6.7_{\pm 1.8}$	$22.0_{\pm 4.7}$	$5.9_{\pm 2.8}$	$8.4_{\pm 1.2}$	$5.4_{\pm 0.7}$
	ETP-D	$9.5_{\pm 2.8}$	$3.2_{\pm 1.0}$	$23.2_{\pm 4.4}$	$3.2_{\pm 1.9}$	$8.9_{\pm 1.2}$	$2.6_{\pm 0.3}$
	ETP	$9.5_{\pm 2.8}$	$3.0_{\pm 1.0}$	$21.9_{\pm 4.6}$	$3.6_{\pm 2.2}$	$8.0_{\pm 1.2}$	$3.0_{\pm 0.4}$
2	BNN	$9.9_{\pm 2.3}$	$8.3_{\pm 1.9}$	$26.1_{\pm 5.3}$	$10.0_{\pm 3.7}$	$8.9_{\pm 1.2}$	$7.3_{\pm 0.9}$
	EDL	$10.5_{\pm 2.1}$	$4.5_{\pm 0.8}$	$29.1_{\pm 5.1}$	$10.1_{\pm 1.6}$	$8.2_{\pm 1.1}$	$4.6_{\pm 0.6}$
	ENP-D	$10.9_{\pm 2.1}$	$4.9_{\pm 1.0}$	$29.1_{\pm 5.4}$	$7.7_{\pm 3.4}$	$8.6_{\pm 1.2}$	$5.1_{\pm 0.7}$
	ENP	$10.2_{\pm 2.3}$	$7.0_{\pm 1.5}$	$26.2_{\pm 5.4}$	$8.0_{\pm 3.6}$	$8.4_{\pm 1.1}$	$5.5_{\pm 0.7}$
	ETP-D	$10.2_{\pm 2.4}$	$3.3_{\pm 0.8}$	$27.2_{\pm 4.9}$	$4.5_{\pm 2.3}$	$8.7_{\pm 1.2}$	$2.7_{\pm 0.3}$
	ETP	$10.1_{\pm 2.4}$	$3.1_{\pm 0.8}$	$26.1_{\pm 5.3}$	$5.2_{\pm 2.8}$	$7.7_{\pm 1.1}$	$3.0_{\pm 0.5}$
3	BNN	$10.7_{\pm 2.3}$	$9.0_{\pm 1.9}$	$30.0_{\pm 7.6}$	$12.3_{\pm 5.3}$	$9.2_{\pm 1.6}$	$7.5_{\pm 1.2}$
	EDL	$11.4_{\pm 2.1}$	$4.9_{\pm 0.9}$	$32.6_{\pm 7.0}$	$9.8_{\pm 1.5}$	$8.5_{\pm 1.5}$	$4.7_{\pm 0.7}$
	ENP-D	$12.0_{\pm 2.2}$	$5.3_{\pm 1.0}$	$32.9_{\pm 7.2}$	$9.6_{\pm 4.3}$	$9.0_{\pm 1.5}$	$5.3_{\pm 0.8}$
	ENP	$11.1_{\pm 2.3}$	$7.5_{\pm 1.4}$	$30.1_{\pm 7.6}$	$10.0_{\pm 4.8}$	$8.8_{\pm 1.5}$	$5.6_{\pm 0.9}$
	ETP-D	$11.1_{\pm 2.5}$	$3.6_{\pm 0.8}$	$30.8_{\pm 6.7}$	$5.9_{\pm 3.2}$	$9.2_{\pm 1.6}$	$2.7_{\pm 0.3}$
	ETP	$11.1_{\pm 2.4}$	$3.5_{\pm 0.9}$	$29.8_{\pm 7.2}$	$6.8_{\pm 3.8}$	$8.2_{\pm 1.5}$	$3.2_{\pm 0.6}$
4	BNN	$12.2_{\pm 4.2}$	$10.2_{\pm 3.6}$	$35.1_{\pm 10.9}$	$15.9_{\pm 8.3}$	$9.9_{\pm 2.5}$	$8.0_{\pm 1.9}$
	EDL	$12.9_{\pm 4.2}$	$5.6_{\pm 1.8}$	$37.3_{\pm 10.0}$	$9.6_{\pm 1.4}$	$9.1_{\pm 2.4}$	$5.1_{\pm 1.0}$
	ENP-D	$13.6_{\pm 4.4}$	$6.1_{\pm 2.2}$	$37.8_{\pm 10.2}$	$12.8_{\pm 6.8}$	$9.7_{\pm 2.4}$	$5.5_{\pm 1.3}$
	ENP	$12.6_{\pm 4.4}$	$8.3_{\pm 2.8}$	$35.2_{\pm 10.9}$	$13.3_{\pm 7.7}$	$9.4_{\pm 2.3}$	$6.0_{\pm 1.4}$
	ETP-D	$12.9_{\pm 4.7}$	$4.3_{\pm 1.7}$	$35.6_{\pm 9.7}$	$8.5_{\pm 5.6}$	$10.0_{\pm 2.5}$	$2.6_{\pm 0.3}$
	ETP	$12.8_{\pm 4.6}$	$4.1_{\pm 1.8}$	$34.8_{\pm 10.3}$	$9.7_{\pm 6.4}$	$9.1_{\pm 2.4}$	$3.2_{\pm 1.0}$
5	BNN	$14.2_{\pm 5.4}$	$11.7_{\pm 4.6}$	$42.4_{\pm 13.7}$	$21.0_{\pm 10.4}$	$10.2_{\pm 2.5}$	$8.2_{\pm 1.9}$
	EDL	$15.1_{\pm 5.8}$	$6.7_{\pm 3.1}$	$43.9_{\pm 12.5}$	$9.5_{\pm 2.4}$	$9.4_{\pm 2.4}$	$5.1_{\pm 1.1}$
	ENP-D	$16.2_{\pm 6.9}$	$7.3_{\pm 3.6}$	$44.7_{\pm 12.6}$	$17.4_{\pm 8.4}$	$9.9_{\pm 2.5}$	$5.7_{\pm 1.3}$
	ENP	$14.7_{\pm 5.9}$	$9.6_{\pm 3.6}$	$42.5_{\pm 13.6}$	$18.2_{\pm 9.6}$	$9.6_{\pm 2.4}$	$6.1_{\pm 1.4}$
	ETP-D	$15.5_{\pm 7.5}$	$5.4_{\pm 3.4}$	$42.5_{\pm 12.4}$	$12.6_{\pm 7.6}$	$10.0_{\pm 2.5}$	$2.7_{\pm 0.4}$
	ETP	$15.3_{\pm 7.1}$	$5.2_{\pm 3.4}$	$42.1_{\pm 13.1}$	$14.1_{\pm 8.6}$	$9.0_{\pm 2.5}$	$3.3_{\pm 1.0}$