Anytime-Valid Confidence Sequences for Consistent Uncertainty Estimation in Early-Exit Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

Early-exit neural networks (EENNs) facilitate adaptive inference by producing predictions at multiple stages of the forward pass. In safety-critical applications, these predictions are only meaningful when complemented with reliable uncertainty estimates. Yet, due to their sequential structure, an EENN's uncertainty estimates should also be *consistent*: labels that are deemed improbable at one exit should not reappear within the confidence interval / set of later exits. We show that standard uncertainty quantification techniques, like Bayesian methods or conformal prediction, can lead to inconsistency across exits. We address this problem by applying anytime-valid confidence sequences (AVCSs) to the exits of EENNs. By design, AVCSs maintain consistency across exits. We examine the theoretical and practical challenges of applying AVCSs to EENNs and empirically validate our approach on both regression and classification tasks.

1 Introduction

Modern predictive models are increasingly deployed in environments where computational resources at test time are either constrained or dynamic. In a constrained setting, there's a fixed amount of available resources. For example, when ML models are deployed on low-resource devices, such as mobile phones, they would ideally make fast, yet accurate predictions to ensure optimal user experience. On the other hand, in a dynamic setting, the available resources can vary due to external conditions. Consider an autonomous vehicle: when it's moving at high speeds, the model must make rapid predictions. However, as the vehicle slows down, the model can afford more time to process or 'think'. Early-exit neural networks (EENNs) (Teerapittayanon et al., 2016; Huang et al., 2018) present a promising solution to challenges arising in both of these settings. As the name implies, these architectures have multiple exits that allow a prediction to be generated at an arbitrary stopping time. This is in contrast to traditional predictive models that yield a single prediction after processing all layers or model components.

To enable the use of EENNs in safety-critical applications, such as autonomous driving, it is necessary to estimate the predictive uncertainty at each exit (McAllister et al., 2017). Presently, standard techniques for uncertainty quantification, such as Bayesian methods (Meronen et al., 2023) and conformal prediction (Schuster et al., 2021), are used for this purpose. However, we are aware of no work that has accounted for the fact that the uncertainties computed for neighboring exits are *dependent*. For example, in a regression task, an uncertainty interval for the prediction at a given exit should be *consistent* with the intervals at the previous and subsequent exits (c.f. Figure 1). If a candidate prediction y_0 is in the interval at exit $t-1$ and drops out of the interval at exit t, y_0 should not re-enter the interval at exit $t+1$. An even worse case would be that the intervals at exit t and $t+1$ are disjoint. Such inconsistent behaviour limits the decisions that can be drawn at the initial exits of EENN, thereby undermining their anytime properties.

We address this open problem by applying *anytime-valid confidence sequences* (AVCSs) (Robbins, 1967; 1970; Lai, 1976) to the task of quantifying uncertainty across the exits of an EENN. AVCSs extend traditional, point-wise confidence intervals to streaming data scenarios (Maharaj et al., 2023). Importantly, AVCSs are guaranteed to have a non-increasing interval width (Howard et al., 2021) and are therefore consistent by definition. Our main insight is that AVCS can be applied even in settings where a single data point

is observed - like in estimating the uncertainty of a model's prediction for a given test point. To achieve this, we construct the AVCS using 'streaming' model parameters (e.g., last-layer weights) at different exits, instead of using streaming data as is traditionally done. We describe the approximations necessary to make AVCSs applicable for predictive inference and additionally provide bounds on the errors introduced by our approximations. In our experiments, we demonstrate that our AVCS-based procedure produces consistent estimates of predictive uncertainty across exits in EENNs for various regression and classification tasks.

2 Background

Data Let $\mathcal{X} \subseteq \mathbb{R}^D$ denote a D-dimensional feature space and $\mathcal{Y}$ the response (output) space. In the case of regression, we have $\mathcal{Y} \subseteq \mathbb{R}$, and for classification $\mathcal{Y} = \{1, \ldots, K\}$. We assume $\boldsymbol{x}$ and y are realizations of the random variables $\mathbf{x}$ and y, drawn from the unknown data distribution $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})\, p(\mathbf{x})$. The training data consists of N feature-response pairs $\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$. Lastly, let $(\boldsymbol{x}^*, y^*)$ denote a test point, which may be drawn from a different distribution than the one used for training.

Early-Exit Neural Networks EENNs (Teerapittayanon et al., 2016) generate predictions at various depths by having several prediction heads branch out from a shared backbone network. Specifically, an EENN defines a sequence of predictive models: $f(\boldsymbol{x}; \boldsymbol{W}_t, \boldsymbol{U}_{1:t})$, $t = 1, \ldots, T$, where $\boldsymbol{W}_t$ represents the parameters of the predictive head at exit t and $\boldsymbol{U}_t$ denotes the parameters of the t-th block in the backbone architecture. EENNs are usually trained by fitting all exits at once

$$\mathcal{L}(\boldsymbol{W}_{1:T}, \boldsymbol{U}_{1:T}; \mathcal{D}) \coloneqq \sum_{n=1}^{N} \frac{1}{T} \sum_{t=1}^{T} \ell\big(y_n, f(\boldsymbol{x}_n; \boldsymbol{W}_t, \boldsymbol{U}_{1:t})\big)$$

where ℓ is a suitable loss function such as negative log-likelihood.

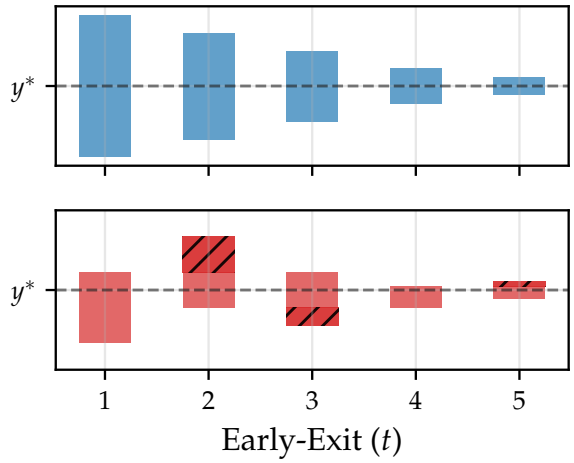


Figure 1: Illustrative example of a 1-dimensional regression problem using an Early-Exit neural network (EENN) with $T = 5$ exits. *Upper*: At each exit, the EENN produces an uncertainty estimate C_t consistent with its previous estimates, i.e., $C_t \subseteq C_{t-1}$. *Lower*: An example of inconsistent uncertainty estimates across different exits, e.g., C_2 contains candidate labels y not included in C_1 (area denoted with (／) lines). Such behavior often results from an EENN becoming overconfident, i.e., exhibiting low uncertainty, too early.

At test time, we can utilize the intermediate predictions of EENNs in various ways. For instance, if the model is deemed sufficiently confident at exit t, we can halt computation without propagating through blocks $t+1, \ldots, T$, thus speeding up prediction time. Naturally, the merit of such an approach relies on quality estimates of the EENN's uncertainty at every exit. Furthermore, EENNs can be employed as anytime predictors (Huang et al., 2018; Jazbec et al., 2023) where the aim is to quickly provide an approximate prediction—ideally with its associated uncertainty—and continuously improve upon it as long as the current environment permits.

Anytime-Valid Confidence Sequences Consider a streaming setting in which new data arrives at every time point t via sampling from an unknown (parametric) model $\boldsymbol{x}_t \sim p(\mathbf{x}|\theta^*)$. Here $\theta^* \in \mathbb{R}$ represents the parameter of the data-generating distribution for which we want to perform statistical inference. An anytime-valid confidence sequence (AVCS) (Robbins, 1967; 1970; Lai, 1976) for θ^* is a sequence of confidence intervals $C_t = (L_t, R_t) \subseteq \mathbb{R}$ that have time-uniform and non-asymptotic coverage guarantees:

$$\mathbb{P}(\forall t,\ \theta^* \in C_t) \geq 1 - \alpha,$$

where $\alpha \in (0, 1)$ represents the level of significance. The anytime (i.e. time-uniform) property allows the user to stop the experiment, 'peek' at the current results, and choose to continue or not, all while preserving the

validity of the statistical inference. This is in contrast with standard confidence intervals based on central limit theorem (CLT) that are valid only pointwise (i.e. for a fixed time / sample size). Of course the stronger theoretical properties of AVCSs come at a cost, as their size is typically larger compared to CLT-based intervals (Howard et al., 2021).

AVCS is constructed by first specifying the family of stochastic processes $\{R_t(\theta) : \theta \in \Theta\}$ that depends only on observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$ available at time t. Next, we require that when evaluated at the parameter of interest, $R_t(\theta^*)$ forms a discrete, non-negative *martingale* (Ramdas et al., 2020)—a stochastic process that remains constant in expectation:[1] $\mathbb{E}_{\mathbf{x}_{t+1}}[R_{t+1}(\theta^*)|\mathbf{x}_1, \ldots, \mathbf{x}_t] = R_t(\theta^*), \forall t$. Additionally, $R_0(\theta^*)$ should have an initial value that is constant (usually one). Once such a martingale is constructed, the AVCS at a given t is implemented by computing $R_t(\theta)$ for all $\theta \in \Theta$ and adding to the set the values for which R_t does not exceed $1/\alpha$: $C_t := \{\theta : R_t(\theta) \leq 1/\alpha\}$. Strong theoretical properties (i.e., time-uniformity) then follow from Ville's inequality for nonnegative (super)martingales: $\mathbb{P}(\exists t : R_t(\theta^*) \geq 1/\alpha) \leq \alpha$. One example of a random variable R_t from which we can construct an AVCS is the prior-posterior ratio: $R_t(\theta) = p(\theta)/p(\theta|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$ (Waudby-Smith & Ramdas, 2020). The time-uniform nature of AVCS enables one to consider the intersection of all previous intervals, given by $C_t = \cap_{s \leq t} C_s$, at time t without sacrificing statistical validity (Shekhar & Ramdas, 2023). This results in nested intervals/sets, i.e., $C_t \subseteq C_{t-1}$. It is this pivotal property of AVCSs that we wish to exploit to ensure that the predictive uncertainty in EENNs remains consistent across exits.

3 Confidence Sequences for Early-Exit NNs

Our contribution is to apply AVCSs to perform inference over the predictions generated by each exit of a EENN. As we will see, this is not a straightforward synthesis: AVCSs have been exclusively used in streaming-data settings, where the goal at every time step is to produce a *confidence interval* covering the parameter of the data generating distribution θ. On the other hand, we want to apply them to EENNs that see just one feature vector $\boldsymbol{x}^*$ at test time. Moreover, we are interested in obtaining a *prediction interval* at every exit that contains the ground-truth label y^* with high probability. We overcome these differences by considering the parameters of the EENN's exits $\boldsymbol{W}_t$ as the sequence of random variables for which the martingale is defined. Below we first give a general recipe for constructing AVCSs for EENNs and then describe a practical implementation for both regression (c.f. Section 4) as well as for classification models (c.f. Section 5).

Bayesian EENN We begin by positing a (last-layer) Bayesian predictive model at every exit:

$$p_t(\mathrm{y}|\boldsymbol{x}^*, \mathcal{D}) = \int p(\mathrm{y}|\boldsymbol{x}^*, \mathbf{W}_t, \boldsymbol{U}_{1:t})\ p(\mathbf{W}_t|\mathcal{D}, \boldsymbol{U}_{1:t})\ d\mathbf{W}_t \tag{1}$$

for $t = 1, \ldots, T$, with T representing the total number of exits. $p(\mathrm{y}|\boldsymbol{x}^*, \mathbf{W}_t, \boldsymbol{U}_{1:t})$ and $p(\mathbf{W}_t|\mathcal{D}, \boldsymbol{U}_{1:t})$ correspond to the likelihood and (exact) posterior distribution, respectively. To ensure minimal overhead of our approach at test time, we treat the backbone parameters $\boldsymbol{U}_{1:t}$ as point estimates (e.g. found through pre-training) that are held constant when constructing the AVCS. To reduce notational clutter, we omit these parameters from here forward. While Bayesian predictives $p_t(\mathrm{y}|\boldsymbol{x}^*, \mathcal{D})$ can be used 'as is' to get uncertainty estimates at each exit (e.g., by constructing a credible interval), we show in Section 7 that this results in an inconsistent, i.e., not nested, sequence of uncertainty estimates. We next present an approach based on AVCSs to rectify such behaviour.

Idealized Construction We first consider an idealized construction that, while impossible to implement exactly, will serve as the foundation of our approach. At test time, upon seeing a new feature vector $\boldsymbol{x}^*$, we wish to compute an interval for its label such that $y^* \in C_t\ \forall t$ with high probability. Assume that we also have observed the true label y^*. For the moment, ignore the circular reasoning that this is the very quantity for which we wish to perform inference. Furthermore, with $(\boldsymbol{x}^*, y^*)$ in hand, assume we can compute (exactly) the posterior for any exit's parameters: $p(\mathbf{W}_t|, \mathcal{D} \cup (\boldsymbol{x}^*, y^*))$. This distribution is the posterior

[1] It is also common to define AVCS in terms of *supermartingales*, which are stochastic processes that decrease in expectation over time: $\mathbb{E}_{\mathbf{x}_{t+1}}[R_{t+1}(\theta^*)|\mathbf{x}_1, \ldots, \mathbf{x}_t] \leq R_t(\theta^*), \forall t$.

update we would perform after observing the new feature-response pair. For notational brevity, we will denote $\mathcal{D}_* := \mathcal{D} \cup (\boldsymbol{x}^*, y^*)$ from now on.

To prepare for the proposition that follows, we define for a given $y \in \mathcal{Y}$ the *predictive-likelihood ratio*

$$R_t^*(y) \;:=\; \prod_{l=1}^{t} \frac{p_l(y|\boldsymbol{x}^*, \mathcal{D})}{p(y|\boldsymbol{x}^*, \boldsymbol{W}_l)}, \;\; \boldsymbol{W}_l \sim p(\mathbf{W}_l|\mathcal{D}_*)\,. \tag{2}$$

Note that only the likelihood terms in the denominator depend on the updated posterior (via samples $\boldsymbol{W}_l$), whereas the predictive terms in the numerator rely solely on training data (via $p(\mathbf{W}_l|\mathcal{D})$). The exact form of our proposed ratio in (2) is inspired by the aforementioned prior-posterior martingale (Waudby-Smith & Ramdas, 2020) yet modified for the predictive setting. We next state our key proposition that will serve as an inspiration for constructing AVCS for y^* in EENNs:

Proposition 1. *For a given test point $(\boldsymbol{x}^*, y^*)$, the predictive-likelihood ratio $R_t^*(y)$ in (2) is a non-negative martingale with $R_0^* = 1$ when evaluated at $y = y^*$. Moreover, the confidence intervals of the form $C_t^* := \{y \in \mathcal{Y} \mid R_t^*(y) \leq 1/\alpha\}$ are $(1-\alpha)$-confidence sequences for y^*, meaning that $\mathbb{P}(\forall t, y^* \in C_t^*) \geq 1 - \alpha$.*

The proof follows the standard procedure for deriving parametric confidence sequences; see Appendix A.1. We term the resulting confidence sequence an *EENN-AVCS*.

Realizable Relaxation Now we return to the aforementioned circular reasoning that we are performing inference for y^* while assuming we have access to it. In practice, we do not have access to y^* at test time, hence we can not compute $R_t^*(y)$ (and consequently C_t^*). As a workaround, we propose to approximate the updated posterior with the one based on the training data only

$$p(\mathbf{W}_t|\mathcal{D}_*) \approx p(\mathbf{W}_t|\mathcal{D}) \tag{3}$$

at every exit $t = 1, \ldots, T$. With $R_t(y)$ and C_t, we denote the resulting predictive-likelihood ratio and confidence sequence based on $p(\mathbf{W}_t|\mathcal{D})$, respectively. While C_t is now computable in a real-world scenario (since it's independent of y^*), it unfortunately does not inherit the statistical validity of C_t^*. Naturally, the degree to which C_t violates validity depends on the quality of approximation in (3). If the posterior distribution $p(\mathbf{W}_t|\mathcal{D})$ is stable—meaning that adding a single new data point $(\boldsymbol{x}^*, y^*)$ would have a minimal effect on it—the approximation is well-justified, and only minor validity violations can be expected. Such stability in the posterior is likely when the training dataset $\mathcal{D}$ is large and the new test datapoint originates from the same distribution. Conversely, if the posterior is unstable, the approximation will likely be poor, leading to larger violations of validity. This intuition can be formalized using the following proposition:

Proposition 2. *Assume C_t^* is a valid $(1-\alpha)$ confidence sequence for a given test datapoint $(\boldsymbol{x}^*, y^*)$ (c.f. Proposition 1). Then the miscoverage probability of the confidence sequence $C_t := \{y \in \mathcal{Y} \mid R_t(y) \leq 1/\alpha\}$ can be upper bounded by*

$$P(\exists l \in \{1, \ldots, t\}, y^* \notin C_l) \leq \alpha + \sqrt{1 - e^{-\sum_{l=1}^{t} KL\left(p(\mathbf{W}_l|\mathcal{D}),\, p(\mathbf{W}_l|\mathcal{D}_*)\right)}}$$

$\forall t = 1, \ldots, T$, where KL denotes the Kullback-Leibler divergence between probability distributions.

See Appendix A.2 for the derivation. Based on the bound in Proposition 2, it is clear that when the posteriors at different exits are stable, i.e. the KL divergence between $p(\mathbf{W}_l|\mathcal{D})$ and $p(\mathbf{W}_l|\mathcal{D}_*)$ is small, the validity violation is minor. As a result, C_t will be a good approximation of the valid sequence C_t^*.

Detecting Violations of Posterior Stability It is also evident from Proposition 2, that when approximation in (3) is poor, i.e. the KL divergence between $p(\mathbf{W}_l|\mathcal{D})$ and $p(\mathbf{W}_l|\mathcal{D}_*)$ is large, the validity of C_t will degrade quickly. As aforementioned, this could happen for a particular $\boldsymbol{x}^*$ if either (i) $\mathcal{D}$ is small and the posterior is not stable yet or (ii) $\boldsymbol{x}^*$ is not drawn from the training distribution. In such cases, the method should ideally fail gracefully. Fortunately, the behavior of invalid AVCSs—ones for which $R_t(y)$ is not a martingale for all $y \in \mathcal{Y}$—has been previously studied for change-point detection (Shekhar & Ramdas,

2023). Based off of their theoretical and empirical results, our procedure should collapse to the empty interval if approximation (3) is poor: $\exists t_0$ such that $C_{t \geq t_0} = \emptyset$. Encouragingly, in Section 7.1, we experimentally validate that such collapses occur for out-of-distribution points for a reasonably small t_0. However, there will be times at which the interval width will be small—which the user might interpret as high confidence—only to later collapse to the empty set (meaning maximum uncertainty). We leave to future work a method for diagnosing when an EENN-AVCS has not yet collapsed but is likely to.

Speeding up convergence of EENN-AVCS In our original formulation (c.f. Eq. (2)), we draw a single weights sample $\boldsymbol{W}_t$ at each exit. Hence, in the first few exits, our confidence sequence will be based on only a few samples, and is thus expected to be large and potentially uninformative. This is analogous to AVCSs being large for the first few datapoints in the data-streaming scenario (Howard et al., 2021). As a workaround, we propose constructing multiple AVCSs in parallel for a given test datapoint $\boldsymbol{x}^*$, and then considering their intersection at a given exit. Importantly, due to its fully parallel nature, such an approach does not introduce additional time overhead. We have also considered alternative approaches, like constructing a single AVCS based on multiple samples at each exit; however, we found that they perform worse in terms of marginal coverage and efficiency. For a further discussion on speeding up the convergence of EENN-AVCS, refer to Appendix C.1.

4 EENN-AVCS for Regression

We next consider a concrete instantiation of our EENN-AVCS procedure proposed in the previous section. We focus on the case of one-dimensional Bayesian regression as it allows for exact inference due to conjugacy. This allows us to assess the quality of approximation (3) without introducing the additional challenge of approximate inference. We summarize our approach for obtaining AVCSs in EENNs in Algorithm 1.

Bayesian Linear Regression Recall from Section 3 that since we require fast and exact Bayesian inference, we keep EENN's backbone parameters $\boldsymbol{U}_t$ fixed and give only the weights $\boldsymbol{W}_t$ of the prediction heads a Bayesian treatment. We define the predictive model at the tth exit as a linear model $f(\boldsymbol{x}; \boldsymbol{W}_t, \boldsymbol{U}_{1:t}) = h_t(\boldsymbol{x})^T \boldsymbol{W}_t$ where $h_t(\cdot\,; \boldsymbol{U}_{1:t}) : \mathcal{X} \to \mathbb{R}^H$ represents the output of the first t backbone layers or blocks. We use a Gaussian likelihood and prior:

$$\mathrm{y} \sim \mathcal{N}\left(\mathrm{y}; h_t(\boldsymbol{x})^T \boldsymbol{W}_t, \sigma_t^2\right), \quad \boldsymbol{W}_t \sim \mathcal{N}\left(\mathbf{W}_t; \hat{\boldsymbol{W}}_t, \sigma_{w,t}^2 \mathbb{I}_H\right),$$

where σ_t^2 is the observation noise, $\sigma_{w,t}^2$ is the prior's variance, and $\hat{\boldsymbol{W}}_t$ are the prediction weights obtained during (pre)training of the EENN. Due to conjugacy, we can obtain a closed form for the posterior and predictive distributions:

$$p(\mathbf{W}_t|\mathcal{D}) = \mathcal{N}\left(\mathbf{W}_t; \bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t\right), \quad p_t(\mathrm{y}|\boldsymbol{x}^*, \mathcal{D}) = \mathcal{N}\left(\mathrm{y}; h_t(\boldsymbol{x}^*)^T \bar{\boldsymbol{\mu}}_t, v_* + \sigma_t^2\right), \tag{4}$$

where $v^* := h_t(\boldsymbol{x}^*)^T \bar{\boldsymbol{\Sigma}}_t h_t(\boldsymbol{x}^*)$. See Appendix A.3 for exact expressions for posterior parameters $\bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t$. To estimate σ_t^2 and $\sigma_{w,t}^2$, we optimize the (exact) marginal likelihood on the training data (type-II maximum likelihood). Combining the obtained Bayesian quantities, we can compute the predictive-likelihood ratio in (2) at every exit.

Solving for Interval Endpoints To construct C_t, we next have to evaluate R_t at every $y \in \mathcal{Y}$ and discard those where the ratio exceeds $1/\alpha$, with α representing a significance level (e.g., 0.05). However, in the case of regression, where the output space is continuous, the method of evaluation is not immediately clear. One possible approach would be to define a grid of points over $\mathcal{Y}$ and then evaluate the predictive-likelihood ratio using a finite number of labels. Fortunately, the Bayesian linear regression model above allows us to obtain the endpoints of the confidence interval, at all times, via a closed-form expression: $C_t = [y_L^t, y_R^t]$. This is computationally valuable since it eliminates the overhead of iterating over $\mathcal{Y}$, which could be prohibitively expensive in the low-resource settings in which EENNs typically operate. To arrive at the analytical form, we first observe that $\log R_t$ represents a convex quadratic function in y:

$$\log R_t(y) = \alpha_t(\boldsymbol{x}^*) \cdot y^2 + \beta_t(\boldsymbol{x}^*, \boldsymbol{W}_{1:t}) \cdot y + \gamma_t(\boldsymbol{x}^*, \boldsymbol{W}_{1:t}).$$

Expressions for the coefficients $\alpha_t, \beta_t, \gamma_t$ are provided in Appendix A.4. To obtain the bounds y_L^t, y_R^t of the confidence interval at the tth exit, we then simply need to find the roots of the quadratic equation $\log R_t(y) - \log(1/\alpha) = 0$. If the discriminant $\beta_t^2 - 4\alpha_t(\gamma_t + \log \alpha)$ is negative, the equation has no real-valued roots, resulting in an empty confidence interval. In such cases, we interpret $\boldsymbol{x}^*$ as an out-of-distribution sample, as mentioned in Section 3.

Epistemic Uncertainty In our assumed Bayesian linear regression scenario, both the posterior and updated posterior are Gaussian. This allows us to derive a closed-form expression for the KL term $KL\big(p(\mathbf{W}_t|\mathcal{D}),\ p(\mathbf{W}_t|\mathcal{D}_*)\big)$ in the upper bound from Proposition 2. See Appendix A.5 for the derivation. We observe that KL remains small for a given $\boldsymbol{x}_*$ when v_* is small. Recall that v_* represents the epistemic uncertainty in a Bayesian regression model (c.f. Eq. (4)), meaning the uncertainty stemming from the fact that we observe limited data. Such uncertainty decreases the more data we collect[2], which, together with Proposition 2, implies that the statistical coverage of our EENN-AVCS will improve with the increasing dataset size. Additionally, v^* is independent of the test label y^*. Thus, we can employ it as a measure of the stability of EENN-AVCS, a point we illustrate in Section 7.1.

5 EENN-AVCS for Classification

In this section, we propose a concrete instantiation of our EENN-AVCS for classification tasks. Unlike the regression scenario in the previous section, an additional challenge is presented by a lack of conjugacy. Specifically, we cannot obtain a closed-form expression for the Bayesian predictive posterior (see Eq. (1)) at every exit when using the usual Gaussian assumption for the posterior over parameters. To circumvent this, we utilize Dirichlet Prior Networks (DPN; Malinin & Gales (2018)), which enable analytically tractable predictive distributions at each exit. Our EENN-AVCS approach for classification is summarized in Algorithm 2.

Dirichlet Prior Networks Instead of positing a distribution over (last-layer) weights $\mathbf{W}_t$ at every exit, we posit a distribution over categorical distributions $p(\boldsymbol{\pi}_t|\mathcal{D}, \boldsymbol{x}^*)$, $\boldsymbol{\pi}_t \in \Delta^K$ [3] for a given test datapoint $\boldsymbol{x}^*$. Additionally, assuming the categorical likelihood and Dirichlet posterior

$$p(\mathrm{y}|\boldsymbol{\pi}_t) = \texttt{Cat}(\mathrm{y}|\boldsymbol{\pi}_t), \quad p(\boldsymbol{\pi}_t|\boldsymbol{x}^*, \mathcal{D}) = \texttt{Dir}(\boldsymbol{\pi}_t|\boldsymbol{\alpha}_t(\boldsymbol{x}^*; \mathcal{D}))$$

where $\boldsymbol{\alpha}_t \in \mathbb{R}^K_{>0}$ represents concentration parameters of the Dirichlet, the predictive distribution has a closed form solution

$$p_t(\mathrm{y} = y|\boldsymbol{x}^*, \mathcal{D}) = \int p(\mathrm{y} = y|\boldsymbol{\pi}_t)\ p(\boldsymbol{\pi}_t|\boldsymbol{x}^*, \mathcal{D})\ d\boldsymbol{\pi}_t = \frac{\alpha_{t,y}}{\sum_{y' \in \mathcal{Y}} \alpha_{t,y'}}\ .$$

In Malinin & Gales (2018), the authors propose to parameterize the Dirichlet concentration parameters via the outputs of the neural network $\boldsymbol{\alpha}_t(\boldsymbol{x}^*; \mathcal{D}) = f(\boldsymbol{x}^*; \boldsymbol{W}_t, \boldsymbol{U}_{1:t})$ and term this approach Dirichlet Prior Networks (DPN). Note that in DPNs, the aim is to capture the so-called *distributional uncertainty* that arises due to the mismatch between test and training distributions, in addition to the so-called *data uncertainty* (often referred to as aleatoric uncertainty). This is in contrast to Bayesian models, which focus on the *model uncertainty* (or epistemic uncertainty) on top of the aleatoric one. We refer the interested reader to Malinin & Gales (2018) for a more in-depth discussion on the different sources of uncertainty.

Classification EENN-AVCS Having a closed-form predictive distribution, we can define the following *predictive-likelihood* ratio for a given $y \in \mathcal{Y}$:

$$R_t^*(y)\ :=\ \prod_{l=1}^{t} \frac{p_l(y|\boldsymbol{x}^*, \mathcal{D})}{p(y|\boldsymbol{\pi}_l)},\ \ \boldsymbol{\pi}_l \sim p(\boldsymbol{\pi}_l|\mathcal{D}^*)\ .$$

[2] $\lim_{N\to\infty} v_* = 0$ where N represents the number of training datapoints (c.f. Section 3.3.2 in Bishop & Nasrabadi (2006)).
[3] $\Delta^K := \{\boldsymbol{\pi} \in \mathbb{R}^K | \sum_{k=1}^K \mu_k = 1, \mu_k \geq 0\}$

Our result from Proposition 1 applies here as well[4], hence it follows that $C_t^* := \{y \in \mathcal{Y} \mid R_t^*(y) \leq 1/\alpha\}$ is a valid $(1-\alpha)$-confidence sequences for y^*. As in the regression case, R_t^* can not be realized in practice as it depends on the unknown label y^*. To address this, we again approximate the posterior over categorical distributions with the one based solely on the training data $p(\boldsymbol{\pi}_l|\mathcal{D}^*) \approx p(\boldsymbol{\pi}_l|\boldsymbol{x}^*, \mathcal{D})$ and denote the resulting predictive-likelihood ratio and confidence sequence as R_t and C_t, respectively. To reason about the quality of this approximation, we can again rely on the Proposition 2.

Post-hoc Implementation While the original DPN formulation (Malinin & Gales, 2018) requires a specialized training procedure to ensure that the NN's outputs represent meaningful concentration parameters of Dirichlet distributions, we opt for a simpler *post-hoc* approach as we have found that it yields satisfactory results for the purposes of our study. Specifically, to obtain the concentration parameters, we start with a pretrained (classification) EENN and pass the logits at each exit through an activation function $a : \mathbb{R} \to \mathbb{R}_{>0}$. We found that a simple choice of ReLU activation $a_t(x) = \texttt{ReLU}(x, \tau_t)$ with a different threshold $\tau_t \geq 1$ at each exit works well in practice.[5] To obtain the ReLU thresholds, we use a validation dataset and pick the largest τ_t such that $(1-\alpha)\%$ of validation datapoints is still contained in the resulting confidence sets at each exit. Lastly, since $\mathcal{Y}$ has a finite support, contrary to the regression case, we iterate over all of $\mathcal{Y}$ when constructing a confidence set C_t at a given exit.

6 Related Work

Early-Exit Neural Networks (EENNs) enable quicker predictions in deep models by allowing predictions at intermediate layers (Teerapittayanon et al., 2016; Huang et al., 2018; Laskaridis et al., 2021). They have been extensively explored for various computer vision (Li et al., 2019; Kaya et al., 2019; Yang et al., 2023) and natural language processing (Schwartz et al., 2020; Zhou et al., 2020; Xu & McAuley, 2022) tasks. The majority of these studies have aimed at enhancing the accuracy-speed trade-off, i.e., ensuring the model exits as early as possible with minimal accuracy loss. However, the aspect of uncertainty quantification within EENNs has so far garnered relatively little attention (Schuster et al., 2021; Meronen et al., 2023; Regol et al., 2023). It has primarily been leveraged to devise better termination criteria. In Meronen et al. (2023), the authors employ a Bayesian predictive model at each exit to enhance the calibration of EENNs. In Schuster et al. (2021), a novel conformal prediction scheme is introduced with the goal of outputting conformal sets/intervals that are marginally guaranteed to contain the prediction of the full EENN. In their work, consistency refers to the property of conformal sets at intermediate exits containing the full model prediction, differing from our focus on nested uncertainty estimates across exits. Yet, none of the preceding works address the fact that uncertainty estimates at successive exits are dependent, which is the main focus of our work.

Anytime-Valid Confidence Sequences (AVCSs) are sequences of confidence intervals designed for streaming data settings, providing time-uniform and non-asymptotic coverage guarantees (Robbins, 1967; Lai, 1976; Howard et al., 2021). They allow for adaptive experimentation where one can 'peek' at the data at any time, make decisions, yet still maintain the validity of the statistical inferences. Recently, AVCSs have found applications in A/B testing that is resistant to 'p-hacking' (Maharaj et al., 2023), Bayesian optimization (Neiswanger & Ramdas, 2021)), and change-point detection (Shekhar & Ramdas, 2023). AVCSs have not been considered before for sequential estimation of predictive uncertainty in EENNs.

7 Experiments

We conduct three sets of experiments and publicly release our code at `GITHUB_REPO`. Initially, in Section 7.1, we explore our method (EENN-AVCS) on synthetic datasets to empirically verify its correctness and assess its feasibility. In the subsequent set of experiments, detailed in Section 7.2, we check that our findings extend to more practical scenarios by applying our proposed EENN-AVCS to semantic textual similarity regression

[4]The only difference in the proof being that the martingale is defined with respect to the sequence of categorical distributions $\boldsymbol{\pi}_t$ instead of the sequence of weights $\mathbf{W}_t$.

[5]Following standard practice, we impose concentration parameters to be larger than 1 due to a strange behavior of the Dirichlet distribution for values of concentration parameters between 0 and 1.

task using a large transformer backbone model (Zhou et al., 2020). Finally, in Section 7.3 we report results on the image classification tasks (CIFAR-10/100, ImageNet) using MSDNet (Huang et al., 2018).

Evaluation metrics To assess the quality of uncertainty estimates at each exit, we utilize the standard combination of *marginal coverage* and *efficiency*, i.e. average interval size, on the test dataset. Marginal coverage serves as a proxy of the statistical validity of the approach, illustrating how frequently the ground-truth falls within the predicted interval on average. Note that among two methods with comparable marginal coverage, the one with smaller interval sizes is preferred. While traditionally, AVCSs aim for a stronger form of conditional coverage, the practical necessity of utilizing approximations of valid AVCS C_t^*—where conditional coverage isn't guaranteed (c.f. Section 3)—is a reason why we focus on the marginal level in our evaluations. To assess the alignment or consistency of uncertainty estimates between different exits, we define a *consistency* metric: at each exit t, we compute $\mathfrak{C}(t) = |\cap_{s \leq t} C_s|/|C_t|$ and report its mean across test data points. A perfectly consistent model with nested uncertainty estimates will have $\mathfrak{C}(t) = 1$, exactly. Otherwise, $\mathfrak{C}(t)$ will be less than one and zero only in the case of disjoint intervals.

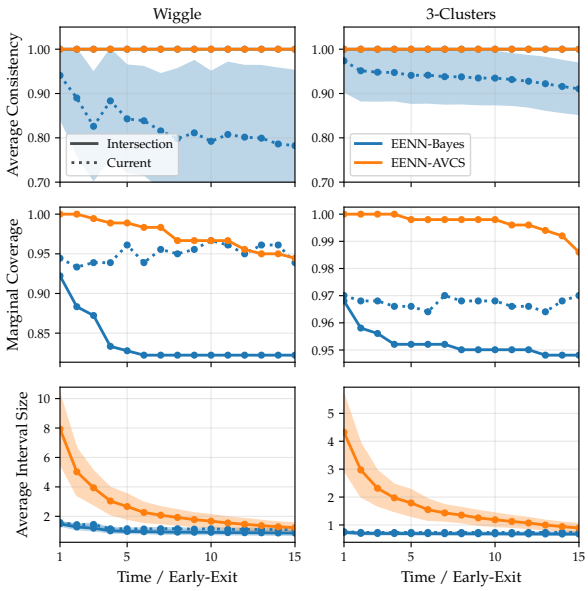


Figure 2: We compare our EENN-AVCS with EENN-Bayes baseline based on average consistency (*top*), marginal coverage (*middle*), and average interval size (*bottom*). EENN-AVCS is the only approach that yields perfect consistency while maintaining reasonably high marginal coverage across exits. The consistency comes at a price of larger intervals in the initial exits, though. Note that in the *top* plot, the consistency curves of EENN-AVCS (—) and EENN-Bayes-intersection (—) overlap at $\mathfrak{C}(t) = 1$.

Baselines As a baseline in our regression experiments, we use the same underlying Bayesian EENN but without the AVCS applied. We term this approach *EENN-Bayes* since it uses the Bayesian predictive distribution at each exit to perform uncertainty quantification. For the baseline in classification tasks, we perform conformal inference independently at every exit. Specifically, we use Regularized Adaptive Predictive Sets algorithm (RAPS; Angelopoulos et al., 2021) to compute conformal scores. The primary difference between our approach and the baselines should be the *consistency* of the intervals across exits. EENN-AVCS have nested intervals whereas EENN-Bayes and RAPS have no such guarantees, which could lead to non-overlapping intervals.

7.1 Synthetic Regression Data

We use two non-linear regression simulations (Antorán et al., 2020): *wiggle* and *3-clusters*. We define the EENN used in this experiment via a backbone architecture of $T = 15$ feed-forward layers with residual connections. Each layer consists of $M = 20$ hidden units, and we attach an output layer on top of it to enable early exiting. We fit the (last-layer) Bayesian linear regression model at each exit using the training data and construct $S = 10$ confidence sequences in parallel at test time for each datapoint. We set the significance level to $\alpha = 0.05$ for EENN-AVCS, while for EENN-Bayes, we plot intervals that capture 2 standard deviations away from the predicted mean. Further details regarding data generation, the model architecture, and the training can be found in Appendix B.1.

In the *top* row of Figure 2, we compare our EENN-AVCS (—) with EENN-Bayes (···) baseline on the test dataset based on how consistent the confidence intervals are across exits. Due to their nested construction, EENN-ACVSs attain perfect consistency. In contrast, EENN-Bayes exhibits deteriorating consistency over time on both datasets considered, indicating that there indeed are labels that reenter the EENN-Bayes confidence intervals after being ruled out at some earlier exits. In the *top* row,

we additionally observe that perfect consistency can be achieved in EENN-Bayes by considering a running intersection of all previous confidence intervals at each exit (denoted with (—) line), similar to EENN-AVCS (the two consistency lines of both intersection methods overlap at $\mathfrak{C}(t) = 1$). However, as shown in the *middle* row, this approach leads to a decrease in marginal coverage, indicating that fewer data points are covered by the intersection of EENN-Bayes intervals as more exits are evaluated. In contrast, EENN-AVCS maintains high marginal coverage despite utilizing an intersection of intervals at each exit. This is a direct consequence of the time-uniform nature of AVCS. The consistency of EENN-AVCS comes at a price, though, as the interval size tends to be larger than that of EENN-Bayes at the initial exits (*lower* plot). This observation is in line with existing work on AVCSs (Howard et al., 2021).

To better understand the differing behaviors of our method on in-distribution (ID) points as compared to out-of-distribution (OOD) points, we construct a new test dataset by considering equidistantly spaced points across the entire $\mathcal{X}$ space[6]. We report results for both datasets considered in Figure 3. Initially, we observe that for ID datapoints (with ID regions of $\mathcal{X}$ depicted using (■) background), our method satisfactorily covers the data distribution, especially at later exits. Encouragingly, AVCSs are also observed to quickly collapse to empty intervals outside of the data distribution (OOD regions are depicted with a white background, and whenever AVCS collapses to an empty interval, we highlight this by altering the mean prediction line from (—) to (- -)). Recall that in our setting, an empty interval represents a distribution shift has been detected (i.e. maximal predictive uncertainty), which is exactly the desired behavior in OOD regions.

On the *wiggle* dataset, we also have the opportunity to study the behavior for the so-called in-between (IB) datapoints that reside between ID and OOD regions (we depict the IB region with a (■) background). We observe that our method encounters challenges in this regime to some extent, as the confidence intervals are, counterintuitively, smaller compared to those in the ID region despite the density of observed training datapoints being lower in the IB area. A partial remedy is provided by the epistemic uncertainty v^* (see Eq. (4)), which in our framework can be interpreted as a proxy for the stability of posterior distributions at different exits as aforementioned in Section 4. As depicted in Figure 3, v^* is larger for IB points compared to the ID ones (as

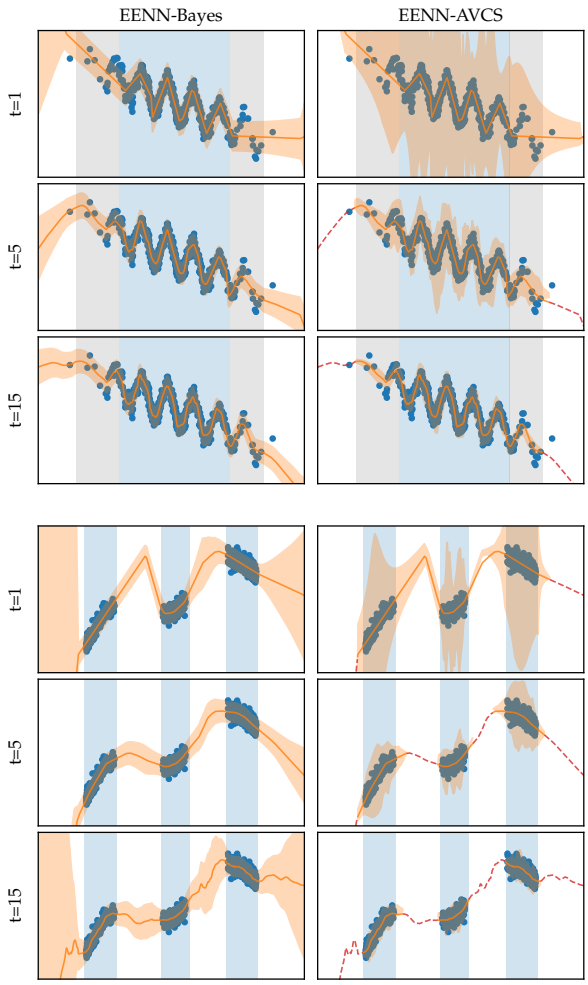


Figure 3: Predicted confidence intervals (■) for EENN-Bayes (*left*) and our EENN-AVCS (*right*) on two simulated regression tasks Antorán et al. (2020): wiggle (*up*) and 3-clusters (*bottom*). Blue points denote training data. In cases where the EENN-AVCS collapses to an empty set (out-of-distribution) we depict the mean prediction by a red dashed line (- -). We set the significance level to $\alpha = 0.05$ for EENN-AVCS, while for EENN-Bayes, we plot intervals that capture 2 standard deviations away from the predicted mean (—). With different background colors we denote different regions of data distribution, see Section 7.1.

[6]Specifically, for $\mathcal{X} = [L, R]$, we construct $X_{test} = \texttt{np.linspace}(L - \epsilon, R + \epsilon, N_{test})$ for $\epsilon > 0$.

expected). Thus, a higher v^* can serve as a warning to the user that the resulting confidence sequence should not be blindly relied upon.[7]

7.2 Semantic Textual Similarity(STS-B)

In this experiment, we examine the STS-B dataset from the GLUE Benchmark (Wang et al., 2018). The model receives two sentences as input, with the objective of predicting the semantic similarity between them. The similarity score is a continuous label ranging between 0 and 5, denoted as $\mathcal{Y} = [0, 5]$. As the backbone model, we employ an ALBERT model with 24 transformer layers (Lan et al., 2019), providing the model an option to early exit after every layer. Bayesian linear regression models are fitted on the development set. At test time, we construct a single AVCS ($S = 1$) with $\alpha = 0.05$ as we observed that constructing multiple AVCSs in parallel leads to a quicker decay of marginal coverage on this dataset. Since we know that the true label is bounded to the $[0, 5]$ range, we clip the resulting prediction intervals for all approaches to this region (in case they extend beyond it). Refer to Appendix B.2 for additional details on data, model, and training for this experiment.

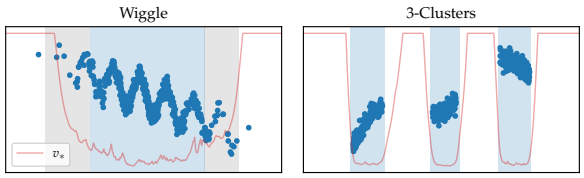


Figure 4: Average epistemic uncertainty v_* (—) across Bayesian linear regression models at different exits. As expected, v_* is larger in the regions where we observe less training data: *out-of-distribution* (denoted with a white background) and *in-between* (denoted with a grey background (■)). Hence, v_* can serve as an indicator for assessing the reliability of EENN-AVCSs.

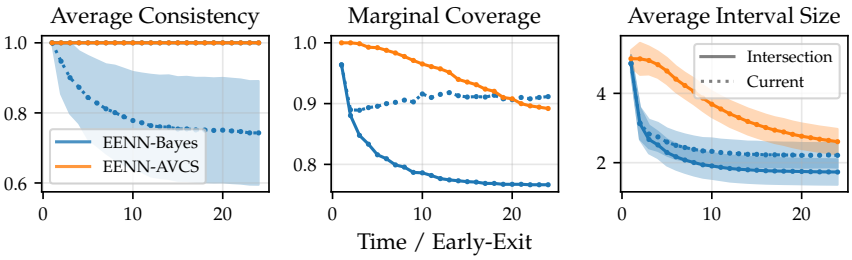


Figure 5: Comparison of our EENN-AVCS with EENN-Bayes baseline on STS-B dataset. Similar to findings on the synthetic data (c.f., Figure 2), EENN-AVCS attains perfect consistency (*left* plot) while maintaining reasonably high marginal coverage across exits (*middle* plot). However, the intervals generated by EENN-AVCS at each exit are larger compared to the baseline (*right* row). Note that in the *left* plot, the consistency curves of EENN-AVCS (—) and EENN-Bayes-intersection (—) overlap at $\mathfrak{C}(t) = 1$.

Results are illustrated in Figure 5. Encouragingly, the observations here align qualitatively with those made on synthetic datasets in Section 7.1. In the *left* plot, the trend of inconsistent uncertainty estimates when considering only the current Bayesian interval at each exit is again evident. While intersecting the EENN-Bayes intervals rectifies this inconsistency, it leads to a larger decay in marginal coverage compared to our EENN-AVCS, as depicted in the *middle* plot. We note that the marginal coverage, in this case, is worse across all approaches when compared to the coverage observed on synthetic data experiments, c.f. Figure 2. We attribute this to a larger shift between training, development, and test data splits for the STS-B dataset, as evidenced by the difference in model performance on each of those splits (see Appendix B.2 for further details on this). Finally, the *right* plot reaffirms that the consistency of EENN-AVCS comes at the expense of larger intervals.

[7]Note that the IB region also poses challenges for some other uncertainty quantification methods; a similar behavior was recently reported in the context of sampling from a Gaussian process posterior using stochastic gradient descent (Lin et al., 2023), where the in-between region is referred to as the *extrapolation region*.

7.3 Image Classification with MSDNet

In our last set of experiments, we consider the task of quantifying uncertainty at every exit on an image classification task. We consider CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and ILSVRC 2012 (ImageNet; Deng et al. (2009)). As our backbone EENN, we employ a Multi-Scale Dense Network (MSDNet; Huang et al., 2018), which consists of stacked convolutional blocks. At each exit, we map the logits to concentration parameters of the Dirichlet distribution using the ReLU activation function, as discussed in Section 5. To find the exact ReLU thresholds at each exit, we allocate 20% of the test dataset as a validation dataset and evaluate the performance on the remaining 80%. We construct a single AVCS ($S = 1$) at each exit and use significance level $\alpha = 0.05$ both for EENN-AVCS and RAPS sets.

Based on the results in Figure 6, we observe that constructing conformal RAPS sets (⋯) at every exit independently leads to inconsistent behavior (see *top* row). Taking the intersection of RAPS sets (—) corrects this; however, as expected this leads to a violation of conformal marginal coverage guarantees (see *middle* row). Similarly, as in our regression experiments, our EENN-AVCS based on the Dirichlet Prior Networks (—) yields perfect consistency while maintaining high marginal coverage. In the *last* row, we also see that, EENN-AVCS sets are roughly two times (or less) larger than the conformal sets, which might be a reasonable price to pay for the consistency.

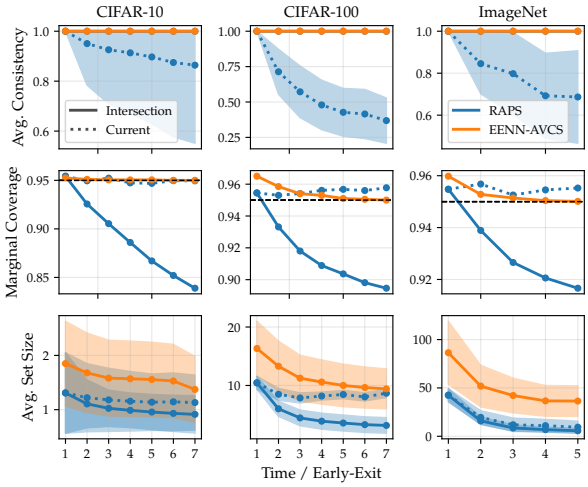


Figure 6: Comparison of our EENN-AVCS with RAPS (Angelopoulos et al., 2021) based on average consistency (*top*), marginal coverage (*middle*), and average interval size (*bottom*) for our image classification experiments using MSDNet as a backbone. EENN-AVCS is the only approach that attains perfect consistency (*top*) while maintaining high marginal coverage across different exits (*middle*). Consistency comes at a price, though, as EENN-AVCS sets are larger compared to the conformal ones (*bottom*). Note that in the *top* plot, the consistency curves of EENN-AVCS (—) and RAPS-intersection (—) overlap at $\mathfrak{C}(t) = 1$.

8 Conclusion

We proposed the use of anytime-valid confidence sequences for uncertainty quantification in predictive tasks using early-exit architectures. We showed that our approach yields consistent, i.e., nested, uncertainty estimates across exits—a property that is lacking in prior work, yet is cruical when deploying EENNs in safety critical applications. We described the theoretical and practical challenges associated with applying AVCSs in the predictive setting, and empirically validated our approach across a range of EENNs and different regression and classification datasets.

Limitations and Future Work For future work, it is paramount to improve the efficiency of EENN-AVCSs, aiming for smaller intervals, especially in the initial exits which are of highest practical interest for resource-constrained settings. While we explored some avenues to achieve this in the present study (c.f., Appendix C.1), further efforts are necessary to ensure faster convergence without sacrificing too much of marginal coverage in the process. Additionally, studying alternatives to our predictive-likelihood ratio (c.f. Eq. (2)) when constructing confidence sequences in the predictive settings might be promising for improved efficiency. Finally, it would be interesting to study the behaviour of EENN-AVCS in the infinite limit of the number of exits. Perhaps implicit deep learning models (Chen et al., 2018; Bai et al., 2020) could be used to this end.

References

Anastasios Nikolas Angelopoulos, Stephen Bates, Michael I. Jordan, and Jitendra Malik. Uncertainty sets for image classifiers using conformal prediction. In *9th International Conference on Learning Representations,*

ICLR, 2021.

Javier Antorán, James Urquhart Allingham, and José Miguel Hernández-Lobato. Depth uncertainty in neural networks. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

Shaojie Bai, Vladlen Koltun, and J Zico Kolter. Multiscale deep equilibrium models. *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Jean Bretagnolle and Catherine Huber. Estimation des densités: risque minimax. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 1979.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems, NeurIPS*, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2009.

Steven R Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform chernoff bounds via nonnegative supermartingales. 2020.

Steven R Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 2021.

Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q. Weinberger. Multi-scale dense networks for resource efficient image classification. In *6th International Conference on Learning Representations, ICLR*, 2018.

Metod Jazbec, James Urquhart Allingham, Dan Zhang, and Eric Nalisnick. Towards anytime classification in early-exit architectures by enforcing conditional monotonicity. *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, 2019.

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

Tze Leung Lai. On confidence sequences. *The Annals of Statistics*, 1976.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Stefanos Laskaridis, Alexandros Kouris, and Nicholas D Lane. Adaptive inference through early-exit networks: Design, challenges and directions. In *Proceedings of the 5th International Workshop on Embedded and Mobile Deep Learning*, 2021.

Hao Li, Hong Zhang, Xiaojuan Qi, Ruigang Yang, and Gao Huang. Improved techniques for training adaptive deep networks. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV*. IEEE, 2019.

Jihao Andreas Lin, Javier Antorán, Shreyas Padhy, David Janz, José Miguel Hernández-Lobato, and Alexander Terenin. Sampling from gaussian process posteriors using stochastic gradient descent. *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

Akash Maharaj, Ritwik Sinha, David Arbour, Ian Waudby-Smith, Simon Z Liu, Moumita Sinha, Raghavendra Addanki, Aaditya Ramdas, Manas Garg, and Viswanathan Swaminathan. Anytime-valid confidence sequences in an enterprise a/b testing platform. In *Companion Proceedings of the ACM Web Conference 2023*, 2023.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems, NeurIPS*, 2018.

Rowan McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: advantages of bayesian deep learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017.

Lassi Meronen, Martin Trapp, Andrea Pilzer, Le Yang, and Arno Solin. Fixing overconfidence in dynamic neural networks. *arXiv preprint arXiv:2302.06359*, 2023.

Willie Neiswanger and Aaditya Ramdas. Uncertainty quantification using martingales for misspecified gaussian processes. In *Algorithmic Learning Theory,*. PMLR, 2021.

Aaditya Ramdas, Johannes Ruf, Martin Larsson, and Wouter Koolen. Admissible anytime-valid sequential inference must rely on nonnegative martingales. *arXiv preprint arXiv:2009.03167*, 2020.

Florence Regol, Joud Chataoui, and Mark Coates. Jointly-learned exit and inference for a dynamic neural network: Jei-dnn. *arXiv preprint arXiv:2310.09163*, 2023.

Herbert Robbins. Confidence sequences for mean, variance, and median. *Proceedings of the National Academy of Sciences*, 1967.

Herbert Robbins. Statistical methods related to the law of the iterated logarithm. *The Annals of Mathematical Statistics*, 1970.

Tal Schuster, Adam Fisch, Tommi S. Jaakkola, and Regina Barzilay. Consistent accelerated inference via confident adaptive transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 2021.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A Smith. The right tool for the job: Matching model and instance complexities. *arXiv preprint arXiv:2004.07453*, 2020.

Shubhanshu Shekhar and Aaditya Ramdas. Sequential changepoint detection via backward confidence sequences. In *Proceedings of the 40th International Conference on Machine Learning, ICML*, 2023.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *23rd International Conference on Pattern Recognition, ICPR*. IEEE, 2016.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Ian Waudby-Smith and Aaditya Ramdas. Confidence sequences for sampling without replacement. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

Canwen Xu and Julian McAuley. A survey on dynamic neural networks for natural language processing. *arXiv preprint arXiv:2202.07101*, 2022.

Le Yang, Ziwei Zheng, Jian Wang, Shiji Song, Gao Huang, and Fan Li. : An adaptive object detection system based on early-exit neural networks. *IEEE Transactions on Cognitive and Developmental Systems*, 2023.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. Bert loses patience: Fast and robust inference with early exit. *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

A Supporting Derivations

A.1 Proof of Proposition 1

The proof can be divided into two steps. In the first step, we demonstrate that the predictive-likelihood ratio $R_t^*(y)$ in (2) is a non-negative martingale when evaluated at the true value y^*, with an initial value of one. In the second step, we utilize Ville's inequality to construct AVCS. Throughout this process, we closely adhere to the proof technique outlined in Waudby-Smith & Ramdas (2020) (refer to Appendix B.1 in that work).

We begin the first step by showing that the expectation of the predictive-likelihood ratio evaluated at y^* remains constant over time:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{W}_{t+1}}[R_{t+1}^*(y^*) \mid \mathbf{W}_1, \ldots, \mathbf{W}_t] = \\
&\int R_{t+1}^*(y^*)\, p(\mathbf{W}_{t+1}|\mathcal{D} \cup (\boldsymbol{x}^*, y^*))\, d\mathbf{W}_{t+1} \overset{(i)}{=} \\
&\int R_{t+1}^*(y^*)\, \frac{p(y^*|\boldsymbol{x}^*, \mathbf{W}_{t+1}) p(\mathbf{W}_{t+1}|\mathcal{D})}{p_{t+1}(y^*|\boldsymbol{x}^*, \mathcal{D})}\, d\mathbf{W}_{t+1} = \\
&\int R_t^*(y^*)\, p(\mathbf{W}_{t+1}|\mathcal{D})\, d\mathbf{W}_{t+1} = \\
&R_t^*(y^*) \int p(\mathbf{W}_{t+1}|\mathcal{D})\, d\mathbf{W}_{t+1} = \\
&R_t^*(y^*)\ ,
\end{aligned}$$

where we have invoked a Bayes rule on the (updated) posterior at (i), and additionally used $p(y^*|\mathcal{D}, \boldsymbol{x}^*, \mathbf{W}_{t+1}) = p(y^*|\boldsymbol{x}^*, \mathbf{W}_{t+1})$ and $p(\mathbf{W}_{t+1}|\boldsymbol{x}^*, \mathcal{D}) = p(\mathbf{W}_{t+1}|\mathcal{D})$. To show that initial value is equal to one, we proceed similarly:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{W}_1}[R_1^*(y^*)] = \\
&\int R_1^*(y^*)\, p(\mathbf{W}_1|\mathcal{D} \cup (\boldsymbol{x}^*, y^*))\, d\mathbf{W}_1 = \\
&\int R_1^*(y^*)\, \frac{p(y^*|\boldsymbol{x}^*, \mathbf{W}_1) p(\mathbf{W}_1|\mathcal{D})}{p_1(y^*|\boldsymbol{x}^*, \mathcal{D})}\, d\mathbf{W}_1 = \\
&\int p(\mathbf{W}_1|\mathcal{D})\, d\mathbf{W}_1 = 1 =: R_0^*\ .
\end{aligned}$$

In the second step, we make use of Ville's inequality, which provides a bound on the probability that a non-negative supermartingale exceeds a threshold $\beta > 0$.

$$\mathbb{P}\left(\exists t : R_t^*(y^*) \geq \beta\right) \leq \mathbb{E}[R_0^*(y^*)] \,/\, \beta\ .$$

Since every martingale is also a supermartingale, Ville's inequality is applicable in our case. Then, for a particular threshold $\alpha \in (0, 1)$ and since we have a constant initial value (one), Ville's inequality implies: $\mathbb{P}\left(\exists t : R_t^*(y^*) \geq 1/\alpha\right) \leq \alpha$. If we define the sequence of sets as $C_t^* := \{y \in \mathcal{Y} \mid R_t^*(y) \leq 1/\alpha\}$, their validity can be shown as

$$\begin{aligned}
&\mathbb{P}(\forall t, y^* \in C_t^*) = \mathbb{P}(\forall t, R_t^*(y^*) \leq 1/\alpha) = \\
&\quad 1 - \mathbb{P}(\exists t : R_t^*(y^*) \geq 1/\alpha) \geq 1 - \alpha\ ,
\end{aligned}$$

which concludes the proof.

A.2 Proof of Proposition 2

We first note that due to C_t^* being a valid $(1 - \alpha)$ confidence sequence, we have

$$P(\exists l \in [t], y^* \notin C_l^*) \leq P(\exists l \in [T], y^* \notin C_l^*) \leq \alpha\ , \tag{5}$$

where we adopt the notation $[t] := \{1, \ldots, t\}$ for brevity. Additionaly we observe that randomness in $P(\exists l \in [t], y^* \notin C_l)$ and $P(\exists l \in [t], y^* \notin C_l^*)$ comes from $p(\mathbf{W}_1, \ldots, \mathbf{W}_t|\mathcal{D})$ and $p(\mathbf{W}_1, \ldots, \mathbf{W}_t|\mathcal{D}_*)$, respectively. Hence, we can use total variation distance (TV) to upper bound the difference

$$\begin{aligned} P(\exists l \in [t], y^* \notin C_l) - P(\exists l \in [t], y^* \notin C_l^*) \leq \\ \left|P(\exists l \in [t], y^* \notin C_l) - P(\exists l \in [t], y^* \notin C_l^*)\right| \leq \\ TV\big(p(\mathbf{W}_1, \ldots, \mathbf{W}_t|\mathcal{D}),\, p(\mathbf{W}_1, \ldots, \mathbf{W}_t|\mathcal{D}_*)\big)\,. \end{aligned}$$

Next, we apply Bretangnolle and Huber inequality (Bretagnolle & Huber, 1979) to upper bound the TV distance in terms of KL divergence and use the fact that weights at different exits are independent which gives rise to a factorized joint distribution

$$\begin{aligned} TV\big(p(\mathbf{W}_1, \ldots, \mathbf{W}_t|\mathcal{D}),\, p(\mathbf{W}_1, \ldots, \mathbf{W}_t|\mathcal{D}_*)\big) \leq \\ \sqrt{1 - e^{-KL\left(p(\mathbf{W}_1,\ldots,\mathbf{W}_t|\mathcal{D}),\, p(\mathbf{W}_1,\ldots,\mathbf{W}_t|\mathcal{D}_*)\right)}} \leq \\ \sqrt{1 - e^{-\sum_{l=1}^{t} KL\left(p(\mathbf{W}_l|\mathcal{D}),\, p(\mathbf{W}_l|\mathcal{D}_*)\right)}} \end{aligned}$$

Rearranging the terms and using (5), the proposition follows

$$\begin{aligned} P(\exists l \in [t], y^* \notin C_l) \leq \\ P(\exists l \in [t], y^* \notin C_l^*) + \sqrt{1 - e^{-\sum_{l=1}^{t} KL_l}} \leq \\ \alpha + \sqrt{1 - e^{-\sum_{l=1}^{t} KL_l}} \end{aligned}$$

where $KL_l := KL\big(p(\mathbf{W}_l|\mathcal{D}),\, p(\mathbf{W}_l|\mathcal{D}_*)\big)$.

A.3 Bayesian Linear Regression

In Section 4, we define the predictive model at the tth exit as a linear model $f(\boldsymbol{x}; \boldsymbol{W}_t, \boldsymbol{U}_{1:t}) = h(\boldsymbol{x}; \boldsymbol{U}_{1:t})^T \boldsymbol{W}_t$. For notational brevity, we omit $\boldsymbol{U}_{1:t}$ and denote $h(\boldsymbol{x}; \boldsymbol{U}_{1:t})$ as $h_t(\boldsymbol{x})$ in this section. Additionally, let $\boldsymbol{y} = [y_1, \ldots, y_N]^T \in \mathbb{R}^N$ and $\boldsymbol{H}_t = [h_t(\boldsymbol{x}_1), \ldots, h_t(\boldsymbol{x}_N)]^T \in \mathbb{R}^{N \times H}$ represent a concatenation of training labels and (deep) features, respectively. Assuming a Gaussian likelihood $\mathcal{N}\left(\mathrm{y}; h_t(\boldsymbol{x})^T \mathbf{W}_t, \sigma_t^2\right)$ and a prior $\mathcal{N}\left(\mathbf{W}_t; \mathbf{0}, \sigma_{w,t}^2 \mathbb{I}_H\right)$, the posterior over weights $\mathbf{W}_t$ has the following form:

$$\begin{aligned} p(\mathbf{W}_t|\mathcal{D}) &= \mathcal{N}\left(\mathbf{W}_t; \bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t\right)\,, \\ \bar{\boldsymbol{\mu}}_t &= \frac{1}{\sigma_t^2} \bar{\boldsymbol{\Sigma}}_t \boldsymbol{H}_t^T \boldsymbol{y}\,, \\ \bar{\boldsymbol{\Sigma}}_t^{-1} &= \frac{1}{\sigma_t^2} \boldsymbol{H}_t^T \boldsymbol{H}_t + \frac{1}{\sigma_{w,t}^2} \mathbb{I}_H\,. \end{aligned}$$

Similarly, for a new test point $\boldsymbol{x}^*$, the posterior predictive can be obtained in a closed-form:

$$p_t(\mathrm{y}|\boldsymbol{x}^*, \mathcal{D}) = \mathcal{N}\left(\mathrm{y}; h_t(\boldsymbol{x}^*)^T \bar{\boldsymbol{\mu}}_t, h_t(\boldsymbol{x}^*)^T \bar{\boldsymbol{\Sigma}}_t h_t(\boldsymbol{x}^*) + \sigma_t^2\right)\,.$$

A.4 Solving for Interval Endpoints

Due to the assumed Bayesian linear regression model at each exit t, $\log R_t$ is a convex quadratic function in y:

$$\begin{aligned} \log R_t(y) = \\ \sum_{l=1}^{t} \log p_l(y|\boldsymbol{x}^*, \mathcal{D}) - \log p(y|\boldsymbol{x}^*, \boldsymbol{W}_l) = \\ \alpha_t(\boldsymbol{x}^*) \cdot y^2 + \beta_t(\boldsymbol{x}^*, \boldsymbol{W}_{1:t}) \cdot y + \gamma_t(\boldsymbol{x}^*, \boldsymbol{W}_{1:t})\,. \end{aligned}$$

Coefficients have the following form:

$$\alpha_t(\boldsymbol{x}^*) = \frac{1}{2}\sum_{l=1}^{t}\left(\frac{1}{\sigma_l^2} - \frac{1}{v_{*,l}+\sigma_l^2}\right),$$

$$\beta_t(\boldsymbol{x}^*, \boldsymbol{W}_{1:t}) = \sum_{l=1}^{t}\frac{h_l(\boldsymbol{x}^*)^T\bar{\boldsymbol{\mu}}_l}{v_l^*+\sigma_l^2} - \frac{h_l(\boldsymbol{x}^*)^T\boldsymbol{W}_l}{\sigma_l^2},$$

$$\gamma_t(\boldsymbol{x}^*, \boldsymbol{W}_{1:t}) =$$

$$\frac{1}{2}\sum_{l=1}^{t}\left(\frac{(h_l(\boldsymbol{x}^*)^T\boldsymbol{W}_l)^2}{\sigma_l^2} - \frac{(h_l(\boldsymbol{x}^*)^T\bar{\boldsymbol{\mu}}_l)^2}{v_l^*+\sigma_l^2} + \log\frac{\sigma_l^2}{v_l^*+\sigma_l^2}\right)$$

where $v_l^* := h_l(\boldsymbol{x}^*)^T\bar{\boldsymbol{\Sigma}}_l h_l(\boldsymbol{x}^*)$, and we provide expressions for $h_l, \bar{\boldsymbol{\mu}}_l, \bar{\boldsymbol{\Sigma}}_l$ in Appendix A.3. It is easy to show that $\alpha_t \geq 0$, from which the convexity follows.

To find AVCS $C_t = \{y \in \mathcal{Y} \mid R_t(y) \leq 1/\alpha\}$, we look for the roots of the equation $\log R_t(y) - \log(1/\alpha) = 0$. This yields an analytical expression for $C_t = [y_L^t, y_R^t]$:

$$y_{L,R}^t = \frac{-\beta_t \pm \sqrt{\beta_t^2 - 4\alpha_t\tilde{\gamma}_t}}{2\alpha_t}$$

where $\tilde{\gamma}_t = \gamma_t + \log\alpha$. See Figure 7 for a concrete example of log-ratios.

A.5 Epistemic Uncertainty and KL Divergence

To compute the KL divergence between the posterior and update posterior in the Bayesian linear regression model (c.f. Appendix A.3), we first use the Bayes rule to rewrite the latter as:

$$p(\mathbf{W}_t|\mathcal{D}_*) = \frac{p(y^*|\boldsymbol{x}^*, \mathbf{W}_t)\, p(\mathbf{W}_t|\mathcal{D})}{p_t(y^*|\boldsymbol{x}^*, \mathcal{D})}.$$

Using the definition of the KL divergence together with the formulas for posterior predictive and posterior distributions from Appendix A.3, we proceed as

$$KL\big(p(\mathbf{W}_t|\mathcal{D}),\, p(\mathbf{W}_t|\mathcal{D}_*)\big) =$$

$$\mathbb{E}_{p(\mathbf{W}_t|\mathcal{D})}\left[\log\frac{p(\mathbf{W}_t|\mathcal{D})}{p(\mathbf{W}_t|\mathcal{D}_*)}\right] =$$

$$\log p_t(y^*|\boldsymbol{x}^*, \mathcal{D}) - \mathbb{E}_{p(\mathbf{W}_t|\mathcal{D})}\big[\log p(y^*|\boldsymbol{x}^*, \mathbf{W}_t)\big] =$$

$$0.5\left(\log\big(\frac{\sigma_t^2}{\sigma_t^2+v_t^*}\big) + \big(\frac{1}{\sigma_t^2+v_t^*} - \frac{1}{\sigma_t^2}\big)r_*^2 + \frac{v_t^*}{\sigma_t^2}\right)$$

where $r_* = y^* - \bar{\boldsymbol{\mu}}_t^T h_t(\boldsymbol{x}^*)$ represents a residual, $v_* = h_t(\boldsymbol{x}^*)^T\bar{\boldsymbol{\Sigma}}_t h_t(\boldsymbol{x}^*)$ denotes epistemic uncertainty, and $\sigma = \sigma_{y,t}$. Based on the obtained expression, it is evident that a small v^*, implies small KL-divergence.

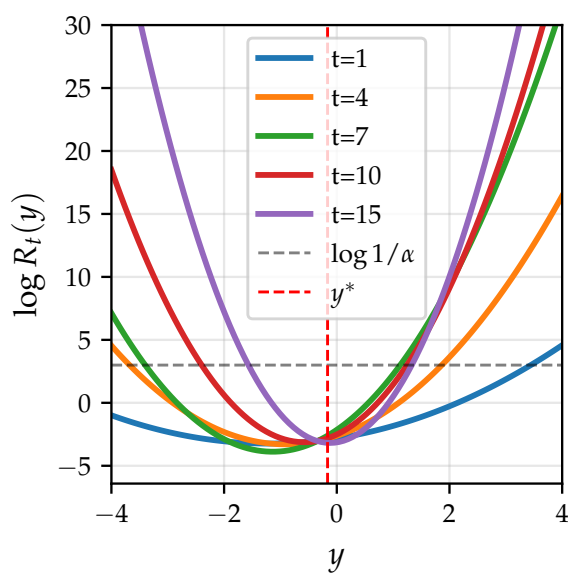


Figure 7: Plot of $\log R_t(y)$ at various exits t for a randomly selected test data point $(\boldsymbol{x}^*, y^*)$ from the *3-clusters* dataset. As described in Appendix A.4, we observe that the log-ratios exhibit a quadratic shape, allowing for an analytical solution for the endpoints of confidence intervals C_t.

B Implementation Details

B.1 Synthetic Data Experiments

Data Generation We closely follow data generation process from Antorán et al. (2020). Specifically, for *wiggle* dataset we sample N points from

$$y = \sin(\pi x) + 0.2\cos(4\pi x) - 0.3x + \epsilon$$

where $\epsilon \sim \mathcal{N}(0, 0.25)$ and $x \sim \mathcal{N}(5, 2, 5)$. For *3-clusters* dataset, we simulate data via

$$y = x - 0.1x^2 + \cos(x\pi/2)$$

where $\epsilon \sim \mathcal{N}(0, 0.25)$ and we sample $N/3$ points from $[-1, 0]$, $[1.5, 2.5]$ and $[4, 5]$, respectively. For both datasets, we sample a total of $N = 900$ points and allocate 80% of the data for training, while the remaining 20% constitutes the test dataset.

Model Architecture Our EENN is composed of an input layer and $T = 15$ residual blocks. The residual blocks consist of a `Dense` layer (with $M = 20$ hidden units), followed by a `ReLU` activation and `BatchNorm` (with default `PyTorch` parameters). We attach an output layer at each residual block to facilitate early exiting.

Training We train our EENN for 500 epochs using `SGD` with a learning rate of 1×10^{-3}, a momentum of 0.9, and a weight decay of 1×10^{-4}. For the loss function, we use the average mean-square error (`MSE`) across all exits.

B.2 STS-B Experiment

Dataset We use the STS-B dataset, the only regression dataset in the GLUE benchmark (Wang et al., 2018). The task is to measure the semantic similarity $y \in [0, 5]$ between the two input sentences. The training, development, and test datasets consist of 5.7K, 1.5K, and 1.4K datapoints, respectively.

Model Architecture and Training For the model architecture and training we reuse the code from Zhou et al. (2020). Specifically, we work with `ALBERT-large` which is a 24-layers transformer model. To facilitate early exiting, a regression head is attached after every transformer block.

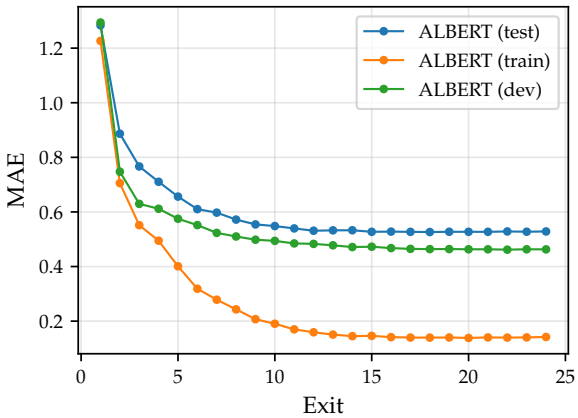


Figure 8: Mean Absolute Error (MAE) performance of the `ALBERT-large` model across different datasets: train, development (dev), and test. A large performance gap between the train and dev/test datasets is observed. Note that in our work, we reuse the exact model and training setup from previous approaches (Zhou et al., 2020).

EENN-AVCS In the results presented in the main text, we construct a single ($S = 1$) AVCS at test time with $\alpha = 0.05$. To fit the Bayesian linear regression models (i.e., empirical Bayes) at every exit, we use the development set. Note that this contrasts with our experiments on the synthetic dataset (c.f., Section 7.1) where we utilized the training dataset for this purpose. We observed that when fitting the regression model on the training dataset for STS-B, the noise parameters $\hat{\sigma}_t$ get underestimated, resulting in a rapid decay of marginal coverage for both EENN-AVCS and EENN-Bayes . We attribute this to a distribution shift present in the STS-B dataset, which is evident based on the different performances (MAE) that the ALBERT model achieves on different datasets, as seen in Figure 8.

C Additional Results

C.1 Speeding up convergence of EENN-AVCS

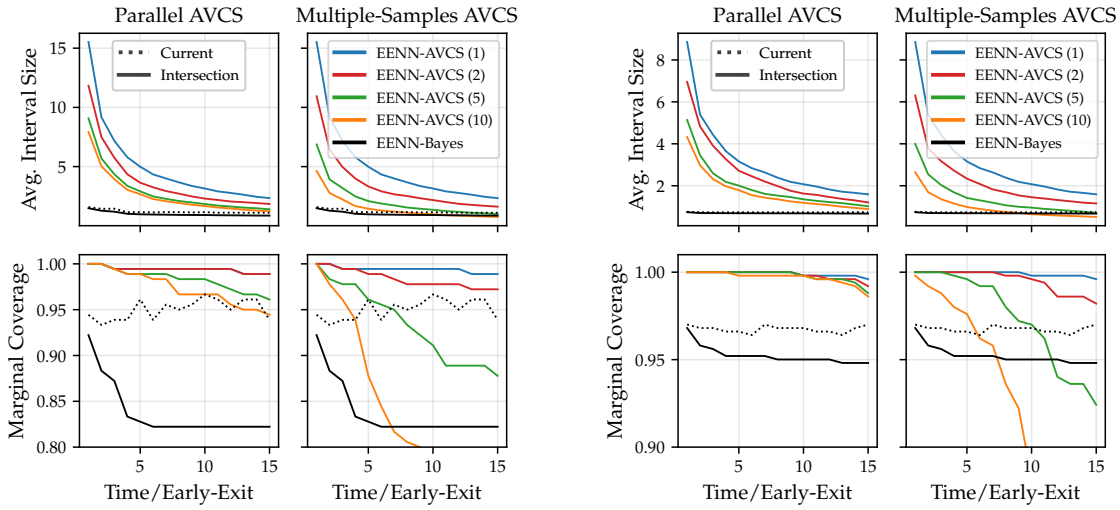
In our original formulation in Section 3, we draw a single sample of the weighs $\boldsymbol{W}_t$ (or predictive distribution $\boldsymbol{\mu}_t$ in the case of classification) at each exit. This invariably leads to large confidence intervals/sets at the initial exits - a phenomenon analogous to AVCSs being large for the initial few observed data points in the conventional data streaming scenario (Howard et al., 2020). In this section, we explore two distinct approaches to mitigate this issue, aiming to attain more efficient confidence estimates right from the initial exits.

In the first approach, we simply take multiple samples $S_t > 1$ at each exit. Consequently, the predictive likelihood ratio for a given test point $\boldsymbol{x}^*$ takes the following form:

$$R_t(y) \coloneqq \prod_{l=1}^{t} \prod_{s=1}^{S_l} \frac{p_l(y|\boldsymbol{x}^*, \mathcal{D})}{p(y|\boldsymbol{x}^*, \boldsymbol{W}_l^{(s)})}, \quad \boldsymbol{W}_l^{(s)} \sim p(\mathbf{W}_l|\mathcal{D}) \,.$$

We term this approach *Multiple-Samples AVCS*. As an alternative, we construct multiple AVCSs $\{C_t^{(s)}\}_{s=1}^{S_t}$ based on a single sample in parallel. At each exit, we then consider their intersection $C_t^{\cap} = \bigcap_{s=1}^{S_t} C_t^{(s)}$ and pass it on to the next exit. We refer to this method as *Parallel AVCS*.

We present the results for both approaches in Figure 9 using synthetic datasets from Section 7.1. While both methods yield more efficient, i.e., smaller, intervals in the initial exits (*top* row), it is interesting to observe that the *Multiple-Samples* approach leads to a much faster decay in marginal coverage compared to the *Parallel* one (see *bottom* row). We attribute this to the fact that by sampling multiple samples within a single confidence sequence at each exit, we are essentially 'committing' more to our approximation of the updated posterior (c.f., Eq. (3)), which results in larger coverage violations. Hence, we recommend using the *Parallel* approach when attempting to speed up the convergence of our EENN-AVCS . Nonetheless, we acknowledge that this area warrants further investigation, and we consider this an important direction for future work.



(a) Wiggle

(b) 3-Clusters

Figure 9: Average interval size and marginal coverage for regression synthetic datasets. While both of the considered approaches yield more efficient intervals (*top* row), the *Parallel* method is better at preserving high marginal coverage (*bottom* row). AVCS(S) denotes a confidence sequence based on S samples at each exit in the case of *Multiple-Samples*, and the sequence based on S parallel ones in the case of *Parallel*.

D EENN-AVCS Algorithm

Algorithm 1: EENN-AVCS Regression

input : Backbone EENN $\{h(\cdot|\boldsymbol{U}_{1:t})\}_{t=1}^{T}$,
Regression models $\{p(\mathbf{W}_t|\mathcal{D}),\ \hat{\sigma}_t^2\}_{t=1}^{T}$,
test datapoint $\boldsymbol{x}^*$, significance level α_S

output: AVCS for $\boldsymbol{x}^*$

$C_0 = \mathcal{Y}$
$\alpha, \beta, \gamma = 0, 0, \log \alpha_S$
for $t = 1, ..., T$ **do**
  $\boldsymbol{W}_t \sim p(\mathbf{W}_t|\mathcal{D}) = \mathcal{N}(\mathbf{W}_t|\bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t)$
  $v_t^* := h_t(\boldsymbol{x}^*)^T \bar{\boldsymbol{\Sigma}}_t h_t(\boldsymbol{x}^*)$
  # update coefficients of $\log R_t(y)$
  $\alpha \mathrel{+}= \frac{1}{2}\left(\frac{1}{\hat{\sigma}_t^2} - \frac{1}{v_t^* + \hat{\sigma}_t^2}\right)$
  $\beta \mathrel{+}= \frac{h_t(\boldsymbol{x}^*)^T \bar{\boldsymbol{\mu}}_t}{v_t^* + \hat{\sigma}_t^2} - \frac{h_t(\boldsymbol{x}^*)^T \boldsymbol{W}_t}{\hat{\sigma}_t^2}$
  $\gamma \mathrel{+}=$
   $\frac{1}{2}\Big(\frac{(h_t(\boldsymbol{x}^*)^T \boldsymbol{W}_t)^2}{\hat{\sigma}_t^2} - \frac{(h_t(\boldsymbol{x}^*)^T \bar{\boldsymbol{\mu}}_t)^2}{v_t^* + \hat{\sigma}_t^2} + \log \frac{\hat{\sigma}_t^2}{v_t^* + \hat{\sigma}_t^2}\Big)$
  # find the roots of quadratic equation
  $y_{L,R}^t = \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}$
  $C_t = C_{t-1} \cap [y_L^t, y_R^t]$
  if $C_t = \emptyset$ **then**
    return $\emptyset$ # OOD
return $\{C_t\}_{t=1}^{T}$

Algorithm 2: EENN-AVCS Classification

input : Backbone EENN $\{f(\cdot|\boldsymbol{U}_{1:t}, \boldsymbol{W}_t)\}_{t=1}^{T}$,
ReLU thresholds $\{\tau_t\}_{t=1}^{T}$,
test datapoint $\boldsymbol{x}^*$, significance level α_S

output: AVCS for $\boldsymbol{x}^*$

$C_0 = \mathcal{Y}$
$R = [1, \ldots, 1]$
for $t = 1, ..., T$ **do**
  # get concentration parameters, only keep classes that "survive" `ReLU`
  $\boldsymbol{\alpha}_t = \texttt{ReLU}(f(\boldsymbol{x}^*|\boldsymbol{U}_{1:t}, \boldsymbol{W}_t), \tau_t)$
  $\tilde{\boldsymbol{\alpha}}_t = \boldsymbol{\alpha}_t[\boldsymbol{\alpha}_t > 0]$
  $\boldsymbol{\pi}_t \sim \texttt{Dir}(\tilde{\boldsymbol{\alpha}}_t)$
  $S_t = \sum_k \alpha_{t,k}$
  # update the predictive-likelihood ratio
  for $k = 1, \ldots, K$ **do**
    if $\alpha_{t,k} > 0$ **then**
      $R[k] \mathrel{*}= \frac{\alpha_{t,k}/S_t}{\pi_{t,k}}$
    else
      $R[k] = \infty$
    if $R[k] \leq \frac{1}{\alpha_S}$ **then**
      C_t`.append`(k)
  $C_t = C_t \cap C_{t-1}$
  if $C_t = \emptyset$ **then**
    return $\emptyset$ # OOD
return $\{C_t\}_{t=1}^{T}$