PAC CONFIDENCE PREDICTIONS FOR DEEP NEURAL NETWORK CLASSIFIERS

Anonymous authors
Paper under double-blind review

ABSTRACT

A key challenge for deploying deep neural networks (DNNs) in safety critical settings is the need to provide rigorous ways to quantify their uncertainty. In this paper, we propose a novel algorithm for constructing predicted classification confidences for DNNs that comes with provable correctness guarantees. Our approach uses Clopper-Pearson confidence intervals for the Binomial distribution in conjunction with the histogram binning approach to calibrated prediction. In addition, we demonstrate how our predicted confidences can be used to enable downstream guarantees in two settings: (i) fast DNN inference, where we demonstrate how to compose a fast but inaccurate DNN with an accurate but slow DNN in a rigorous way to improve performance without sacrificing accuracy, and (ii) safe planning, where we guarantee safety when using a DNN to predict whether a given action is safe based on visual observations. In our experiments, we demonstrate that our approach can be used to provide guarantees for state-of-the-art DNNs.

1 INTRODUCTION

Due to the recent success of machine learning, there has been increasing interest in using predictive models such as deep neural networks (DNNs) in safety-critical settings, such as robotics (e.g., obstacle detection (Ren et al., 2015) and forecasting (Kitani et al., 2012)) and healthcare (e.g., diagnosis (Gulshan et al., 2016; Esteva et al., 2017) and patient care management (Liao et al., 2020)).

One of the key challenges is the need to provide guarantees on the safety or performance of DNNs used in these settings. The potential for failure is inevitable when using DNNs, since they will inevitably make some mistakes in their predictions. Instead, our goal is to design tools for *quantifying* the uncertainty of these models; then, the overall system can estimate and account for the risk inherent in using the predictions made by these models. For instance, a medical decision-making system may want to fall back on a doctor when its prediction is uncertain whether its diagnosis is correct, or a robot may want to stop moving and ask a human for help if it is uncertain to act safely. Uncertainty estimates can also be useful for human decision-makers—*e.g.,* for a doctor to decide whether to trust their intuition over the predicted diagnosis.

While many DNNs provide confidences in their predictions, especially in the classification setting, these are often overconfident. This phenomenon is most likely because DNNs are designed to overfit the training data (*e.g.,* to avoid local minima (Safran & Shamir, 2018)), which results in the predicted probabilities on the training data being very close to one for the correct prediction. Recent work has demonstrated how to *calibrate* the confidences to significantly reduce overconfidence (Guo et al., 2017). Intuitively, these techniques rescale the confidences on a held-out calibration set. Because they are only fitting a small number of parameters, they do not overfit the data as was the case in the original DNN training. However, these techniques do not provide theoretical guarantees on their correctness, which can be necessary in safety-critical settings to guarantee correctness.

We propose a novel algorithm for calibrated prediction in the classification setting that provides theoretical guarantees on the predicted confidences. We focus on *on-distribution* guarantees—*i.e.,* where the test distribution is the same as the training distribution. In this setting, we can build on ideas from statistical learning theory to provide *probably approximately correctness (PAC)* guarantees (Valiant, 1984). Our approach is based on a calibrated prediction technique called histogram binning (Zadrozny & Elkan, 2001), which rescales the confidences by binning them and then rescal-

ing each bin independently. We use Clopper-Pearson bounds on the tails of the binomial distribution to obtain PAC upper/lower bounds on the predicted confidences.

Next, we study how it enables theoretical guarantees in two applications. First, we consider the problem of speeding up DNN inference by composing a fast but inaccurate model with a slow but accurate model—*i.e.,* by using the accurate model for inference only if the confidence of the inaccurate one is underconfident (Teerapittayanon et al., 2016). We use our algorithm to obtain guarantees on accuracy of the composed model. Second, for safe planning, we consider using a DNN to predict whether or not a given action (*e.g.,* move forward) is safe (*e.g.,* do not run into obstacles) given an observation (*e.g.,* a camera image). The robot only continues to act if the predicted confidence is above some threshold. We use our algorithm to ensure safety with high probability. Finally, we evaluate the efficacy of our approach in the context of these applications.

Related work. Calibrated prediction (Murphy, 1972; DeGroot & Fienberg, 1983; Platt, 1999) has recently gained attention as a way to improve DNN confidences (Guo et al., 2017). Histogram binning is a non-parametric approach that sorts the data into finitely many bins and rescales the confidences per bin (Zadrozny & Elkan, 2001; 2002; Naeini et al., 2015). However, traditional approaches do not provide theoretical guarantees on the predicted confidences. There has been work on predicting confidence sets (*i.e.,* predict a set of labels instead of a single label) with theoretical guarantees (Park et al., 2020a), but this approach does not provide the confidence of the most likely prediction, as is often desired. There has also been work providing guarantees on the overall calibration error (Kumar et al., 2019), but this approach does not provide per-prediction guarantees.

There has been work speeding up DNN inference (Hinton et al., 2015). One approach is to allow intermediate layers to be dynamically skipped (Teerapittayanon et al., 2016; Figurnov et al., 2017; Wang et al., 2018), which can be thought of as composing multiple models that share a backbone. Unlike our approach, they do not provide guarantees on the accuracy of the composed model.

There has also been work on safe learning-based control (Akametalu et al., 2014; Fisac et al., 2019; Bastani, 2019; Li & Bastani, 2020; Wabersich & Zeilinger, 2018; Alshiekh et al., 2018); however, these approaches are not applicable to perception-based control. The most closely related work is Dean et al. (2019), which handles perception, but they are restricted to known linear dynamics.

2 PAC Confidence Prediction

In this section, we begin by formalizing the PAC confidence coverage prediction problem; then, we describe our algorithm for solving this problem based on histogram binning.

Calibrated prediction. Let $x \in \mathcal{X}$ be an example and $y \in \mathcal{Y}$ be one of a finite label set, and let D be a distribution over $\mathcal{X} \times \mathcal{Y}$. A *confidence predictor* is a model $\hat{f} : \mathcal{X} \to \mathcal{P}_{\mathcal{Y}}$, where $\mathcal{P}_{\mathcal{Y}}$ is the space of probability distributions over labels. In particular, $\hat{f}(x)_y$ is the predicted confidence that the true label for x is y. We let $\hat{y} : \mathcal{X} \to \mathcal{Y}$ be the corresponding *label predictor—i.e.,* $\hat{y}(x) \coloneqq \arg\max_{y \in \mathcal{Y}} \hat{f}(x)_y$—and let $\hat{p} : \mathcal{X} \to \mathbb{R}_{\geq 0}$ be corresponding *top-label confidence predictor—i.e.,* $\hat{p}(x) \coloneqq \max_{y \in \mathcal{Y}} \hat{f}(x)_y$. While traditional DNN classifiers are confidence predictors, a naively trained DNN is not reliable—*i.e.,* predicted confidence does not match to the true confidence; recent work has studied heuristics for improving reliability (Guo et al., 2017). In contrast, our goal is to construct a confidence predictor that comes with theoretical guarantees.

We first introduce the definition of calibration (DeGroot & Fienberg, 1983; Zadrozny & Elkan, 2002; Park et al., 2020b)—*i.e.,* what we mean for a predicted confidence to be "correct". In many cases, the main quantity of interest is the confidence of the top prediction. Thus, we focus on ensuring that the top-label predicted confidence $\hat{p}(x)$ is calibrated (Guo et al., 2017); our approach can easily be extended to providing guarantees on all confidences predicted using $\hat{f}$. Then, we say a confidence predictor $\hat{f}$ is *well-calibrated* with respect to distribution D if

$$\mathbb{P}_{(x,y)\sim D}\left[y = \hat{y}(x) \mid \hat{p}(x) = t\right] = t \qquad (\forall t \in [0,1]).$$

That is, among all examples x such that the label prediction $\hat{y}(x)$ has predicted confidence $t = \hat{p}(x)$, $\hat{y}(x)$ is the correct label for exactly a t fraction of these examples. Using a change of variables (Park

et al., 2020b), $\hat{f}$ being well-calibrated is equivalent to

$$\hat{p}(x) = c^*_{\hat{f}}(x) \coloneqq \mathbb{P}_{(x',y')\sim D}\left[y' = \hat{y}(x') \mid \hat{p}(x') = \hat{p}(x)\right] \qquad (\forall x \in \mathcal{X}). \tag{1}$$

Then, the goal of well-calibration is to make $\hat{p}$ equal to $c^*_{\hat{f}}$. Note that $\hat{f}$ appears on both sides of the equation $\hat{p}(x) = c^*_{\hat{f}}(x)$—implicitly in $\hat{p}$—which is what makes it challenging to satisfy. Indeed, in general, it is unlikely that (1) holds exactly for all x.

Instead, based on the idea of histogram binning (Zadrozny & Elkan, 2001), we consider a variant where we partition the data into a fixed number of bins and then construct confidence coverages separately for each bin. In particular, consider K bins $B_1, \ldots, B_K \subseteq [0, 1]$, where

$$B_k = \begin{cases} [0, b_1] & \text{if } k = 1 \\ (b_{k-1}, b_k] & \text{otherwise.} \end{cases}$$

Here, K and $0 \leq b_1 \leq \cdots \leq b_{K-1} \leq b_K = 1$ are hyperparameters. Given $\hat{f}$, let $\kappa_{\hat{f}} : \mathcal{X} \to \{1, \ldots, K\}$ to denote the index of the bin that contains $\hat{p}(x)$—*i.e.,* $\hat{p}(x) \in B_{\kappa_{\hat{f}}(x)}$.

Definition 1 We say $\hat{f}$ is *well-calibrated* for a distribution D and bins $B_1, \ldots, B_K$ if

$$\hat{p}(x) = c_{\hat{f}}(x) \coloneqq \mathbb{P}_{(x',y')\sim D}\left[y' = \hat{y}(x') \mid \hat{p}(x') \in B_{\kappa_{\hat{f}}(x)}\right] \qquad (\forall x \in \mathcal{X}), \tag{2}$$

where we refer to $c_{\hat{f}}(x)$ as the *true confidence*. Intuitively, this definition "coarsens" the calibration problem across the bins—*i.e.,* rather than sorting the inputs x into a continuum of "bins" $\hat{p}(x) = t$ for each $t \in [0, 1]$ as in (1), we sort them into a finite number of bins $\hat{p}(x) \in B_k$; intuitively, we have $c^*_{\hat{f}} \approx c_{\hat{f}}$ if the bin sizes are close to zero. It may not be obvious what downstream guarantees can be obtained based on this definition; we provide examples in Sections 3 & 4.

Problem formulation. We formalize the problem of PAC calibration. We focus on the setting where the training and test distributions are identical—*e.g.,* we cannot handle adversarial examples or changes in covariate distribution in the reinforcement learning setting. Importantly, while we assume a pre-trained confidence predictor $\hat{f}$ is given, we make no assumptions about $\hat{f}$—*e.g.,* it can be heuristically calibrated, uncalibrated, or even poorly calibrated. Intuitively, if $\hat{f}$ performs poorly, then the predicted confidences will be close to $1/2$—*i.e.,* express no confidence in the predictions. Note that it is fine for the predictions of $\hat{f}$ to be poorly calibrated; the important property is confidences predicted by $\hat{f}$ have similar true confidences.

The challenge in formalizing PAC calibration is that quantifying over all x in (2). One approach is to provide guarantees in expectation over x (Kumar et al., 2019); however, this approach does not provide guarantees for individual predictions.

Instead, our goal is to find a *set* of predicted confidences that includes the true confidence with high probability. Of course, we could simply predict the interval $[0, 1]$, which always contains the true confidence; thus, simultaneously want to make the size of the interval small. To this end, we consider a *confidence coverage predictor* $\hat{C} : \mathcal{X} \to 2^{\mathbb{R}}$, where $c_{\hat{f}}(x) \in \hat{C}(x)$ with high probability. In particular, $\hat{C}(x)$ outputs an interval $[\underline{c}, \overline{c}] \subseteq \mathbb{R}$, where $\underline{c} \leq \overline{c}$, instead of a set. We only consider a single interval (rather than disjoint intervals) since one suffices to localize the true confidence $c_{\hat{f}}$.

We are interested in providing theoretical guarantees for an algorithm used to construct confidence coverage predictor $\hat{C}$ given a held-out calibration set $Z \subseteq \mathcal{X} \times \mathcal{Y}$. In addition, we assume the algorithm is given a pretrained confidence predictor $\hat{f}$. Thus, we consider $\hat{C}$ as depending on Z and $\hat{f}$, which we denote by $\hat{C}(\cdot; \hat{f}, Z)$. Then, we want $\hat{C}$ to satisfy the following guarantee:

Definition 2 Given $\delta \in \mathbb{R}_{>0}$ and $n \in \mathbb{N}$, $\hat{C}$ is *probably approximately correct (PAC)* if for all D,

$$\mathbb{P}_{Z\sim D^n}\left[\bigwedge_{x\in\mathcal{X}} c_{\hat{f}}(x) \in \hat{C}(x; \hat{f}, Z)\right] \geq 1 - \delta. \tag{3}$$

We provide a formal connection to PAC learning theory in Appendix A.

Algorithm. We propose a confidence coverage predictor that satisfies the PAC property. The problem of estimating the confidence interval $\hat{C}(x)$ of the binned true confidence $c_{\hat{f}}(x)$ is closely related to the binomial proportion confidence interval estimation; consider a Bernoulli random variable $b \sim B \coloneqq \text{Bernoulli}(\theta)$, where $b = 1$ denotes a success and $b = 0$ denotes a failure, and θ is unknown. Given a sequence of observations $b_{1:n} \coloneqq (b_1, \ldots, b_n) \sim B^n$, the goal is to construct an interval $\hat{\Theta}(b_{1:n}) \subseteq \mathbb{R}$ that includes θ with high probability—*i.e.,*

$$\mathbb{P}_{b_{1:n} \sim B^n} \left[\theta \in \hat{\Theta}(b_{1:n}) \right] \geq 1 - \alpha, \tag{4}$$

where $\alpha \in \mathbb{R}_{>0}$ is a given confidence level. In particular, the Clopper-Pearson interval

$$\hat{\Theta}_{\text{CP}}(b_{1:n}; \alpha) \coloneqq \left[\inf_{\theta} \left\{ \theta \;\middle|\; \mathbb{P}_\theta \left[S \geq s \right] \geq \frac{\alpha}{2} \right\}, \; \sup_{\theta} \left\{ \theta \;\middle|\; \mathbb{P}_\theta \left[S \leq s \right] \geq \frac{\alpha}{2} \right\} \right],$$

guarantees (4) (Clopper & Pearson, 1934; Brown et al., 2001), where $s = \sum_{i=1}^{n} b_i$ is the number of observed successes, n is the number of observations, and S is a Binomial random variable $S \sim \text{Binomial}(n, \theta)$. Intuitively, the interval is constructed such that the number of observed success falls in the region with high-probability for any θ in the interval. The following expression is equivalent due to the relationship between the Binomial and Beta distributions (Hartley & Fitch, 1951; Brown et al., 2001)—*i.e.,* $\mathbb{P}_\theta[S \geq s] = I_\theta(s, n - s + 1)$, where I_θ is the CDF of $\text{Beta}(s, n - s + 1)$:

$$\hat{\Theta}_{\text{CP}}(b_{1:n}; \alpha) = \left[\frac{\alpha}{2} \text{ quantile of Beta}(s, n - s + 1), \; \left(1 - \frac{\alpha}{2}\right) \text{ quantile of Beta}(s + 1, n - s) \right].$$

Now, for each of the K bins, we apply $\hat{\Theta}_{\text{CP}}$ with confidence level $\alpha = \frac{\delta}{K}$—*i.e.,*

$$\hat{C}(x; \hat{f}, Z, \delta) \coloneqq \hat{\Theta}_{\text{CP}} \left(W_{\kappa_{\hat{f}}(x)}; \frac{\delta}{K} \right) \text{ where } W_k \coloneqq \left\{ \mathbb{1}(\hat{y}(x) = y) \;\middle|\; (x, y) \in Z \text{ s.t. } \kappa_{\hat{f}}(x) = k \right\}.$$

Here, W_k is the set of observations of successes vs. failures corresponding to the subset of labeled examples $(x, y) \in Z$ such that $\hat{p}(x)$ falls in the bin B_k, where a success is defined to be a correct prediction $\hat{y}(x) = y$. We note that for efficiency, the confidence interval for each of the K bins can be precomputed. Our construction of $\hat{C}$ satisfies the following; see Appendix B for a proof:

Theorem 1 *Our confidence coverage predictor $\hat{C}$ satisfies the PAC property.*

Finally, an important special case is when there is a single bin $B = [0, 1]$—*i.e.,*

$$\hat{C}_0(x; \hat{f}, Z', \delta) \coloneqq \hat{\Theta}_{\text{CP}}(W; \delta) \qquad \text{where} \qquad W \coloneqq \{\mathbb{1}(\hat{y}(x') = y') \mid (x', y') \in Z'\}.$$

Note that $\hat{C}_0$ does not depend on x, so we drop it—*i.e.,* $\hat{C}_0(\hat{f}, Z', \delta) \coloneqq \hat{\Theta}_{\text{CP}}(W; \delta)$—*i.e.,* $\hat{C}_0$ computes the Clopper-Pearson interval over Z', which is a subset of the original set Z.

3 Application to Fast DNN Inference

A key application of predicted confidences is to perform *model composition* to improve the running time of DNNs without sacrificing accuracy. The idea is to use a fast but inaccurate model when it is confident in its prediction, and switch to an accurate but slow model otherwise (Bolukbasi et al., 2017); we refer to the combination as the *composed model*. To further improve performance, we can have the two models share a backbone—*i.e.,* the fast model shares the first few layers of the slow model (Teerapittayanon et al., 2016). We refer to the decision of whether to skip the slow model as the *exit condition*; then, our goal is to construct confidence thresholds for exit conditions in a way that provides theoretical guarantees on the overall accuracy.

Problem setup. The early-stopping approach for fast DNN inference can be formalized as a sequence of branching classifiers organized in a cascading way—*i.e.,*

$$\hat{y}_C(x; \gamma_{1:M-1}) \coloneqq \begin{cases} \hat{y}_m(x) & \text{if } \bigwedge_{i=1}^{m-1} (\hat{p}_i(x) < \gamma_i) \wedge (\hat{p}_m(x) \geq \gamma_m) \;\; (\forall m \in \{1, ..., M-1\}) \\ \hat{y}_M(x) & \text{otherwise,} \end{cases}$$

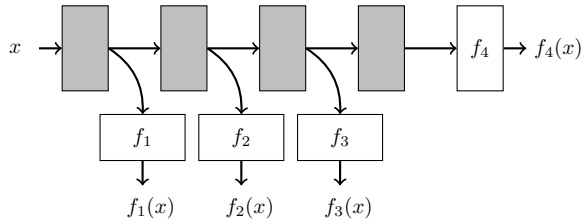


Figure 1: A composed model in a cascading way for $M = 4$.

where M is the number of branches, $\hat{f}_m$ is the confidence predictor, and $\hat{y}_m$ and $\hat{p}_m$ are the associated label and top-label confidence predictor, respectively. For conciseness, we denote the exit condition of the mth branch by d_m (*i.e.,* $d_m(x) \coloneqq \mathbb{1}(\bigwedge_{i=1}^{m-1} (\hat{p}_i(x) < \gamma_i) \wedge (\hat{p}_m(x) \geq \gamma_m)))$ with thresholds $\gamma_1, \ldots, \gamma_m \in [0, 1]$. The $\hat{f}_m$ share a backbone and are trained in the standard way; see Appendix F.4 for details. Figure 1 illustrates the composed model for $M = 4$; the gray area represents the shared backbone. We refer to an overall model composed in this way as a *cascading classifier*.

Desired error bound. Given $\xi \in \mathbb{R}_{>0}$, our goal is to choose $\gamma_{1:M-1} \coloneqq (\gamma_1, \ldots, \gamma_{M-1})$ so the error difference of the cascading classifier $\hat{y}_C$ and the slow classifier $\hat{y}_M$ is at most ξ—*i.e.,*

$$p_{\text{err}} \coloneqq \mathbb{P}_{(x,y)\sim D}\left[\hat{y}_C(x) \neq y\right] - \mathbb{P}_{(x,y)\sim D}\left[\hat{y}_M(x) \neq y\right] \leq \xi. \tag{5}$$

To obtain the desired error bound, an example x exits at the mth branch if $\hat{y}_m$ is likely to classify x correctly, allowing for at most ξ fraction of errors total. Intuitively, if the confidence of $\hat{y}_m$ on x is sufficiently high, then $\hat{y}_m(x) = y$ with high probability. In this case, $\hat{y}_M$ either correctly classifies or misclassifies the same example; if the example is misclassified, it contributes to decrease p_{err}; otherwise, we have $\hat{y}_m(x) = y = \hat{y}_M(x)$ with high probability, which contributes to maintain p_{err}.

Fast inference. To minimize running time, we prefer to allow higher error rates at the lower branches—*i.e.,* we want to choose γ_m as small as possible at lower branches m.

Algorithm. Our algorithm takes prediction branches $\hat{f}_m$ (for $m \in \{1, \ldots, M\}$), the desired relative error $\xi \in [0, 1]$, a confidence level $\delta \in [0, 1]$, and a calibration set $Z \subseteq \mathcal{X} \times \mathcal{Y}$, and outputs $\gamma_{1:M-1}$ so that (5) holds with probability at least $1 - \delta$. It iteratively chooses the thresholds from γ_1 to γ_{M-1}; at each step, it chooses γ_m as small as possible subject to $p_{\text{err}} \leq \xi$. Note that γ_m implicitly appears in p_{err} in the constraint due to the dependence of $d_m(x)$ on γ_m. The challenge is enforcing the constraint since we cannot evaluate it. To this end, let

$$\begin{aligned} e_m &\coloneqq \mathbb{P}_{(x,y)\sim D}\left[\hat{y}_m(x) \neq y \wedge \hat{y}_m(x) \neq \hat{y}_M(x) \wedge d_m(x) = 1\right] \\ e'_m &\coloneqq \mathbb{P}_{(x,y)\sim D}\left[\hat{y}_M(x) \neq y \wedge \hat{y}_m(x) \neq \hat{y}_M(x) \wedge d_m(x) = 1\right], \end{aligned}$$

then it is possible to show that $p_{\text{err}} = \sum_{m=1}^{M-1} e_m - e'_m$ (see proof of Theorem 2 in Appendix C). Then, we can compute bounds on e_m and e'_m using the following:

$$\begin{aligned} \mathbb{P}\left[\hat{y}_m(x) = y \mid \hat{y}_m(x) \neq \hat{y}_M(x) \wedge d_m(x) = 1\right] &\in [\underline{c}_m, \bar{c}_m] \coloneqq \hat{C}_0\left(\hat{f}_m, Z_m, \frac{\delta}{3(M-1)}\right) \\ \mathbb{P}\left[\hat{y}_M(x) = y \mid \hat{y}_m(x) \neq \hat{y}_M(x) \wedge d_m(x) = 1\right] &\in [\underline{c}'_m, \bar{c}'_m] \coloneqq \hat{C}_0\left(\hat{f}_M, Z_m, \frac{\delta}{3(M-1)}\right) \\ \mathbb{P}\left[\hat{y}_m(x) \neq \hat{y}_M(x) \wedge d_m(x) = 1\right] &\in [\underline{r}_m, \bar{r}_m] \coloneqq \hat{\Theta}_{\text{CP}}\left(W_m; \frac{\delta}{3(M-1)}\right), \end{aligned}$$

where

$$\begin{aligned} Z_m &\coloneqq \{(x, y) \in Z \mid \hat{y}_m(x) \neq \hat{y}_M(x) \wedge d_m(x) = 1\} \\ W_m &\coloneqq \{\mathbb{1}(\hat{y}_m(x) \neq \hat{y}_M(x) \wedge d_m(x) = 1) \mid (x, y) \in Z\}. \end{aligned}$$

Thus, we have $e_m \leq \bar{c}_m \bar{r}_m$ and $e'_m \geq \underline{c}'_m \underline{r}_m$, in which case it suffices to solve

$$\gamma_m = \operatorname*{arg\,min}_{\gamma \in [0,1]} \gamma \qquad \text{subj. to} \qquad \sum_{i=1}^{m} \bar{c}_i \bar{r}_i - \underline{c}'_i \underline{r}_i \leq \xi. \tag{6}$$

Here, $\bar{c}_m$, $\bar{r}_m$, $\underline{c}_m$, and $\underline{r}_m$ are implicitly a function of γ, which we can optimize using line search. We have the following guarantee; see Appendix C for a proof:

Theorem 2 *We have $p_{err} \leq \xi$ with probability at least $1 - \delta$ over $Z \sim D^n$.*

Moreover, the proposed greedy algorithm (6) is actually optimal in reducing inference speed when $M = 2$. Intuitively, we are always better off in terms of inference time by classifying more examples using a faster model. In particular, we have the following theorem; see Appendix D for a proof:

Theorem 3 *If $M = 2$, γ_1^* minimizes (6), and the classifiers $\hat{y}_m$ are faster for smaller m, then the resulting $\hat{y}_C$ is the fastest cascading classifier among cascading classifiers that satisfy $p_{err} \leq \xi$.*

4 Application to Safe Planning

Robots acting in open world environments must rely on deep learning for tasks such as object recognition—*e.g.,* detect objects in a camera image; providing guarantees on these predictions is critical for safe planning. Safety requires not just that the robot is safe while taking the action, but that it can safely come to a stop afterwards—*e.g.,* that a robot can safely come to a stop before running into a wall. We consider a binary classification DNN trained to predict a probability $\hat{f}(x) \in [0, 1]$ of whether the robot is unsafe in this sense.[1] If $\hat{f}(x) \geq \gamma$ for some threshold $\gamma \in [0, 1]$, then the robot comes to a stop (*e.g.,* to ask a human for help). If the label $\mathbb{1}(\hat{f}(x) \geq \gamma)$ correctly predicts safety, then this policy ensures safety as long as the robot starts from a safe state (Li & Bastani, 2020). We apply our approach to choose γ to ensure safety with high probability.

Problem setup. Given a performant but potentially unsafe policy $\hat{\pi}$ (*e.g.,* a DNN policy trained to navigate to the goal), our goal is to override $\hat{\pi}$ as needed to ensure safety. We assume that $\hat{\pi}$ is trained in a simulator, and our goal is to ensure that $\hat{\pi}$ is safe according to our model of the environment, which is already a challenging problem when $\hat{\pi}$ is a deep neural network over perceptual inputs. In particular, we do not address the sim-to-real problem.

Let $x \in \mathcal{X}$ be the states, $\mathcal{X}_{\text{safe}} \subseteq \mathcal{X}$ be the safe states (*i.e.,* our goal is to ensure the robot stays in $\mathcal{X}_{\text{safe}}$), $o \in \mathcal{O}$ be the observations, $u \in \mathcal{U}$ be the actions, $g : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ be the (deterministic) dynamics, and $h : \mathcal{X} \to \mathcal{O}$ be the observation function. A state x is *recoverable* (denoted $x \in \mathcal{X}_{\text{rec}}$) if the robot can use $\hat{\pi}$ in state x and then safely come to a stop using a *backup policy* π_0 (*e.g.,* braking).

Then, the *shield policy* uses $\hat{\pi}$ if $x \in \mathcal{X}_{\text{rec}}$, and π_0 otherwise (Bastani, 2019). This policy guarantees safety as long as an initial state is recoverable—*i.e.,* $x_0 \in \mathcal{X}_{\text{rec}}$. The challenge is determining whether $x \in \mathcal{X}_{\text{rec}}$. When we observe x, we can use model-based simulation to perform this check. However, in many settings, we only have access to observations—*e.g.,* camera images or LIDAR scans—so this approach does not apply. Instead, we propose to train a DNN to predict recoverability—*i.e.,*

$$\hat{y}(o) \coloneqq \begin{cases} 1 \text{ (“un-recoverable”)} & \text{if } \hat{f}(o) \geq \gamma \\ 0 \text{ (“recoverable”)} & \text{otherwise} \end{cases} \qquad \text{where} \qquad o = h(x),$$

with the goal that $\hat{y}(o) \approx y^*(x) \coloneqq \mathbb{1}(x \notin \mathcal{X}_{\text{rec}})$, resulting in the following the shield policy π_{shield}:

$$\pi_{\text{shield}}(o) \coloneqq \begin{cases} \hat{\pi}(o) & \text{if } \hat{y}(o) = 0 \\ \pi_0(o) & \text{otherwise.} \end{cases}$$

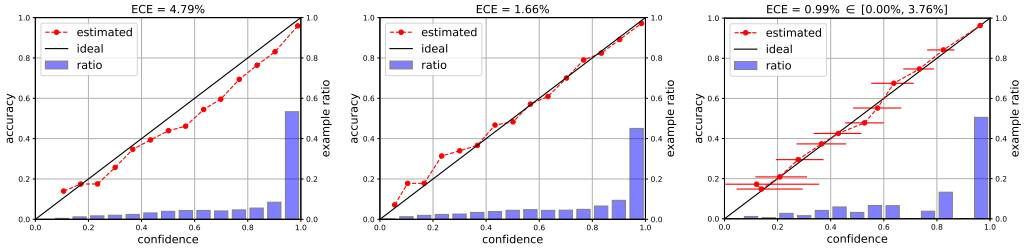
Safety guarantee. Our main goal is to choose γ so that π_{shield} ensures safety with high probability—*i.e.,* given $\xi \in \mathbb{R}_{>0}$ and any distribution D over initial states $\mathcal{X}_0 \subseteq \mathcal{X}_{\text{rec}}$, we have

$$p_{\text{unsafe}} \coloneqq \mathbb{P}_{\zeta \sim D_{\pi_{\text{shield}}}}[\zeta \not\subseteq \mathcal{X}_{\text{safe}}] \leq \xi, \tag{7}$$

where $\zeta(x_0, \pi) \coloneqq (x_0, x_1, \dots)$ is a rollout from $x_0 \sim D$ generated using π—*i.e.,* $x_{t+1} = g(x_t, \pi(h(x_t)))$.[2] We assume the rollout terminates either once the robot reaches its goal, or once it switches to π_0 and comes to a stop; in particular, the robot never switches from π_0 back to $\hat{\pi}$.

[1] Since $|\mathcal{Y}| = 2$, $\hat{f}$ can be represented as a map $\hat{f} : \mathcal{X} \to [0, 1]$; the second component is simply $1 - \hat{f}(x)$.

[2] We can handle infinitely long rollouts, but in practice rollouts will be finite (but possibly arbitrarily long). and $D_{\pi_{\text{shield}}}$ is an induced distribution over rollouts $\zeta(x_0, \pi_{\text{shield}})$.



(a) Naïve softmax (b) Temperature scaling (c) Proposed

Figure 2: Calibration comparison; default parameters of $\hat{C}$ are $K = 20$, $n = 20,000$, and $\delta = 10^{-2}$.

Success rate. To maximize the success rate (*i.e.,* the rate at which the robot achieves its goal), we need to minimize how often π_{shield} switches to π_0, which corresponds to maximizing γ.

Algorithm. Our algorithm takes the confidence predictor $\hat{f}$, desired bound $\xi \in \mathbb{R}_{>0}$ on the unsafety probability, confidence level $\delta \in [0, 1]$, calibration set $W \subseteq \mathcal{X}^\infty$ of rollouts $\zeta \sim D_{\hat{\pi}}$, and calibration set $Z \subseteq \mathcal{O}$ of samples from distribution $\tilde{D}$ described below; see Appendix F.5 for details on sampling ζ and constructing W, Z, and $\tilde{D}$. We want to maximize γ subject to $p_{\text{unsafe}} \leq \xi$, where p_{unsafe} is implicitly a function of γ. However, we cannot evaluate p_{unsafe}, so we need an upper bound.

To this end, consider a rollout that the first unsafe state is encountered on step t (*i.e.,* $x_t \notin \mathcal{X}_{\text{safe}}$ but $x_i \in \mathcal{X}_{\text{safe}}$ for all $i < t$), which we call an *unsafe rollout*, and denote the event that the unsafe rollout is encountered by E_t; we exploit the unsafe rollouts to bound p_{unsafe}. In particular, let $p_t := \mathbb{P}_{\zeta \sim D_{\hat{\pi}}}[E_t]$, and let $\bar{p} := \sum_{t=0}^{\infty} p_t$ be the probability that a rollout is unsafe. Then, consider a new distribution $\tilde{D}$ over $\mathcal{O}$ with a probability density function $p_{\tilde{D}}(o) := \sum_{t=0}^{\infty} p_{D_{\hat{\pi}}}(o \mid E_t) \cdot p_t / \bar{p}$, where $p_{D_{\hat{\pi}}}$ is the original probability density function of $D_{\hat{\pi}}$; in particular, we can draw an observation $o \sim \tilde{D}$ by sampling the observation of the first unsafe state from a rollout sample (and rejecting if the entire rollout is safe). Then, we can show the following (see a proof of Theorem 4 in Appendix E):

$$p_{\text{unsafe}} \leq \mathbb{P}_{o \sim \tilde{D}}[\hat{y}(o) = 0] \cdot \bar{p} =: \bar{p}_{\text{unsafe}}. \tag{8}$$

We use our confidence coverage predictor $\hat{C}_0$ to compute bounds on $\bar{p}_{\text{unsafe}}$—*i.e.,*

$$\mathbb{P}_{o \sim \tilde{D}}[\hat{y}(o) = 1] \in [\underline{c}, \bar{c}] := \hat{C}_0\left(\hat{f}, Z', \frac{\delta}{2}\right) \qquad \text{where} \qquad Z' := \{(o, 1) \mid o \in Z\}$$

$$\bar{p} \in [\underline{r}, \bar{r}] := \hat{\Theta}_{\text{CP}}\left(W', \frac{\delta}{2}\right) \qquad \text{where} \qquad W' := \{\mathbb{1}(\zeta \not\subseteq \mathcal{X}_{\text{safe}}) \mid \zeta \in W\}.$$

Here, Z' is a labeled version of Z, where "1" denotes "un-recoverable", $n := |W|$, and $n' := |Z|$, where $n \geq n'$. Then, we have $\bar{p}_{\text{unsafe}} \leq \bar{r} \cdot (1 - \underline{c})$, so it suffices to solve the following problem:

$$\gamma := \underset{\gamma' \in [0,1]}{\arg\max}\ \gamma' \qquad \text{subj. to} \qquad \bar{r} \cdot (1 - \underline{c}) \leq \xi \tag{9}$$

Here, $\underline{c}$ is implicitly a function of γ'; thus, we use line search to solve this optimization problem. We have the following safety guarantee, see Appendix E for a proof:

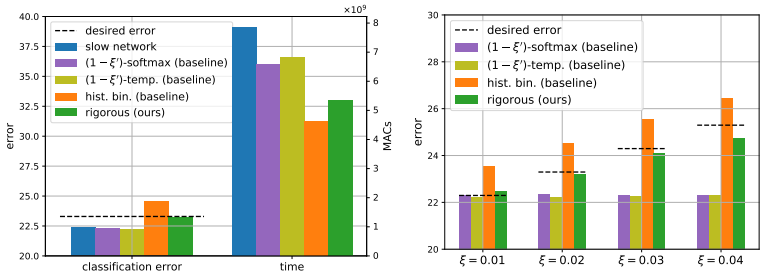
Theorem 4 *We have $p_{unsafe} \leq \xi$ with probability at least $1 - \delta$ over $W \sim D_{\hat{\pi}}^n$ and $Z \sim \tilde{D}^{n'}$.*

5 Experiments

We demonstrate that how our proposed approach can be used to obtain provable guarantees in our two applications: fast DNN inference and safe planning. Additional results are in Appendix G.

5.1 Calibration

We illustrate the calibration properties of our approach using reliability diagrams, which show the empirical accuracy of each bin as a function of the predicted confidence (Guo et al., 2017). Ideally,



(a) Comparison to baselines. (b) Varying desired error rate ξ.

method	top-1 error (%)	CPU (µs)	GPU (µs)	MACs
slow network	22.32	214.31	2.95	7.83×10^9
$(1-\xi')$-softmax (baseline)	22.34	104.24	1.78	6.60×10^9
$(1-\xi')$-temperature scaling (baseline)	22.22	109.57	1.81	6.83×10^9
histogram binning (baseline)	24.54	70.72	1.19	4.62×10^9
rigorous (ours)	23.26	85.58	1.34	5.33×10^9

(c) Running time comparison.

Figure 3: Fast DNN inference results; default parameters are $n = 20,000$, $\xi = 0.02$, and $\delta = 10^{-2}$.

the accuracy should equal the predicted confidence, so the ideal curve is the line $y = x$. To draw our predicted confidence intervals in these plots, we need to rescale them; see Appendix F.3.

Setup. We use the ImageNet dataset (Russakovsky et al., 2015) and ResNet101 (He et al., 2016) for evaluation. We split the ImageNet validation set into $20,000$ calibration and $10,000$ test images.

Baselines. We consider three baselines: (i) naïve softmax of $\hat{f}$, (ii) temperature scaling (Guo et al., 2017), and (iii) histogram binning (Zadrozny & Elkan, 2001); see Appendix F.2 for details. For histogram binning and our approach, we use $K = 20$ bins of the same size.

Metric. We use expected calibration error (ECE) and reliability diagrams (see Appendix F.3).

Results. Results are shown in Figure 2. The ECE of the naïve softmax baseline is 4.79% (Figure 2a), of temperature scaling enhances this to 1.66% (Figure 2b), and of histogram binning is 0.99% (Figure 2c). Our approach predicts an interval that include the empirical accuracy in all bins (solid red lines in Figure 2c); furthermore, the upper/lower bounds of the ECE over values in our bins is $[0.0\%, 3.76\%]$, which includes zero ECE. See Appendix G.1 for additional results.

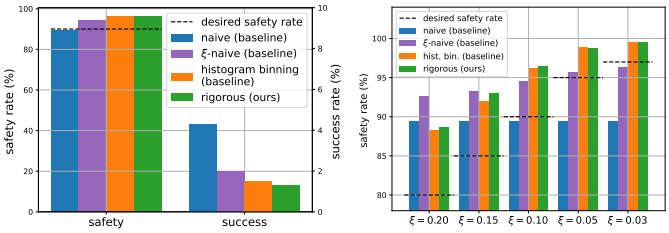
5.2 Fast DNN Inference

Setup. We use the same ImageNet setup along with ResNet101 as the calibration task. For the cascading classifier, we use the original ResNet101 as the slow network, and add a single exit branch (*i.e.,* $M = 2$) at a quarter of the way from the input layer. We train the newly added branch using the standard procedure for training ResNet101.

Baselines. We compare to naïve softmax and to calibrated prediction via temperature scaling, both using a threshold $\gamma_1 = 1 - \xi'$, where ξ' is the sum of ξ and the validation error of the slow model; intuitively, this threshold is the one we would use if the predicted probabilities are perfectly calibrated. We also compare to histogram binning—*i.e.,* our approach but using the means of each bin instead of the upper/lower bounds. See Appendix F.2 for details.

Metrics. First, we measure test set top-1 classification error (*i.e.,* $1 -$ accuracy), which we want to guarantee this lower than a desired error (*i.e.,* the error of the slow model and desired relative error ξ). To measure inference time, we consider the average number of multiplication-accumulation operations (MACs) used in inference per example. Note that the MACs are averaged over all examples in the test set since the combined model may use different numbers of MACs for different examples.

Results. The comparison results with the baselines are shown in Figure 3a. The original neural network model is denoted by "slow network", our approach (6) by "rigorous", and our baseline by "$(1-\xi')$-softmax", "$(1-\xi')$-temp.", and "hist. bin.". For each method, we plot the classification



(a) Comparison to baselines. (b) Varying desired safety rate ξ.

Figure 4: Safe planning results; default parameters are: $n = 20,000$, $\xi = 0.1$, $\delta = 10^{-2}$.

error and time in MACs. The desired error upper bound is plotted as a dotted line; the goal is for the classification error to be lower than this line. As can be seen, our method is guaranteed to achieve the desired error, while improving the inference time by 32% compared to the slow model. On the other hand, the histogram baseline improves the inference time but fails to satisfy the desired error. Asymptotically, histogram binning is guaranteed to be perfectly calibrated, but it makes mistakes due to finite sample errors. The other baselines do not improve inference time. Next, Figure 3b shows the classification error as we vary the desired relative error ξ; our approach always achieves the desired error on the test set, and is often very close (which maximizes speed). However, the baselines often violate the desired error bound. Finally, the MACs metric is only an approximation of the actual inference time. To complement MACs, we also measure CPU and GPU time using the PyTorch profiler. In Figure 3c, we show the inference times for each method, where trends are as before; our approach improves running time by 54%, while only reducing classification error by 1%. The histogram baseline is faster than our approach, but does not satisfy the error guarantee. These results include the time needed to compute the intervals, which is negligible.

5.3 Safe Planning

Setup. We evaluate on AI Habitat (Savva et al., 2019), an indoor robot simulator that provides agents with observations $o = h(x)$ that are RGB camera images. The safety constraint is to avoid colliding with obstacles such as furniture in the environment. We use the learned policy $\hat{\pi}$ available with the environment. Then, we train a recoverability predictor, trained using 500 rollouts with a horizon of 100. We calibrate this model on an additional n rollouts.

Baselines. We compare to three baselines: (i) histogram binning—*i.e.,* our approach but using the means of each bin rather than upper/lower bounds, (ii) a naïve approach of choosing $\gamma = 0.5$, and (iii) a naïve but adaptive approach of choosing $\gamma = \xi$, called "ξ-naïve".

Metrics. We measure both the safety rate and the success rate; in particular, a rollout is successful if the robot reaches its goal state, and a rollout is safe if it does not reach any unsafe states.

Results. We show results in Figure 4a. The desire safety rate ξ is shown by the dotted line—*i.e.,* we expect the safety rate to be above this line. As can be seen, our approach achieves the desired safety rate. While it sacrifices success rate, this is because the underlying learned policy $\hat{\pi}$ is frequently unsafe; in particular, it is only safe in about 30% of rollouts. The naïve approach fails to satisfy the safety constraint. The ξ-naïve approach tends to be optimistic, and also fails when $\xi = 0.03$ (Figure 4b). The histogram baseline performs similarly to our approach. The main benefit of our approach is providing the absolute guarantee on safety, which the histogram baseline does not provide. Thus, in this case, our approach can provide this guarantee while achieving similar performance. Figure 4b shows the safety rate as we vary the desired safety rate ξ. Both our approach and the baseline satisfy the desired safety guarantee, whereas the naive approaches do not always do so.

6 Conclusion

We have proposed a novel algorithm for calibrated prediction that provides PAC guarantees, and demonstrated how our approach can be applied to fast DNN inference and safe planning. There are many directions for future work—*e.g.,* leveraging these techniques in more application domains and extending our approach to settings with distribution shift (see Appendix F.1 for a discussion).

REFERENCES

A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin. Reachability-based safe learning with gaussian processes. In *53rd IEEE Conference on Decision and Control*, pp. 1424–1431, 2014.

Mohammed Alshiekh, R. Bloem, R. Ehlers, Bettina Könighofer, S. Niekum, and U. Topcu. Safe reinforcement learning via shielding. *ArXiv*, abs/1708.08611, 2018.

Osbert Bastani. Safe planning via model predictive shielding, 2019.

Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 527–536. JMLR. org, 2017.

Lawrence D Brown, T Tony Cai, and Anirban DasGupta. Interval estimation for a binomial proportion. *Statistical science*, pp. 101–117, 2001.

Maxime Cauchois, Suyash Gupta, Alnur Ali, and John C. Duchi. Robust validation: Confident predictions even when distributions shift, 2020.

Charles J Clopper and Egon S Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Sarah Dean, Nikolai Matni, Benjamin Recht, and Vickie Ye. Robust guarantees for perception-based control, 2019.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

Michael Figurnov, Maxwell D Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov. Spatially adaptive computation time for residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1039–1048, 2017.

J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2019.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410, 2016.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.

HO Hartley and ER Fitch. A chart for the incomplete beta-function and the cumulative binomial distribution. *Biometrika*, 38(3/4):423–426, 1951.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *European Conference on Computer Vision*, pp. 201–214. Springer, 2012.

Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems*, pp. 3792–3803, 2019.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pp. 6402–6413, 2017.

S. Li and O. Bastani. Robust model predictive shielding for safe reinforcement learning with stochastic dynamics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7166–7172, 2020.

Shuo Li and Osbert Bastani. Robust model predictive shielding for safe reinforcement learning with stochastic dynamics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7166–7172. IEEE, 2020.

Peng Liao, Kristjan Greenewald, Predrag Klasnja, and Susan Murphy. Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–22, 2020.

Allan H Murphy. Scalar and vector partitions of the probability score: Part i. two-state situation. *Journal of Applied Meteorology*, 11(2):273–282, 1972.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Sangdon Park, Osbert Bastani, Nikolai Matni, and Insup Lee. Pac confidence sets for deep neural networks via calibrated prediction. In *International Conference on Learning Representations*, 2020a. URL `https://openreview.net/forum?id=BJxVI04YvB`.

Sangdon Park, Osbert Bastani, James Weimer, and Insup Lee. Calibrated prediction with covariate shift via unsupervised domain adaptation. In *The 23rd International Conference on Artificial Intelligence and Statistics*, 2020b.

John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 1999.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Itay Safran and Ohad Shamir. Spurious local minima are common in two-layer relu neural networks. In *International Conference on Machine Learning*, pp. 4433–4441. PMLR, 2018.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 2464–2469. IEEE, 2016.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in Neural Information Processing Systems*, 32:2530–2540, 2019.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Kim Wabersich and Melanie Zeilinger. Linear model predictive safety certification for learning-based control. 03 2018. doi: 10.1109/CDC.2018.8619829.

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 409–424, 2018.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *In Proceedings of the Eighteenth International Conference on Machine Learning*. Citeseer, 2001.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699. ACM, 2002.

A CONNECTION TO PAC LEARNING THEORY

We explain the connection to PAC learning theory. First, note that we can represent $\hat{C}$ as a confidence interval around the empirical estimate of $c_{\hat{f}}(x)$—*i.e.,* $\hat{c}_{\hat{f}}(x) := \sum_{(x',y') \in S_x} \mathbb{1}(y' = \hat{y}(x'))/|S_x|$, where $S_x = \{(x', y') \mid \hat{p}(x') \in B_{\kappa_{\hat{f}}(x)}\}$. Then, we can write $\hat{C}(x) = [\hat{c}_{\hat{f}}(x) - \underline{\epsilon}_x, \hat{c}_{\hat{f}}(x) + \bar{\epsilon}_x]$. In this case, (3) is equivalent to

$$\mathbb{P}_{Z \sim D^n} \left[\bigwedge_{x \in \mathcal{X}} \hat{c}_{\hat{f}}(x) - \underline{\epsilon}_{\kappa_{\hat{f}}(x)} \leq c_{\hat{f}}(x) \leq \hat{c}_{\hat{f}}(x) + \bar{\epsilon}_{\kappa_{\hat{f}}(x)} \right] \geq 1 - \delta, \tag{10}$$

for some $\underline{\epsilon}_1, \bar{\epsilon}_1, \ldots, \underline{\epsilon}_K, \bar{\epsilon}_K$. This bound is similar to concentration inequalities such as Hoeffding's inequality, where "approximately" refers to the bound on the error of our empirical estimate $\hat{c}_{\hat{f}}(x)$ compared to the true value $c_{\hat{f}}(x)$, and "probably" refers to the fact that this error bound holds with high probability over the training data $Z \sim D^n$.

Finally, the following provides a more formal connection to PAC learning theory:

Definition 3 Given $\epsilon, \delta \in \mathbb{R}_{>0}$ and $n \in \mathbb{N}$, $\hat{C}$ is *probably approximately correct (PAC)* if, for any distribution D, we have

$$\mathbb{P}_{Z \sim D^n} \left[\mathbb{P}_{x \sim D} \left[c_{\hat{f}}(x) \in \hat{C}(x; \hat{f}, Z) \right] \geq 1 - \epsilon \right] \geq 1 - \delta. \tag{11}$$

The following theorem shows that the proposed confidence coverage predictor $\hat{C}$ satisfies the PAC guarantee in Definition 3.

Theorem 5 *Our confidence coverage predictor $\hat{C}$ satisfies Definition 3 for all $\epsilon > 0$.*

Proof. We exploit the independence of each bin for the proof. Let $\theta_{\kappa_{\hat{f}}(x)} := c_{\hat{f}}(x)$, which is the parameter of the Binomial distribution of the $\kappa_{\hat{f}}(x)$th bin, the following holds:

$$
\begin{aligned}
&\mathop{\mathbb{P}}_{Z\sim D^n}\left[\mathop{\mathbb{P}}_{x\sim D}\left[c_{\hat{f}}(x) \in \hat{C}(x;\hat{f},Z,\delta)\right] \geq 1-\epsilon\right] \\
&= \mathop{\mathbb{P}}_{Z\sim D^n}\left[\mathop{\mathbb{P}}_{x\sim D}\left[c_{\hat{f}}(x) \in \hat{C}(x;\hat{f},Z,\delta) \wedge \bigvee_{k=1}^{K} \hat{p}(x) \in B_k\right] \geq 1-\epsilon\right] \\
&= \mathop{\mathbb{P}}_{Z\sim D^n}\left[\sum_{k=1}^{K} \mathop{\mathbb{P}}_{x\sim D}\left[c_{\hat{f}}(x) \in \hat{C}(x;\hat{f},Z,\delta) \wedge \hat{p}(x) \in B_k\right] \geq 1-\epsilon\right] \\
&= \mathop{\mathbb{P}}_{Z\sim D^n}\left[\sum_{k=1}^{K} \mathop{\mathbb{P}}_{x\sim D}\left[c_{\hat{f}}(x) \in \hat{C}(x;\hat{f},Z,\delta) \;\middle|\; \hat{p}(x) \in B_k\right] \mathop{\mathbb{P}}_{x\sim D}\left[\hat{p}(x) \in B_k\right] \geq 1-\epsilon\right] \\
&= \mathop{\mathbb{P}}_{Z\sim D^n}\left[\sum_{k=1}^{K} \mathop{\mathbb{P}}_{x\sim D}\left[\theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right) \;\middle|\; \hat{p}(x) \in B_k\right] \mathop{\mathbb{P}}_{x\sim D}\left[\hat{p}(x) \in B_k\right] \geq 1-\epsilon\right] \\
&= \mathop{\mathbb{P}}_{Z\sim D^n}\left[\sum_{k=1}^{K} \mathbb{1}\left[\theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right)\right] \mathop{\mathbb{P}}_{x\sim D}\left[\hat{p}(x) \in B_k\right] \geq 1-\epsilon\right] \\
&\geq \mathop{\mathbb{P}}_{Z\sim D^n}\left[\sum_{k=1}^{K} \mathbb{1}\left[\theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right)\right] \mathop{\mathbb{P}}_{x\sim D}\left[\hat{p}(x) \in B_k\right] \geq 1-\epsilon \right. \\
&\qquad\qquad \left. \wedge \bigwedge_{k=1}^{K} \mathbb{1}\left[\theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right)\right] = 1\right] \\
&= \mathop{\mathbb{P}}_{Z\sim D^n}\left[\sum_{k=1}^{K} \mathbb{1}\left[\theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right)\right] \mathop{\mathbb{P}}_{x\sim D}\left[\hat{p}(x) \in B_k\right] \geq 1-\epsilon \;\middle|\; \bigwedge_{k=1}^{K} \mathbb{1}\left[\theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right)\right] = 1\right] \\
&\qquad\qquad \mathop{\mathbb{P}}_{Z\sim D^n}\left[\bigwedge_{k=1}^{K} \mathbb{1}\left[\theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right)\right] = 1\right] \\
&= \mathop{\mathbb{P}}_{Z\sim D^n}\left[\sum_{k=1}^{K} \mathop{\mathbb{P}}_{x\sim D}\left[\hat{p}(x) \in B_k\right] \geq 1-\epsilon \;\middle|\; \bigwedge_{k=1}^{K} \mathbb{1}\left[\theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right)\right] = 1\right] \\
&\qquad\qquad \mathop{\mathbb{P}}_{Z\sim D^n}\left[\bigwedge_{k=1}^{K} \mathbb{1}\left[\theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right)\right] = 1\right] \\
&= \mathop{\mathbb{P}}_{Z\sim D^n}\left[1 \geq 1-\epsilon \;\middle|\; \bigwedge_{k=1}^{K} \mathbb{1}\left[\theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right)\right] = 1\right] \mathop{\mathbb{P}}_{Z\sim D^n}\left[\bigwedge_{k=1}^{K} \mathbb{1}\left[\theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right)\right] = 1\right] \\
&= \mathop{\mathbb{P}}_{Z\sim D^n}\left[\bigwedge_{k=1}^{K} \mathbb{1}\left[\theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right)\right] = 1\right] \\
&= \mathop{\mathbb{P}}_{Z\sim D^n}\left[\bigwedge_{k=1}^{K} \theta_k \in \hat{\Theta}_{\text{CP}}\left(W_k;\frac{\delta}{K}\right)\right] \\
&\geq 1-\delta,
\end{aligned}
$$

where the last inequality holds by the union bound. □

B PROOF OF THEOREM 1

We prove this by exploiting the independence of each bin. Recall that $\hat{C}(x) \coloneqq [\hat{c}_{\hat{f}}(x) - \underline{\epsilon}_x, \hat{c}_{\hat{f}}(x) + \bar{\epsilon}_x]$, and the interval is obtained by applying the Clopper-Pearson interval with confidence level $\frac{\delta}{K}$ at each bin. Then, the following holds due the Clopper-Pearson interval for all $k \in \{1, 2, \ldots, K\}$:

$$\mathbb{P}\left[|c_{\hat{f},k} - \hat{c}_{\hat{f},k}| > \epsilon_k\right] \leq \frac{\delta}{K}$$

where $c_{\hat{f},k} \coloneqq c_{\hat{f}}(x)$ and $\hat{c}_{\hat{f},k} \coloneqq \hat{c}_{\hat{f}}(x)$ for x such that $\kappa_{\hat{f}}(x) = k$, and $\epsilon_k \coloneqq \max(\underline{\epsilon}_x, \bar{\epsilon}_x)$. By applying the union bound, the following also holds:

$$\mathbb{P}\left[\bigwedge_{k=1}^{K} |c_{\hat{f},k} - \hat{c}_{\hat{f},k}| > \epsilon_k\right] \leq \delta,$$

Considering the fact that $\mathcal{X}$ is partitioned into K spaces due to the binning and the equivalence form (10) of the PAC criterion in Definition 3, the claimed statement holds.

C PROOF OF THEOREM 2

We drop probabilities over $(x, y) \sim D$. First, we decompose the error of a cascading classifier $\mathbb{P}\left[\hat{y}_C(x) \neq y\right]$ as follows:

$$\begin{aligned}
\mathbb{P}\left[\hat{y}_C(x) \neq y\right] &= \mathbb{P}\left[\hat{y}_C(x) \neq y \wedge \Big(\hat{y}_C(x) = \hat{y}_M(x) \vee \hat{y}_C(x) \neq \hat{y}_M(x)\Big)\right] \\
&= \mathbb{P}\left[\Big(\hat{y}_C(x) \neq y \wedge \hat{y}_C(x) = \hat{y}_M(x)\Big) \vee \Big(\hat{y}_C(x) \neq y \wedge \hat{y}_C(x) \neq \hat{y}_M(x)\Big)\right] \\
&= \mathbb{P}\left[\hat{y}_C(x) \neq y \wedge \hat{y}_C(x) = \hat{y}_M(x)\right] + \mathbb{P}\left[\hat{y}_C(x) \neq y \wedge \hat{y}_C(x) \neq \hat{y}_M(x)\right],
\end{aligned}$$

where the last equality holds since the events of $\hat{y}_C(x) = \hat{y}_M(x)$ and of $\hat{y}_C(x) \neq \hat{y}_M(x)$ are disjoint. Similarly, for the error of a slow classifier $\mathbb{P}\left[\hat{y}_M(x) \neq y\right]$, we have:

$$\mathbb{P}\left[\hat{y}_M(x) \neq y\right] = \mathbb{P}\left[\hat{y}_M(x) \neq y \wedge \hat{y}_C(x) = \hat{y}_M(x)\right] + \mathbb{P}\left[\hat{y}_M(x) \neq y \wedge \hat{y}_C(x) \neq \hat{y}_M(x)\right].$$

Thus, the error difference can be represented as follows:

$$\begin{aligned}
&\mathbb{P}\left[\hat{y}_C(x) \neq y\right] - \mathbb{P}\left[\hat{y}_M(x) \neq y\right] \\
&\qquad = \mathbb{P}\left[\hat{y}_C(x) \neq y \wedge \hat{y}_C(x) \neq \hat{y}_M(x)\right] - \mathbb{P}\left[\hat{y}_M(x) \neq y \wedge \hat{y}_C(x) \neq \hat{y}_M(x)\right]. \quad (12)
\end{aligned}$$

To complete the proof, we need to upper bound (12) by ξ. Define the following events:

$$\begin{aligned}
D_m &\coloneqq \bigwedge_{i=1}^{m-1} (\hat{p}_i(x) < \gamma_i) \wedge \hat{p}_m(x) \geq \gamma_m \qquad (\forall m \in \{1, ..., M-1\}) \\
D_M &\coloneqq \bigwedge_{i=1}^{M-1} (\hat{p}_i(x) < \gamma_i) \\
E_C &\coloneqq \hat{y}_C(x) \neq \hat{y}_M(x) \\
E_m &\coloneqq \hat{y}_m(x) \neq \hat{y}_M(x) \qquad (\forall m \in \{1, ..., M-1\}) \\
F_C &\coloneqq \hat{y}_C(x) \neq y \\
F_m &\coloneqq \hat{y}_m(x) \neq y \qquad (\forall m \in \{1, ..., M-1\}) \\
G &\coloneqq \hat{y}_M(x) \neq y,
\end{aligned}$$

where $D_1, D_2, \dots, D_M$ form a partition of a sample space. Then, we have:

$$\begin{aligned}
\mathbb{P}\left[\hat{y}_C(x) \neq y \wedge \hat{y}_C(x) \neq \hat{y}_M(x)\right] &= \mathbb{P}\left[F_C \wedge E_C\right] \\
&= \mathbb{P}\left[F_C \wedge E_C \wedge \bigvee_{m=1}^{M} D_m\right] \\
&= \mathbb{P}\left[\bigvee_{m=1}^{M} \left(F_C \wedge E_C \wedge D_m\right)\right] \\
&= \sum_{m=1}^{M} \mathbb{P}\left[F_C \wedge E_C \wedge D_m\right] \\
&= \sum_{m=1}^{M} \mathbb{P}\left[F_m \wedge E_m \wedge D_m\right] \\
&= \sum_{m=1}^{M} \mathbb{P}\left[F_m \mid E_m \wedge D_m\right] \cdot \mathbb{P}\left[E_m \wedge D_m\right],
\end{aligned}$$

Similarly, we have:

$$\mathbb{P}\left[\hat{y}_M(x) \neq y \wedge \hat{y}_C(x) \neq \hat{y}_M(x)\right] = \sum_{m=1}^{M} \mathbb{P}\left[G \mid E_m \wedge D_m\right] \cdot \mathbb{P}\left[E_m \wedge D_m\right].$$

Thus, (12) can be rewritten as follows:

$$\begin{aligned}
&\mathbb{P}\left[\hat{y}_C(x) \neq y\right] - \mathbb{P}\left[\hat{y}_M(x) \neq y\right] \\
&= \sum_{m=1}^{M} \left(\mathbb{P}\left[F_m \mid E_m \wedge D_m\right] \cdot \mathbb{P}\left[E_m \wedge D_m\right] - \mathbb{P}\left[G \mid E_m \wedge D_m\right] \cdot \mathbb{P}\left[E_m \wedge D_m\right]\right) \\
&= \sum_{m=1}^{M} (e_m - e'_m) \\
&= \sum_{m=1}^{M-1} (e_m - e'_m) \\
&\leq \xi,
\end{aligned}$$

where the last equality holds since $e_M - e'_M = 0$, and the last inequality holds due to (6) with probability at least $1 - \delta$, thus proves the claim.

D Proof of Theorem 3

Suppose there is $\gamma_1^{'}$ which is different to γ_1^* and produces a faster cascading classifier than the cascading classifier with γ_1^*. Since γ_1^* is the optimal solution of (6), $\gamma_1' > \gamma_1^*$. This further implies that the less number of examples exits at the first branch of the cascading classifier with γ_1', but these examples are classified by the upper, slower branch. This means that the overall inference speed of the cascading classifier with γ_1' is slower then that with γ_1^*, which leads to a contradiction.

E Proof of Theorem 4

For clarity, we use r to denote a state x is "recoverable" (*i.e.,* $y^*(x) = 0$) and u to denote a state x is "un-recoverable" (*i.e.,* $y^*(x) = 1$). Now, note that a rollout $\zeta(x_0, \pi_{\text{shield}}) \coloneqq (x_0, x_1, \dots)$ is unsafe if (i) at some step t, we have $y^*(x_t) = u$ (*i.e.,* x_t is not recoverable), yet $\hat{y}(o_t) = r$, where $o_t = h(x_t)$ (*i.e.,* $\hat{y}$ predicts x_t is recoverable), and furthermore (ii) for every step $i \leq t-1$, $y^*(x_i) = \hat{y}(o_i) = r$—*i.e.,*

$$p_{\text{unsafe}} = \mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}}\left[\bigvee_{t=0}^{\infty}\left(\bigwedge_{i=0}^{t-1}\Big(y^*(x_i) = r \wedge \hat{y}(o_i) = r\Big) \wedge \Big(y^*(x_t) = u \wedge \hat{y}(o_t) = r\Big)\right)\right]. \quad (13)$$

Condition (i) is captured by the second parenthetical inside the probability; intuitively, it says that $\hat{y}(o_t)$ is a false negative. Condition (ii) is captured by the first parenthetical inside the probability; intuitively, it says that $\hat{y}(o_i)$ is a true negative for any $i \leq t-1$. Next, let the event E_t be

$$E_t \coloneqq \bigwedge_{i=0}^{t-1} y^*(x_i) = r \wedge y^*(x_t) = u,$$

then the following holds:

$$\begin{aligned}
&\mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}} \left[\bigvee_{t=0}^{\infty} \left(\bigwedge_{i=0}^{t-1} \Big(y^*(x_i) = r \wedge \hat{y}(o_i) = r \Big) \wedge \Big(y^*(x_t) = u \wedge \hat{y}(o_t) = r \Big) \right) \right] \\
&\quad = \mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}} \left[\bigvee_{t=0}^{\infty} \left(\bigwedge_{i=0}^{t-1} \Big(y^*(x_i) = r \wedge y^*(x_t) = u \Big) \wedge \left(\bigwedge_{i=0}^{t-1} \hat{y}(o_i) = r \wedge \hat{y}(o_t) = r \right) \right) \right] \\
&\quad = \mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}} \left[\bigvee_{t=0}^{\infty} \left(E_t \wedge \bigwedge_{i=0}^{t-1} \hat{y}(o_i) = r \wedge \hat{y}(o_t) = r \right) \right] \\
&\quad \leq \mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}} \left[\bigvee_{t=0}^{\infty} \Big(E_t \wedge \hat{y}(o_t) = r \Big) \right].
\end{aligned}$$

Recall that $\bar{p} \coloneqq \sum_{t=0}^{\infty} \mathbb{P}_{\xi \sim D_{\hat{\pi}}}[E_t]$ and $p_{\tilde{D}}(o) \coloneqq \sum_{t=0}^{\infty} p_{D_{\hat{\pi}}}(o \mid E_t) \cdot \mathbb{P}_{\xi \sim D_{\hat{\pi}}}[E_t]/\bar{p}$; then we can upper-bound (13) as follows:

$$\begin{aligned}
p_{\text{unsafe}} &\leq \mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}} \left[\bigvee_{t=0}^{\infty} \Big(E_t \wedge \hat{y}(o_t) = r \Big) \right] \\
&= \sum_{t=0}^{\infty} \mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}} \left[E_t \wedge \hat{y}(o_t) = r \right] \\
&= \sum_{t=0}^{\infty} \mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}} [\hat{y}(o_t) = r \mid E_t] \cdot \mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}} [E_t] \\
&= \sum_{t=0}^{\infty} \mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}} [\hat{y}(o) = r \mid E_t] \cdot \mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}} [E_t] \\
&= \sum_{t=0}^{\infty} \int \mathbb{1}(\hat{y}(o) = r) \cdot p_{D_{\pi_{\text{shield}}}}(o \mid E_t) \cdot \mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}} [E_t] \, \mathrm{d}o \\
&= \int \mathbb{1}(\hat{y}(o) = r) \sum_{t=0}^{\infty} p_{D_{\pi_{\text{shield}}}}(o \mid E_t) \cdot \mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}} [E_t] \, \mathrm{d}o \\
&\leq \int \mathbb{1}(\hat{y}(o) = r) \sum_{t=0}^{\infty} p_{D_{\hat{\pi}}}(o \mid E_t) \cdot \mathbb{P}_{\xi \sim D_{\hat{\pi}}} [E_t] \, \mathrm{d}o \\
&= \int \mathbb{1}(\hat{y}(o) = r) p_{\tilde{D}}(o) \bar{p} \, \mathrm{d}o \\
&= \bar{p} \cdot \mathbb{P}_{o \sim \tilde{D}}[\hat{y}(o) = r],
\end{aligned}$$

where we use the fact that E_t are disjoint by construction for the first equality, and we use o without time index t for the third equality since it clearly represents the last observation if it is conditioned on E_t. Moreover, the last inequality holds due to the following: (i) $p_{D_{\pi_{\text{shield}}}}(o \mid E_t) = p_{D_{\hat{\pi}}}(o \mid E_t)$, since E_t implies that the backup policy of π_{shield} isn't activated up to the step t, so $\pi_{\text{shield}} = \hat{\pi}$, and (ii) $\mathbb{P}_{\xi \sim D_{\pi_{\text{shield}}}}[E_t] \leq \mathbb{P}_{\xi \sim D_{\hat{\pi}}}[E_t]$, since π_{shield} is less likely to reach unsafe states by its design than $\hat{\pi}$. Thus, the constraint in (9) implies $p_{\text{unsafe}} \leq \xi$ with probability at least $1 - \delta$, so the claim follows.

F ADDITIONAL DISCUSSION

F.1 LIMITATION TO ON-DISTRIBUTION SETTING

Our PAC guarantees (*i.e.,* Theorems 1 & 5) transfer to the test distribution if it is identical to the validation distribution. We believe that providing theoretical guarantees for out-of-distribution data is an important direction; however, we believe that our work is an important stepping stone towards this goal. In particular, to the best of our knowledge, we do not know of any existing work that provides theoretical guarantees on calibrated probabilities even for the in-distribution case. One possible direction is to use our approach in conjunction with covariate shift detectors—*e.g.,* (Gretton et al., 2012). Alternatively, it may be possible to directly incorporate ideas from recent work on calibrated prediction with covariate shift (Park et al., 2020b) or uncertainty set prediction with covariate shift (Cauchois et al., 2020; Tibshirani et al., 2019). In particular, we can use importance weighting $q(x)/p(x)$, where $p(x)$ is the training distribution and $q(x)$ is the test distribution, to reweight our training examples, enabling us to transfer our guarantees from the training set to the test distribution. The key challenge is when these weights are unknown. In this case, we can estimate them given a set of unlabeled examples from the test distribution (Park et al., 2020b), but we then need to account for the error in our estimates.

F.2 BASELINES

The following includes brief descriptions on baselines that we used in experiments.

Histogram binning. This algorithm calibrates the top-label confidence prediction of $\hat{f}$ by sorting the calibration examples (x, y) into bins B_i based on their predicted top-label confidence—*i.e.,* (x, y) is associated with B_i if $\hat{p}(x) \in B_i$. Then, for each bin, it computes the empirical confidence $\hat{p}_i := \frac{1}{|S_i|} \sum_{(x,y)\in S_i} \mathbb{1}(\hat{y}(x) = y)$, where S_i is the set of labeled examples that are associated with bin B_i—*i.e.,* the empirical counterpart of the true confidence in (2). Finally, during the test-time, it returns a predicted confidence $\hat{p}_i$ for all future test examples x if $\hat{p}(x) \in B_i$.

$(1-\xi^{'})$-softmax. In fast DNN inference, a threshold can be heuristically chosen based on the desired relative error ξ and the validation error of the slow model. In particular, when a cascading classifier consists of two branches—*i.e.,* $M = 2$, the threshold of the first branch is chosen by $\gamma_1 = 1 - \xi^{'}$, where $\xi^{'}$ is the sum of ξ and the validation error of the slow model. We call this approach $(1-\xi^{'})$-softmax.

$(1-\xi^{'})$-temperature scaling. A more advanced approach is to first calibrate each branch to get better confidence. We consider using temperature scaling to do so—*i.e.,* we first calibrate each branch using the temperature scaling, and then use the branch threshold by $\gamma_1 = 1 - \xi^{'}$ when $M = 2$. We call this approach $(1-\xi^{'})$-temperature scaling.

F.3 CALIBRATION: INDUCED INTERVALS FOR ECE AND RELIABILITY DIAGRAM

ECE. The expected calibration error (ECE), which is one way to measure calibration performance, is defined as follows:

$$\text{ECE} := \sum_{j=1}^{J} \frac{|S_j|}{|S|} \left| \frac{1}{|S_j|} \sum_{(x,y)\in S_j} \hat{p}(x) - \frac{1}{|S_j|} \sum_{(x,y)\in S_j} \mathbb{1}(\hat{y}(x) = y) \right|,$$

where J is the total number of bins for ECE, $S \subseteq \mathcal{X} \times \mathcal{Y}$ is the evaluation set, and S_j is the set of labeled examples associated to the jth bin—*i.e.,* $(x, y) \in S_j$ if $\hat{p}(x) \in B_j$.

A confidence coverage predictor $\hat{C}(x)$ outputs an interval instead of a point estimate $\hat{p}(x)$ of the confidence. To evaluate the confidence coverage predictor, we remap intervals using the ECE formulation. In particular, we equivalently represents $\hat{C}(x)$ by a mean confidence $\hat{c}_{\hat{f}}(x)$ and differences from the mean—*i.e.,* $\hat{C}(x) = [\underline{c}(x), \overline{c}(x)] = [\hat{c}_{\hat{f}}(x) - \underline{\epsilon}_x, \hat{c}_{\hat{f}}(x) + \overline{\epsilon}_x]$ (see Appendix A for a description on this equivalent representation). Then, we sort each labeled example into

bins using $\hat{c}_{\hat{f}}(x)$ to form S_j. Next, we consider an interval instead of $\hat{p}(x)$ to compute ECE—*i.e.,* $\text{ECE}_{\text{induced}} := \left[\underline{\text{ECE}}, \overline{\text{ECE}}\right]$, where

$$\underline{\text{ECE}} := \sum_{j=1}^{J} \frac{|S_j|}{|S|} \inf_{\hat{p}_j \in \text{Conf}_j} \left| \hat{p}_j - \frac{1}{|S_j|} \sum_{(x,y)\in S_j} \mathbb{1}(\hat{y}(x) = y) \right|,$$

$$\overline{\text{ECE}} := \sum_{j=1}^{J} \frac{|S_j|}{|S|} \sup_{\hat{p}_j \in \text{Conf}_j} \left| \hat{p}_j - \frac{1}{|S_j|} \sum_{(x,y)\in S_j} \mathbb{1}(\hat{y}(x) = y) \right|, \text{ and}$$

$$\text{Conf}_j := \left[\underline{\text{Conf}}_j, \overline{\text{Conf}}_j\right] := \left[\min_{(x,y)\in S_j} \underline{c}(x), \max_{(x,y)\in S_j} \overline{c}(x)\right].$$

Reliability diagram. This evaluation technique is a pictorial summary of the ECE, where the x-axis represents the mean confidence $\frac{1}{|S_j|}\sum_{(x,y)\in S_j} \hat{p}(x)$ for each bin, and the y-axis represents the mean accuracy $\frac{1}{|S_j|}\sum_{(x,y)\in S_j} \mathbb{1}(\hat{y}(x) = y)$ for each bin. If an interval from a confidence coverage predictor is given, then the mean confidence is replaced by Conf_j, resulting in visualizing an interval instead of a point.

F.4 Fast DNN Inference: Cascading Classifier Training

We describe a way to train a cascading classifier with M branches. Basically, we consider to independently train M different neural networks with a shared backbone. In particular, we first train the Mth branch using a training set by minimizing a cross-entropy loss. Then, we train the $(M-1)$th branch, which consists of two parts: a backbone part from the Mth branch, and the head of the $(M-1)$th branch. Here, the backbone part is already trained in the previous stage, so we do not update the backbone part and only train the head of this branch using the *same* training set by minimizing the *same* cross-entropy loss (with the *same* optimization hyperparameters). This step is done repeatedly down to the first branch.

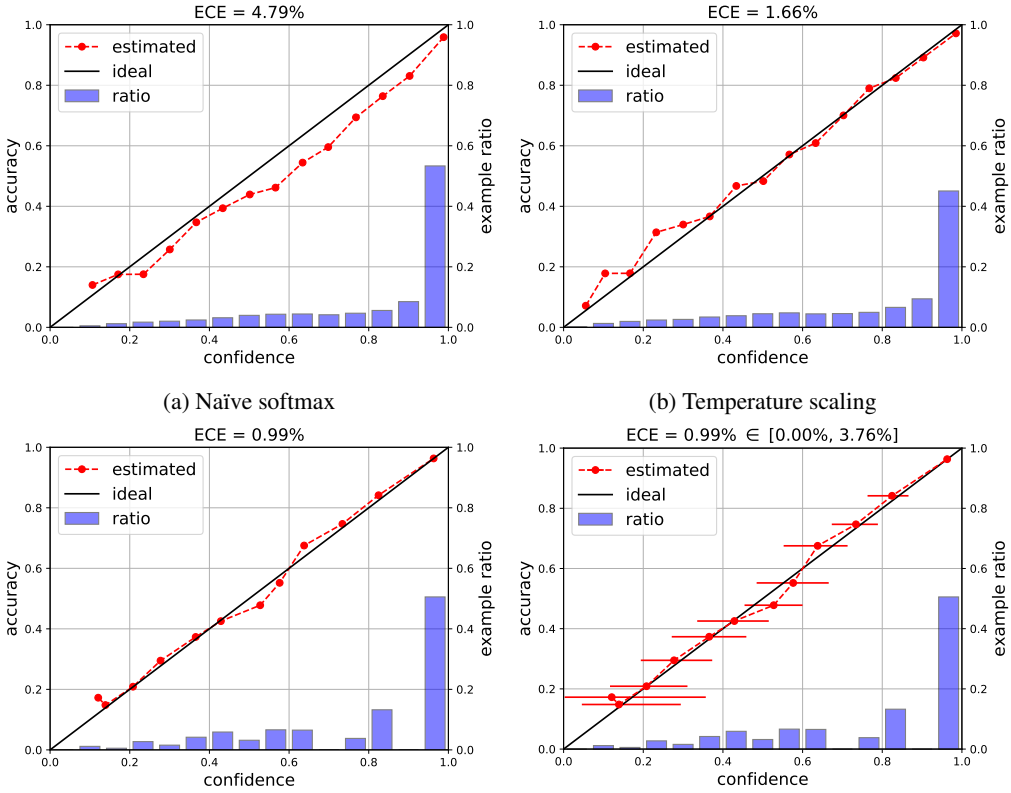
F.5 Safe Planning: Data Collection from a Simulator

We collect required data from a simulator, where a given policy $\hat{\pi}$ is already learned over the simulator. We describe how we form the necessary data from rollouts sampled from the simulator.

First, to sample a rollout ζ, we use $\hat{\pi}$ from a random initial state $x_0 \sim D$; we denote the sequence of states visited as a rollout $\zeta(x_0, \hat{\pi}) := (x_0, x_1, \dots)$. We denote the induced distribution over rollouts by $\zeta \sim D_{\hat{\pi}}$. Note that the ζ contains unsafe states since $\hat{\pi}$ is potentially unsafe. However, when constructing our recoverability classifier, we only use the sequence of safe states, followed by a single unsafe state. In particular, we let W be a set of i.i.d. sampled rollouts $\zeta \sim D_{\hat{\pi}}$. Next, for a given rollout, we consider the observation in the first unsafe state in that rollout (if one exists); we denote the distribution over such observations by $\tilde{D}$. Finally, we take Z to be a set of sampled observations $o \sim \tilde{D}$.

G Additional Experiments

G.1 Calibration



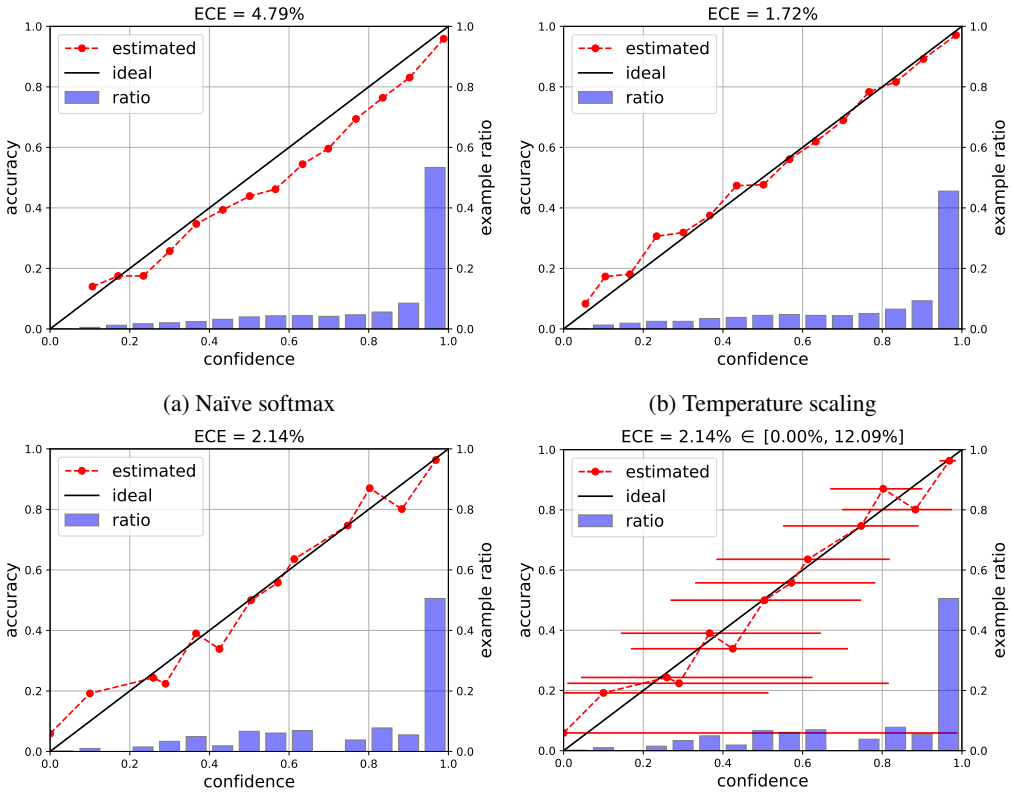
(a) Naïve softmax

(b) Temperature scaling

(c) Histogram binning

(d) Proposed

Figure 5: Calibration comparison in reliability diagrams and ECEs. The size of validation set for calibration is $20,000$. The blue histogram represents the example ratio for the corresponding bin. The diagonal line labeled "ideal" is the best reliability diagram, which produces the zero ECE. The estimated reliability diagram is represented "estimated" in dotted red. (a) The naïve softmax output from a neural network is unreliable in ECE. (b, c) The temperature scaling and histogram binning are fairly good calibration approaches, which decreases ECE. (d) The proposed approach generates "induced" intervals (see Appendix F.3) on top of the histogram binning approach, where each interval contains the ideal diagonal line with high probability. Moreover, the proposed one also produces induced ECE interval, where it contains the zero ECE with high probability.



(a) Naïve softmax

(b) Temperature scaling

(c) Histogram binning

(d) Proposed

Figure 6: Calibration comparison via reliability diagrams and ECE. The size of validation set for calibration is $2,000$. See the caption of Figure 5 for interpretation. For the induced intervals, the length of the interval is larger than that with $n = 20,000$ due to the estimation error.

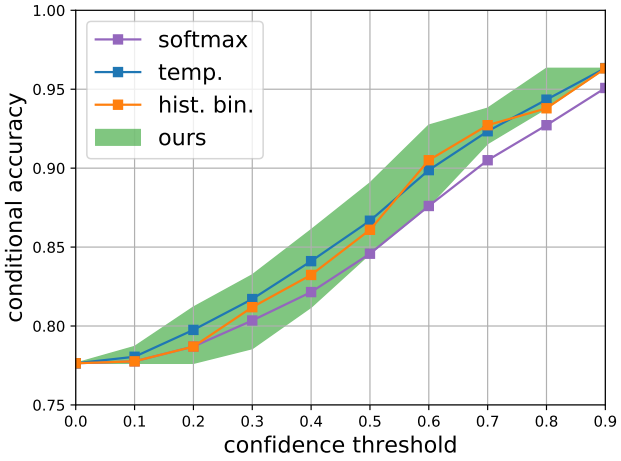
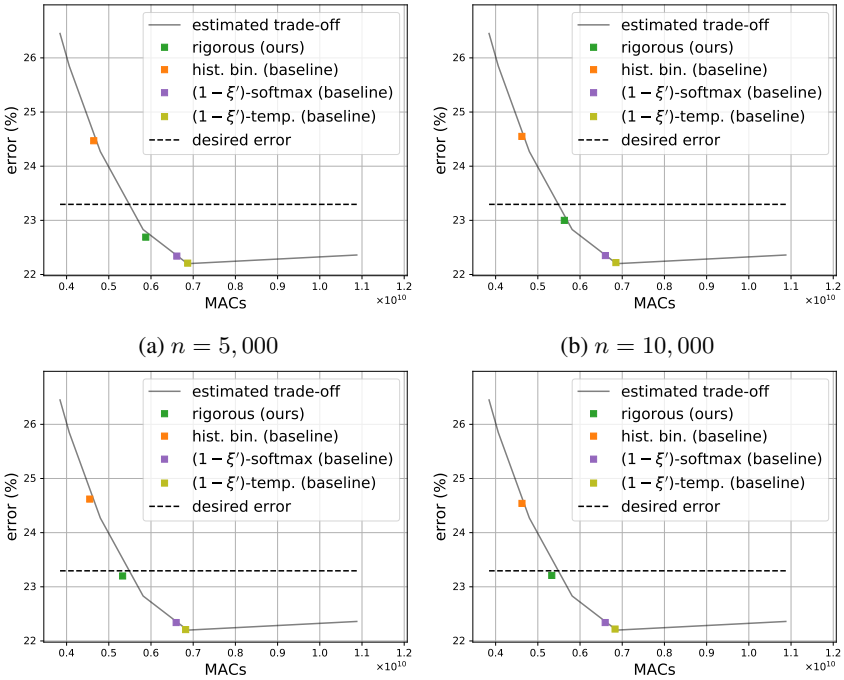


Figure 7: Accuracy-confidence plot. This plot is useful for choosing a proper confidence threshold that achieves a desired conditional accuracy (Lakshminarayanan et al., 2017); the x-axis is the confidence threshold t and the y-axis is the empirical value of the conditional accuracy $\mathbb{P}\left[\hat{y}(x) = y \mid \hat{p}(x) \geq t\right]$. Since our approach outputs an interval $[\underline{c}(x), \overline{c}(x)]$, we plot $\mathbb{P}\left[\hat{y}(x) = y \mid \underline{c}(x) \geq t\right]$ and $\mathbb{P}\left[\hat{y}(x) = y \mid \overline{c}(x) \geq t\right]$ for the upper and lower bound of the green area, respectively.

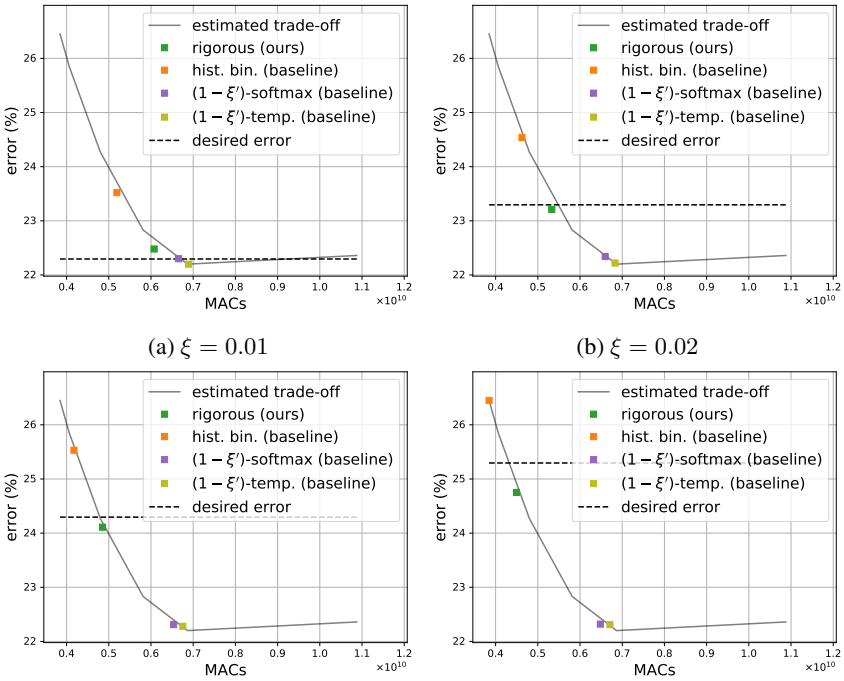
G.2 FAST DNN INFERENCE



(a) $n = 5,000$ (b) $n = 10,000$

(c) $n = 15,000$ (d) $n = 20,000$

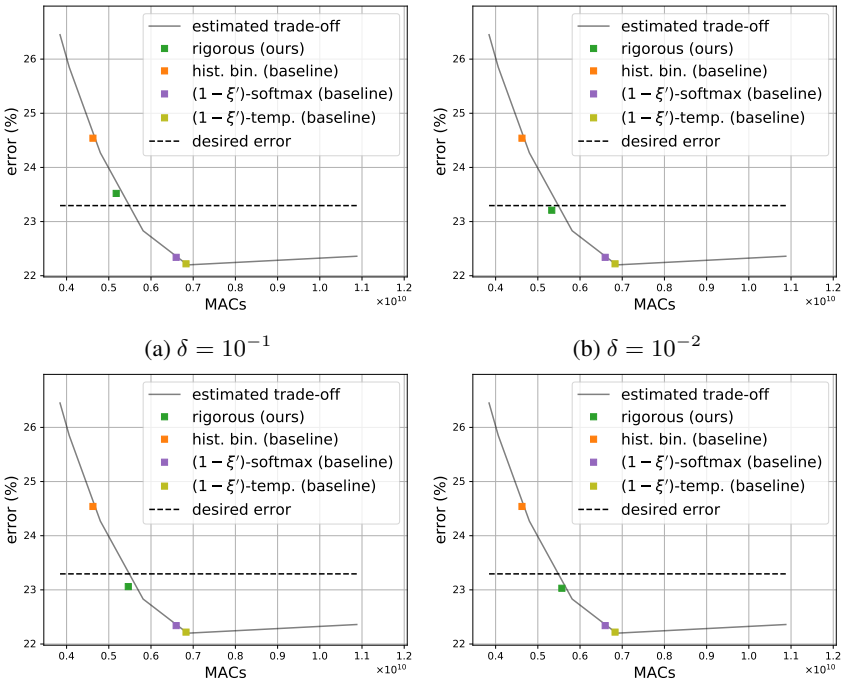
Figure 8: Ablation study on various n for $\xi = 0.02$, $\delta = 10^{-2}$, and $M = 2$. Each plot uses the shown number of samples n with the same ξ and δ. The "estimated trade-off" represents the error and MACs trade-off depending on threshold γ_1. The markers show the trade-off by the baselines. The "desired error" is a user-specified error bound, where the error of each method need to be below of this line. The proposed approach reduces the inference speed as the number of sample increases, while satisfying the desired error bound. The baselines are either fails to satisfy the desired error bound or overly conservative to satisfy the bound.



(a) $\xi = 0.01$ (b) $\xi = 0.02$

(c) $\xi = 0.03$ (d) $\xi = 0.04$

Figure 9: Ablation study on various ξ for $n = 20,000$, $\delta = 10^{-2}$, and $M = 2$. Each plot uses the shown desired relative error ξ with the same n and δ. The "estimated trade-off" represents the error and MACs trade-off depending on threshold γ_1. The markers show the trade-off by the shown baselines. The "desired error" is a user-specified error bound, where the error of each method need to be below of this line. The proposed approach allows more error as at most specified by the desired error, to reduce the inference speed. Other baselines either overly increase the error or conservatively maintain overly low error.

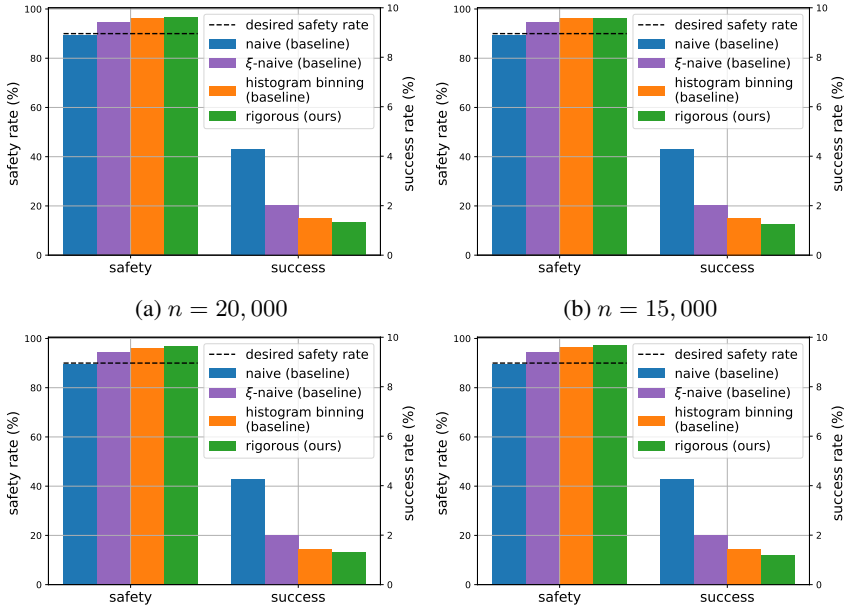


(a) $\delta = 10^{-1}$ (b) $\delta = 10^{-2}$

(c) $\delta = 10^{-3}$ (d) $\delta = 10^{-4}$

Figure 10: Ablation study on various δ for $n = 20,000$, $\xi = 0.02$, and $M = 2$. Each plot uses the shown misconfidence level δ with the same n and ξ. The "estimated trade-off" represents the error and MACs trade-off depending on threshold γ_1. The markers show the trade-off by the shown baselines. The "desired error" is a user-specified error bound, where the error of each method need to be below of this line. The proposed approach produces larger error gap toward the desired error as we enforce stronger confidence level—*i.e.,* from $\delta = 10^{-1}$ to $\delta = 10^{-4}$. Note that other baselines do not depend on δ as designed.

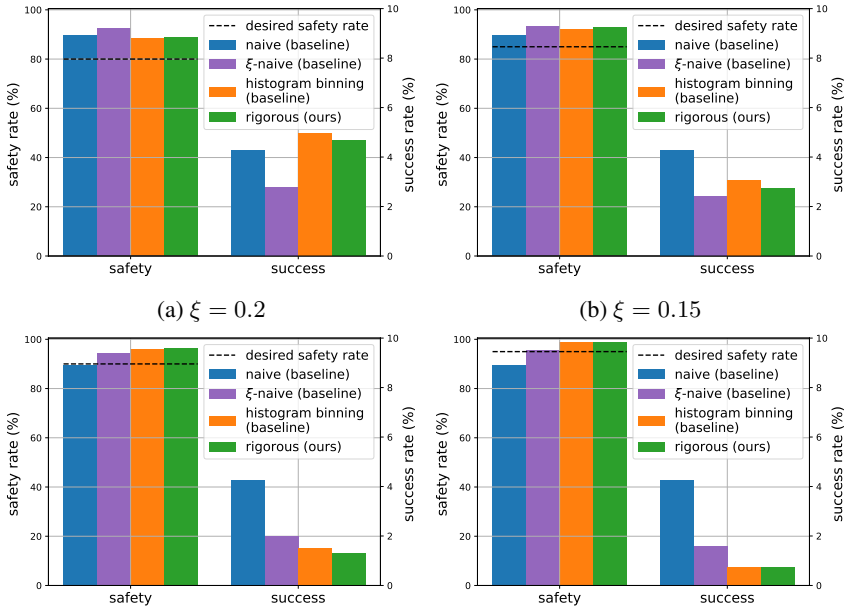
G.3 SAFE PLANNING



(a) $n = 20,000$ (b) $n = 15,000$

(c) $n = 10,000$ (d) $n = 5,000$

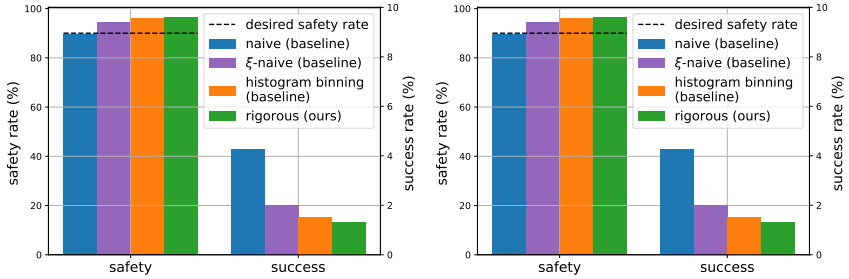
Figure 11: Ablation study on various n for $\xi = 0.1$ and $\delta = 10^{-2}$. Each plot uses the shown sample size n with the same ξ and δ. The "naive", "ξ-naive", and "histogram binning" represent baseline results on the safety rate and success rate. The proposed approach is labeled "rigorous". The desired safety rate is a user-specified rate, where the safety rate of each method need to be above of this line. The safety rate of the proposed approach is above the desired safety rate, and it tends to be closer to the desired safety rate as n increases.



(a) $\xi = 0.2$ (b) $\xi = 0.15$

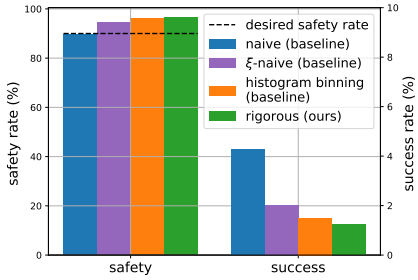
(c) $\xi = 0.1$ (d) $\xi = 0.05$

Figure 12: Ablation study on various ξ for $n = 20,000$ and $\delta = 10^{-2}$. Each plot uses the shown desired unsafe rate ξ with the same n and δ. The "naive", "ξ-naive", and "histogram binning" represent baseline results on the safety rate and success rate. The proposed approach is labeled "rigorous". The desired safety rate is a user-specified rate, where the safety rate of each method need to be above of this line. The safety rate of the proposed approach is closely above the desired safety rate to satisfy the safety constraint. However, the naive can violate the safety constraint, and the ξ-naive can be overly optimistic. The histogram binning and ξ-naive look empirically fine, but they can in theory violate desired safety rate (*e.g.,* see Figure 4b).

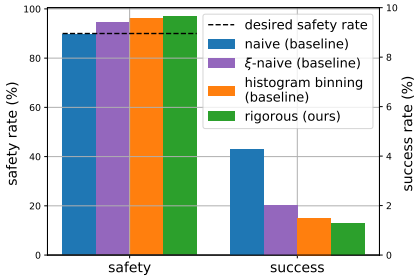


(a) $\delta = 10^{-1}$

(b) $\delta = 10^{-2}$



(c) $\delta = 10^{-3}$



(d) $\delta = 10^{-4}$

Figure 13: Ablation study on various δ for $n = 20,000$ and $\xi = 0.1$. Each plot uses the shown misconfidence level ξ with the same n and ξ. The "naive", "ξ-naive", and "histogram binning" represent baseline results on the safety rate and success rate. The proposed approach is labeled "rigorous". The desired safety rate is a user-specified rate, where the safety rate of each method need to be above of this line. The proposed approach produces more conservative safety rates—*i.e.,* larger gap between the estimated safety rate and desired safety rate—as we enforce stronger confidence level—*i.e.,* from $\delta = 10^{-1}$ to $\delta = 10^{-4}$. Note that other baselines do not depend on δ as designed.