PAGER: Accurate Failure Characterization in Deep Regression Models

Jayaraman J. Thiagarajan [1] **Vivek Narayanaswamy** [1] **Puja Trivedi** [2] **Rushil Anirudh** [3]

Abstract

Safe deployment of AI models requires proactive detection of failures to prevent costly errors. To this end, we study the important problem of detecting failures in deep regression models. Existing approaches rely on epistemic uncertainty estimates or inconsistency w.r.t the training data to identify failure. Interestingly, we find that while uncertainties are necessary they are insufficient to accurately characterize failure in practice. Hence, we introduce PAGER (Principled Analysis of Generalization Errors in Regressors), a framework to systematically detect and characterize failures in deep regressors. Built upon the principle of anchored training in deep models, PAGER unifies both epistemic uncertainty and complementary manifold non-conformity scores to accurately organize samples into different risk regimes.

1. Introduction

Ensuring the safety of AI models involves proactively detecting failures to help avoid costly errors. While existing efforts have predominantly focused on classification models (Guillory et al., 2021; Narayanaswamy et al., 2022; Baek et al., 2022), we are interested more challenging problem of failure detection in deep regressors. Though continuous-valued prediction is prevalent in high-impact applications including healthcare (Luo et al., 2022), physical sciences (Raissi et al., 2019) and reinforcement learning, the notion of failure in regression models is often application-specific.

A common approach is to use epistemic uncertainty (Lakshminarayanan et al., 2017; Gal & Ghahramani, 2016; He et al., 2020; Amini et al., 2020) as a surrogate for expected risk (Lahlou et al., 2023). However, we uncover a crucial insight that uncertainties alone are insufficient for a comprehensive characterization of failures in regression models. Figure 1 empirically illustrates the lack of correlation between uncertainty and the true risk using a simple 1D function (with two experiment designs) – low uncertainty regimes can still correspond to a higher risk due to feature heterogeneity in the training data (Seedat et al., 2022), and similarly data regimes outside the training support may correspond to low risk if the model extrapolates well.

To circumvent this, we introduce PAGER (Principled Analysis of Generalization Errors in Regressors), a new framework for failure analysis. A key contribution of this work is that we advocate for incorporating manifold non-conformity, *i.e.*, adherence to the joint data distribution, as an essential complement to uncertainties. Building upon the principle of anchored training (Thiagarajan et al., 2022), we make a critical finding that non-conformity scores can be estimated through *reverse anchoring* without the need for auxiliary models. Additionally, we propose a flexible analysis of model errors through the concept of risk regimes, thus avoiding the need for a rigid definition of failure or additional calibration data. Finally, we introduce a suite of metrics to holistically assess failure detectors in regression tasks. Empirical results reveal that, when compared to state-of-the-art detectors, the risk regimes identified by PAGER align best with the true risk.

2. Background and Related Work

Preliminaries. We consider a predictive model F, parameterized by θ, trained on a labeled dataset $\mathcal{D} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{M}$ with M samples. Each input $\mathrm{x}_i \in X$ and label $\mathrm{y}_i \in y$ belong to the spaces of inputs X (in $d-$dimensions) and continuous-valued targets y respectively. Given a non-negative loss function $\mathcal{L}$, e.g., absolute error $|\mathrm{y} - \hat{\mathrm{y}}|$, the sample-level risk of a predictor can be defined as $\mathrm{R}(\mathrm{x}; F_\theta) = \mathbb{E}_{y|\mathrm{x}}\, \mathcal{L}(\mathrm{y}, F_\theta(\mathrm{x}))$. Since estimating true risk is non-trivial due to the need for access to the unknown joint distribution $P(X, y)$, an alternative is to identify risk regimes in accordance to the expected risk. We now define the different risk regimes that one needs to characterize: (i) In-distribution (ID): This is the scenario where $P(\mathrm{x}_t \in X) > 0$ and $P(\mathrm{x}_t \in \mathcal{D}) > 0$, *i.e.*, there is

[1]Lawrence Livermore National Labs, CA, USA [2]University of Michigan, USA [3]Amazon, CA, USA. Correspondence to: Jayaraman J. Thiagarajan <jjayaram@llnl.gov>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

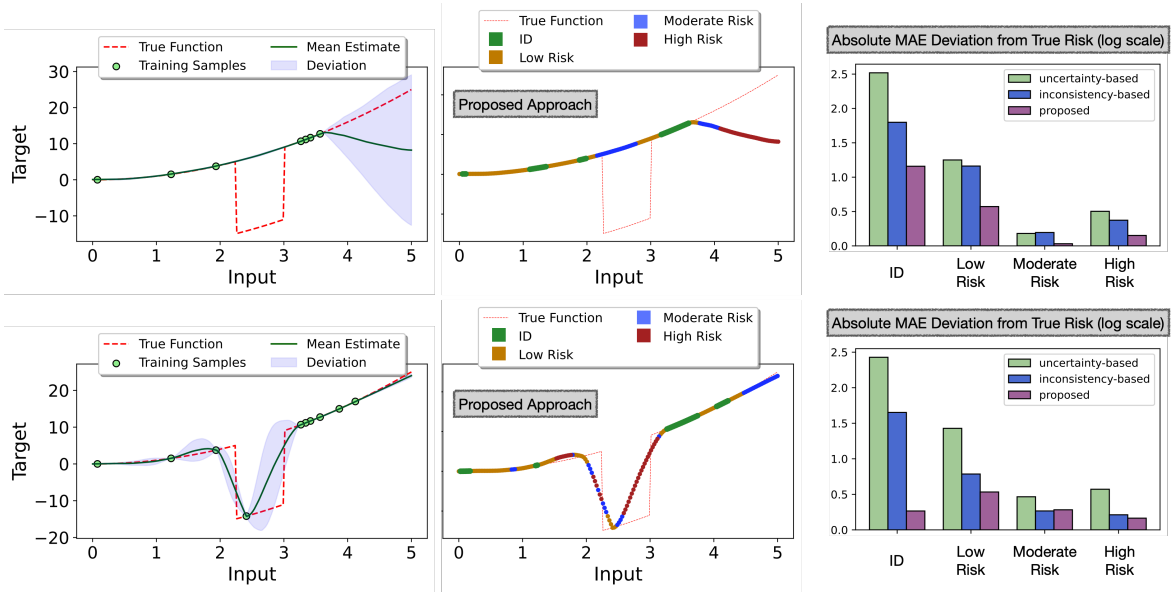


Figure 1. **Epistemic uncertainty, while necessary, is not sufficient to completely characterize all risk regimes.** Top: Out-of-support (OOS) samples in the range of $[2.2 - 2.7]$ exhibit low uncertainty but moderate risk due to significant deviation from true function. Bottom: Even with better experiment designs, uncertainty alone in the extrapolating regime $[4.5 - 5]$ is unreliable due to potential drift from the truth. We propose PAGER , a framework that leverages anchoring (Thiagarajan et al., 2022) to unify prediction uncertainty and non-conformity to the training data manifold. PAGER accurately flags those erroneous regimes as Moderate Risk (shown in blue) and outperforms existing baselines in accurately categorizing samples consistent with the true risk (lower MAE).

likelihood for observing the test sample in the training dataset; (ii) Out-of-Support (OOS): The scenario where $P(\mathrm{x}_t \in X) > 0$ but $P(\mathrm{x}_t \in \mathcal{D}) = 0$, *i.e.*, the train and test sets have different supports, even though they are drawn from the same space; (iii) Out-of-Distribution (OOD): This is the scenario where $P(\mathrm{x}_t \in X) = 0$, *i.e.*, the input spaces for train and test data are disjoint. Figure 2 illustrates the differences between OOS and OOD using 1D and 2D examples. In case of 1D, OOS corresponds to regimes where the likelihood of observing data in the training support is zero but is non-zero in the input-space. Similarly, in 2D, OOS constitutes regimes with new combinations of features (light blue) which are not jointly seen in the train data.

Failure Characterization. In classification tasks, incorrectly assigned labels are considered failures. Hence, failure detectors can estimate either the sample-level *correctness* (Ng et al., 2022; Jiang et al., 2022) or distribution-level scores such as generalization gap (Guillory et al., 2021; Narayanaswamy et al., 2022; Chen et al., 2021; Jiang et al., 2019; Deng & Zheng, 2021). Risk estimation in regression models is more challenging since the the notion of failure is highly subjective, *i.e.*, permissible tolerance levels on prediction errors can vary across use-cases. Among existing methods, DEUP (Lahlou et al., 2023) is a recent approach that utilized predictive uncertainty as a surrogate for total risk, which we illustrated to be insufficient for failure detection in Figure 1. Conformal prediction (CP) forms another popular class of uncertainty estimation methods (Vovk et al., 2005; Lei et al., 2018), that can be leveraged to identify risk regimes. However, with OOS and OOD data, the exchangeability assumption made by CP frameworks is violated (Tibshirani et al., 2019) and hence the estimated intervals can be erroneous. Finally, approaches such as DataSUITE (Seedat et al., 2022) qualify failure solely based on feature inconsistency with respect to the training data distribution. However, our results show that such methods are incapable of identifying errors in OOS regimes.

Anchoring in Deep Models. Anchored training involves reparameterizing an input sample x (referred to as the *query*) into a tuple comprising an *anchor* r randomly drawn from the training data and the residual $\Delta\mathrm{x}$ denoted by $[\mathrm{r}, \Delta\mathrm{x}] = [\mathrm{r}, \mathrm{x} - \mathrm{r}]$ (Thiagarajan et al., 2022). It induces a joint distribution that depends not only on $P(X)$, but also on the distribution of residuals $P(\Delta)$. In practice, the sole modification lies in the input layer, requiring additional dimensions for vector-valued data or channels for images, and a modest addition of parameters to the first layer. This approach is adaptable to any architecture (e.g., MLP, CNN, ViT). During training, we enforce consistency in predictions for a query x across all possible anchors. Consequently, at inference time, one can obtain predictions and corresponding uncertainties by marginalizing out the anchor choice. A detailed description can be found in Appendix A.1.

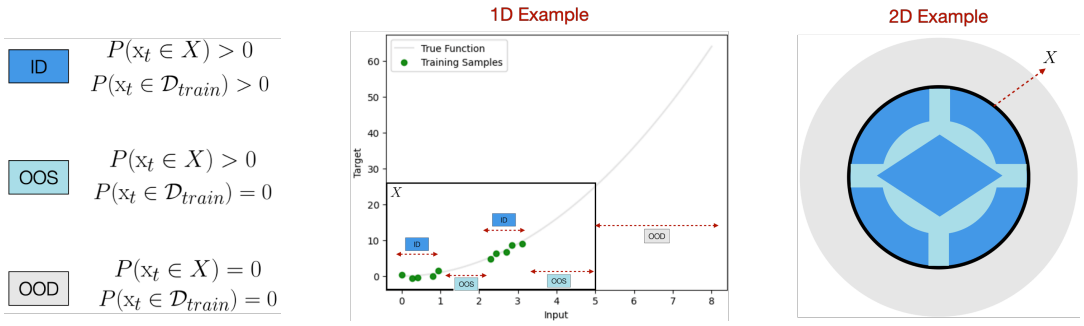


Figure 2. **An illustration of different risk regimes.** Using examples in 1D and 2D, we show ID, OOS and OOD regimes.

3. Failure Detection in Deep Regressors

The central idea of our approach is to obtain not only uncertainty estimates, but also manifold non-conformity (MNC) scores, for failure detection. This is motivated by the observation that, regardless of the uncertainty, a model can induce a large error for test sample x_t, when $(\mathrm{x}_t, \mathrm{y}_t) \notin P(X, y)$, *i.e.*, the risk can be high when the sample does not adhere to the data manifold. While there exist several approaches for estimating epistemic uncertainty (Gawlikowski et al., 2023; Yang et al., 2021), measuring non-conformity without ground truth is not straightforward. Consequently, it is typical to adopt scoring functions only based on inputs (Seedat et al., 2022) or utilize CP strategies to transform scores into calibrated intervals (Teng et al., 2023). While the former approach does not leverage the characteristics of the task, the latter is not applicable in our scenario due to the violation of the exchangeability condition w.r.t OOS and OOD regimes. Hence, we propose an alternative approach based on anchored neural networks.

Uncertainty via Forward Anchoring. An anchored model is trained by transforming a sample x into a tuple, $[\mathrm{r}, \mathrm{x} - \mathrm{r}]$ based on an anchor r, which is also drawn randomly from training data. Building upon findings from (Thiagarajan et al., 2022), multiple anchors can be used to obtain the mean predictions and uncertainties for a test sample as follows:

$$\mu(y_t|\mathrm{x}_t) = \frac{1}{K}\sum_{k=1}^{K} F([\mathrm{r}_k, \mathrm{x}_t - \mathrm{r}_k]);$$
$$\sigma(y_t|\mathrm{x}_t) = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(F([\mathrm{r}_k, \mathrm{x}_t - \mathrm{r}_k]) - \mu)^2}, \quad (1)$$

where μ and σ are estimated by marginalizing across K anchors $\{\mathrm{r}_k\}_{k=1}^{K}$ sampled from $\mathcal{D}$.

Non-conformity via Reverse Anchoring. Turning our attention to the assessment of non-conformity, we make a noteworthy observation regarding the flexibility of an anchored neural network. It is not only able to capture the relative representation of a query in relation to an anchor, but also the reverse scenario. To elaborate, for a given test sample x_t, we swap the roles of query x_t and anchor r to obtain the prediction for the anchor as $F([\mathrm{x}_t, \mathrm{r} - \mathrm{x}_t])$. Since the ground truth function value is known for the training samples, we can measure the non-conformity score for a query based on its ability to accurately recover the target of the anchor. Note, unlike existing approaches, this can be directly applied to unlabeled test samples and does not require explicit calibration.

Looking from another perspective, the original 'anchor-centric' model provides reliable predictions for an input $[\mathrm{r}, \Delta]$ only when $\mathrm{r} \in \mathcal{D}$ and $\Delta \in P(\Delta)$. However, for OOS or OOD samples, if $\Delta \notin P(\Delta)$, the estimated uncertainty becomes large everywhere, and thus becomes inherently unreliable to rank them by levels of expected risk. In contrast, the proposed 'query-centric' score overcomes this by directly measuring the discrepancy with respect to the ground truth target. We define our MNC score as follows:

$$\texttt{Score}_1(\mathrm{x}) = \max_{\mathrm{r}\in\mathcal{D}} \left\| \mathrm{y}_\mathrm{r} - F([\mathrm{x}, \mathrm{r} - \mathrm{x}]) \right\|_1 \quad (2)$$

Note that, we measure the largest discrepancy across the training dataset. In practice, this can be done for a small batch of randomly chosen training samples (e.g., 100).

Resolving Moderate and High Risk Regimes Better. A closer look of equation 2 reveals that for samples that are far away from the training manifold, the model prediction can be uniformly bad (i.e., extrapolation), as both $\mathrm{x} \notin \mathcal{D}$ and $\Delta \notin P(\Delta)$. This can make distinguishing between samples with moderate and high risk challenging. To mitigate this, we propose to transform both the query x (used as the anchor in reverse anchoring) and Δ to be in-distribution so that the anchored model F can produce reliable predictions. We achieve this using the following optimization problem:

$$\texttt{Score}_2(\mathrm{x}) = \max_{r\in\mathcal{D}} \left\| \mathrm{x} - \arg\min_{\bar{\mathrm{x}}} \left(\left\| \mathrm{y}_\mathrm{r} - F([\bar{\mathrm{x}}, \mathrm{r} - \bar{\mathrm{x}}]) \right\|_1 + \lambda \mathcal{R}(\bar{\mathrm{x}}) \right) \right\|_2, \quad (3)$$

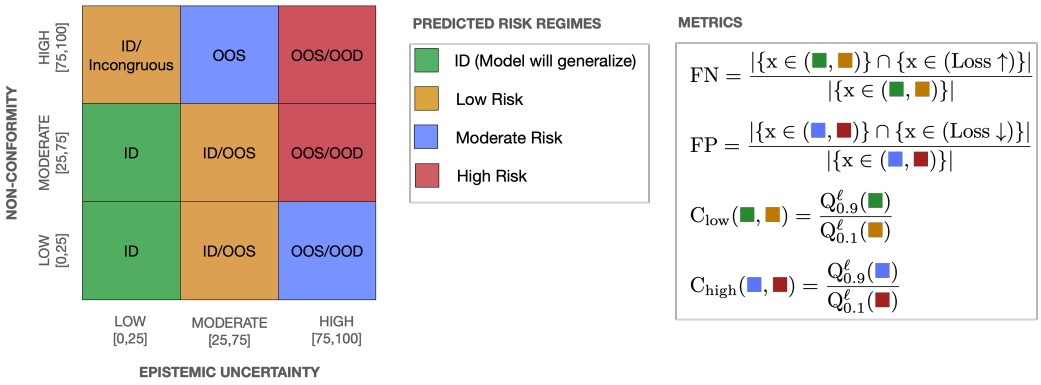


Figure 3. **Overview of our proposed framework.** PAGER organizes test examples into bins (*low*, *moderate* and *high*) using both predictive uncertainty and MNC scores. With such a categorization, PAGER associates samples into 4 levels of expected risk (ID, Low Risk, Moderate Risk and High Risk). We also advocate a suite of metrics that enables a holistic assessment of failure detectors.

where $\mathcal{R}(\bar{\mathrm{x}}) = \left\|\bar{\mathrm{x}} - A([\mathrm{x}, \bar{\mathrm{x}} - \mathrm{x}])\right\|_2 + \left\|\mathrm{x} - A([\bar{\mathrm{x}}, \mathrm{x} - \bar{\mathrm{x}}])\right\|_2$. In this approach, the score is measured as the discrepancy in the input space to a new fictitious sample that serves as an intermediate anchor, such that its prediction matches the known prediction on the training sample. In other words, we optimize the modification of the query sample x to $\bar{\mathrm{x}}$ in such a way that we accurately match the true target for the anchor r. The MNC is then quantified as the amount of movement required in x to match the target. To ensure that the resulting $\bar{\mathrm{x}}$ remains within the input data manifold, we incorporate a regularizer $\mathcal{R}(\bar{\mathrm{x}})$. Specifically, we train an anchored auto-encoder A on the training dataset $\mathcal{D}$ and enforce cyclical consistency, where A is required to recover x using $\bar{\mathrm{x}}$ as the anchor and vice versa. We provide the algorithm listings and details of all these methods in Appendix A.2.

3.1. PAGER Framework

Since it is challenging to accurately estimate and interpret sample-level error estimates, particularly in OOS or OOD regimes, a more tractable approach is to analyze sample groups that correspond to varying levels of expected risk. To this end, PAGER organizes a set of test samples into different risk regimes. Without loss of generality, we assume that a typical test set contains samples close to the training distribution, as well as OOS and potential OOD samples.

In our implementation, both scores are split into three bins using conditional quantile ranges (*low*:$[0, 25]$, *moderate*:$[25, 75]$ and *high*:$[75, 100]$), thereby creating a non-trivial partition of the test data into risk regimes. Note that, the number of bins and the threshold choices used are only for demonstration, and can be adapted based on application needs (e.g., pick top $k\%$ of high-risk samples). As discussed earlier, PAGER does not involve any calibration step and can directly work on the unlabeled test set. We now describe the different risk regimes in PAGER.

ID (■): The model generalizes well in this regime and is expected to produce low prediction error. In PAGER, this corresponds to samples with low uncertainty as well as low/moderate MNC scores;

Low Risk (■): Even when the uncertainty is low, the model can produce higher error than the ID samples, when there is incongruity (e.g., samples within a neighborhood having different target values). Similarly, for OOS samples with moderate uncertainties, the model can still extrapolate well and produce reduced risk. Hence, we define this regime as the collection of (low uncertainty, high MNC) and (moderate uncertainty, low/moderate MNC) samples;

Moderate Risk (■): Since epistemic uncertainties can be inherently miscalibrated, OOS samples, which the model cannot extrapolate to, can be associated with moderate uncertainties. On the other hand, the model could reasonably generalize to OOD samples that are flagged with high uncertainties. Hence, we define this regime as the collection of (moderate uncertainty, high MNC) and (high uncertainty, low/moderate MNC) samples;

High Risk (■): Finally, when both the uncertainty and non-conformity scores are high, there is no evidence that the model will behave predictably on those samples. In practice, this can correspond to both OOS and OOD samples.

3.2. Evaluation Metrics

Evaluation metrics typically adopted to assess failure detectors for regression models, e.g., Spearman correlation between the true risk and the predicted risk on a held-out test set or the average error in top inconsistent samples, do not comprehensively reflect the quality of detectors in different risk regimes. Hence, we propose to utilize the following metrics (see Figure 3):

False Negatives (FN)(↓) This is the most important metric

Table 1. **Metrics for 1D Benchmarks.** We report the FN, FP, C_{low} and C_{high} metrics on evaluation data across the entire target regime (lower the better). Note that for every metric, we identify the first and second best approach across the different benchmarks.

Metric	Method	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	Metric	Method	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
FN↓	DEUP	6.19	6.56	16.57	27.13	C_{low}↓	DEUP	65.90	57.86	34.13	169.54
	DataSUITE	14.03	8.8	16.31	7.2		DataSUITE	59.42	24.61	22.44	89.51
	MNC-only	6.19	2.26	13.73	8.84		MNC-only	57.54	40.1	31.66	52.24
	Anchor UQ-only	5.95	5.37	14.49	11.80		Anchor UQ-only	40.7	19.88	25.59	98.92
	PAGER ($\texttt{Score}_1$)	5.61	0.0	11.63	2.40		PAGER ($\texttt{Score}_1$)	28.08	7.19	19.94	12.05
	PAGER ($\texttt{Score}_2$)	4.79	5.59	8.43	5.59		PAGER ($\texttt{Score}_2$)	20.61	17.82	16.57	19.74
FP↓	DEUP	8.91	3.41	8.54	9.09	C_{high}↓	DEUP	91.64	4.47	59.46	16.56
	DataSUITE	18.67	15.97	19.96	5.33		DataSUITE	3.66	46.02	58.32	6.81
	MNC-only	9.93	10.42	8.82	12.18		MNC-only	33.09	18.85	29.98	20.31
	Anchor UQ-only	5.05	4.93	6.54	6.01		Anchor UQ-only	36.05	7.75	17.92	11.56
	PAGER ($\texttt{Score}_1$)	2.67	0.0	4.67	6.67		PAGER ($\texttt{Score}_1$)	3.09	3.43	8.78	6.88
	PAGER ($\texttt{Score}_2$)	1.33	2.67	4.33	4.00		PAGER ($\texttt{Score}_2$)	3.09	4.67	10.99	5.71

in applications where the cost of missing to detect high risk failures is high. We measure the ratio of samples in the ID or low risk regimes that actually have high true risk (top 20^{th} percentile of all test samples).

False Positives (FP)(↓) This reflects the penalty for scenarios where arbitrarily flagging harmless samples as failures. Here, we measure the ratio of samples in the moderate or high risk regimes that actually have low true risk (bottom 20^{th} percentile of all test samples).

Confusion in Low Risk Regimes (C_{low})(↓) A common challenge in fine-grained sample grouping (ID vs low risk) is that the detector can confuse samples between neighboring regimes. We define this metric as the ratio between the 90^{th} percentile of the ID regime and the 10^{th} percentile of the low risk regime. The selection of the 90^{th} percentile and 10^{th} percentile to gauge the error ratio is intentionally made stringent. However, one can relax these thresholds depending on the desired error tolerance in practice.

Confusion in High Risk Regimes (C_{high})(↓) This is similar to the previous case and instead measures the confusion between the moderate and high risk regimes.

4. Experiments

Datasets. We evaluate the effectiveness of PAGER using a suite of tabular and imaging benchmarks.

1. 1D Benchmark Functions: We consider the following black-box functions t:
   (a) $f_1(x) = \begin{cases} x^2 & \text{if } x < 2.25 \text{ or } x > 3.01 \\ x^2 - 20 & \text{otherwise} \end{cases}$ (Figure 1)
   (b) $f_2(x) = \sin(2\pi x)$, $x \in [-0.5, 2.5]$
   (c) $f_3(x) = a\exp(-bx) + \exp(\cos(cx)) - a - \exp(1)$, $x \in [-5, 5]$, $a = 20, b = 0.2, c = 2\pi$
   (d) $f_4(x) = \sin(x)\cos(5x)\cos(22x)$, $x \in [-1, 2]$

   In each of these functions, we used 200 test samples drawn from an uniform grid and computed the metrics.

2. HD Regression Benchmarks: We also considered a set of regression datasets comprising different domains and varying dimensionality. (a) Camel (2D), (b) Levy (2D) (ben) characterized by multiple local minima, (c) Airfoil (5D), (d) NO2 (7D), (e) Kinematics (8D), (f) Puma (8D) (del) which are simulated datasets of the forward dynamics of different robotic control arms, (g) Boston Housing (13D) (bh), (h) Ailerons (39D) (ail) which is a dataset for predicting control action of the ailerons of an F16 aircraft, and (i) Drug-Target Interactions (32000D). For each benchmark, we created two variants: Gaps (training exposed to data with targets between $(0-30^{th})$ and $(60-100^{th})$ percentiles) and Tails (training exposed to $(0-70^{th})$ percentiles of the targets). Additionally, we considered the Skillcraft dataset (Yao et al., 2022), which represents real-world distribution shifts arising from change in the league index.

3. Image Regression: We used three image regression benchmarks namely chair (yaw) angle, cell count and CIFAR-10 rotation prediction respectively. In each case, we synthesized two different variants – tails and gaps in the target variable, similar to the HD regression experiments. The range of target values used in each of the experiments can be found in Figure 5.

Baselines. (i) DEUP (Lahlou et al., 2023) is a state-of-the-art epistemic uncertainty-based failure detection approach. It utilizes a post-hoc, auxiliary error predictor that learns to predict the risk of the underlying model which is considered

Table 2. **Assessing the identified risk regimes for regression benchmarks (Gaps) with dimensionality ranging between 2 and 32,000.** We report the FN, FP, C_{low} and C_{high} metrics on evaluation data across the entire target regime (lower the better). Note that for every metric, we identify the first and second best approach across the different benchmarks.

Metrics	Method	Camel	Levy	Airfoil	NO2	Kinematics	Puma	Housing	Ailerons	DTI
FN↓	DEUP	15.79	9.25	8.81	2.27	17.58	13.21	11.46	14.39	16.51
	DataSUITE	21.74	19.69	5.95	6.58	18.40	16.77	17.71	11.23	29.18
	PAGER ($\texttt{Score}_1$)	12.15	10.86	0.75	0.0	6.42	10.37	6.25	0.91	9.26
	PAGER ($\texttt{Score}_2$)	11.39	10.65	1.04	0.93	6.38	10.84	7.29	1.20	10.11
FP↓	DEUP	17.48	10.04	6.24	11.79	18.67	12.05	10.34	15.96	19.73
	DataSUITE	15.74	15.32	6.35	18.33	10.67	17.33	12.07	8.03	30.93
	PAGER ($\texttt{Score}_1$)	3.36	5.04	3.56	4.18	12.04	9.67	8.62	4.05	9.94
	PAGER ($\texttt{Score}_2$)	7.56	4.18	3.82	3.05	10.67	8.83	9.07	1.33	10.29
C_{low}↓	DEUP	50.59	34.67	28.23	19.32	10.71	14.82	13.86	15.55	5.46
	DataSUITE	42.92	71.06	37.11	47.6	21.96	15.26	14.8	30.78	12.8
	PAGER ($\texttt{Score}_1$)	14.05	13.62	11.8	7.01	12.91	12.44	13.33	12.90	2.56
	PAGER ($\texttt{Score}_2$)	10.13	10.41	9.93	6.15	10.93	8.71	10.42	11.18	2.83
C_{high}↓	DEUP	15.47	12.42	17.99	7.71	11.28	6.18	3.36	23.94	10.05
	DataSUITE	37.51	36.55	14.85	6.82	5.97	10.57	22.56	4.23	18.93
	PAGER ($\texttt{Score}_1$)	8.89	10.39	4.72	4.12	7.71	8.09	3.19	1.69	5.22
	PAGER ($\texttt{Score}_2$)	11.03	9.37	3.90	2.83	7.01	7.30	2.95	1.65	4.19

as a surrogate for uncertainties; (ii) DataSUITE (Seedat et al., 2022) is a task-agnostic approach that estimates the inconsistencies in the data regimes to assess data quality. Both baselines rely on the use of additional, curated calibration data to either train the error predictor in case of DEUP, or to obtain non-conformity scores that assess the sample level quality in the latter.

Training Protocols. For experiments on all tabular benchmarks, we used an MLP (Bishop & Nasrabadi, 2007) with 4 layers each with a hidden dimension of 128. While we used the WideResNet40-2 model (Zagoruyko & Komodakis, 2016) for the first two image regression datasets, in the case of CIFAR-10, we randomly applied a rotation transformation [0 - 90 degrees] to each 32×32×3 image and trained a ResNet-34 model to predict the angle of rotation. For evaluation, we used the held-out test sets (e.g., 10K randomly rotated images for CIFAR-10). Without loss of generality, we used the L_1 objective for training all the models. We provide the implementation details along with hyper-parameters choices in Appendix A.3.

5. Main Findings & Discussion

5.1. Results on 1D benchmarks

We expect an effective failure detector to align well with the training distribution (ID) and progressively flag regions of low, moderate and high risk as we move away from the inferred data manifold. From the results in Table 1 for standard 1D benchmark functions, PAGER achieves this effectively, while also consistently outperforming the baselines across all the metrics. Furthermore, from Figure 1, we notice that PAGER accurately identifies the training data regimes (Green) as ID. As we traverse further from the training manifold, PAGER assigns low risk (Yellow) to unseen examples that are close to the training data. Notably, as we encounter samples that are significantly out-of-distribution, it consistently flags them as moderate/high risk. As an ablation, we also include the performance obtained by (a) using only the MNC ($\texttt{Score}_1$) and (b) using only uncertainties from PAGER, in order to demonstrate the importance of considering them jointly. While the MNC-only baseline can reasonably control FP, it is not able to reduce the FN. Since those scores are unnormalized, they behave differently across different data regimes. Hence, they are insufficient to accurately rank samples on their own. On the other hand, we observe that Anchor-UQ is a stronger baseline, even outperforming DEUP in many cases.

In addition to the fidelity metrics, computational efficiency is another important aspect of failure detectors. Hence, we provide the inference run-times for each of the methods, measured using a test set of 1000 samples on the 1D benchmarks with a single GPU. While DataSUITE (**29.8s**) involves training an autoencoder followed by conformalization, DEUP (**18.2s**) requires training an auxiliary risk estimator to evaluate risk. In comparison, computing $\texttt{Score}_1$ with PAGER is very efficient (**1.55s**) as it basically involves only forward passes with the anchored model. While $\texttt{Score}_2$ comes with an increased computational cost (40.9s), we find that it helps in resolving regimes of moderate and high risk better, and handling corruptions at test time (Figure 6).

Table 3. **Assessing the identified risk regimes for regression benchmarks (Tails) with dimensionality ranging between 2 and 32,000.** For every metric, we identify the first and second best approach across the different benchmarks.

Metrics	Method	Camel	Levy	Airfoil	NO2	Kinematics	Puma	Housing	Ailerons	DTI
FN↓	DEUP	10.53	7.34	11.28	13.76	14.39	16.82	2.11	18.37	19.23
	Data SUITE	3.84	9.21	11.02	12.16	17.59	22.38	17.89	17.58	20.06
	PAGER ($\texttt{Score}_1$)	0.0	4.56	1.94	3.65	8.02	8.78	1.05	9.59	6.67
	PAGER ($\texttt{Score}_2$)	0.25	4.82	2.48	3.25	7.18	10.38	2.32	9.59	7.13
FP↓	DEUP	9.53	7.35	10.82	9.11	13.02	14.67	8.77	12.01	17.34
	Data SUITE	3.83	6.38	9.15	9.75	24.0	26.67	19.3	12.0	14.07
	PAGER ($\texttt{Score}_1$)	0.42	1.68	2.85	4.27	6.33	13.33	3.51	0.80	9.09
	PAGER ($\texttt{Score}_2$)	1.68	2.52	4.29	6.18	6.18	12.23	4.26	0.38	8.36
C_{low}↓	DEUP	34.04	52.74	29.11	16.95	6.36	5.37	13.0	11.07	48.25
	Data SUITE	42.08	81.06	57.01	33.47	7.34	5.67	17.73	16.52	90.11
	PAGER ($\texttt{Score}_1$)	15.59	26.44	8.25	15.09	6.58	4.61	5.14	17.19	19.94
	PAGER ($\texttt{Score}_2$)	14.37	14.04	10.08	11.73	5.73	5.5	6.67	11.38	17.01
C_{high}↓	DEUP	23.69	20.75	14.56	27.34	6.83	2.63	5.69	7.25	39.94
	Data SUITE	17.49	27.32	18.09	31.58	10.08	6.41	5.15	4.97	64.48
	PAGER ($\texttt{Score}_1$)	7.5	17.93	15.19	12.08	7.14	2.46	5.07	2.31	13.35
	PAGER ($\texttt{Score}_2$)	6.7	15.18	16.64	10.68	7.09	2.81	4.05	2.43	11.06

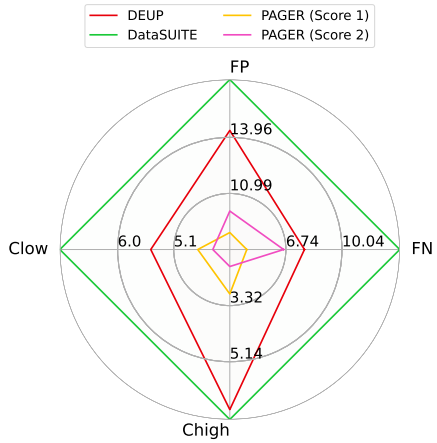


Figure 4. **PAGER can detect failures under complex distribution shifts effectively.** We assess PAGER on the Skillcraft dataset characterized by real-world shifts (change in league index), and find that it achieves reductions in all metrics over the baselines.

Table 4. **Predictive performance of anchoring.** The use of anchored training for failure characterization does not compromise on the performance (R^2 scores), regardless of whether the test data comes from observed or unobserved regimes.

Dataset	Training	R^2 (Observed)	R^2 (Unobserved)
CIFAR-10 (rotation angle)	Standard	0.92	0.77
	Anchoring	0.93	0.81
Chairs (yaw angle)	Standard	0.97	0.73
	Anchoring	0.97	0.75
Cells (count)	Standard	0.88	0.69
	Anchoring	0.89	0.72

5.2. Results on HD Regression Datasets

The observations from our 1D experiments persist even with higher dimensional regression benchmarks, thus evidencing the efficacy of PAGER. From Tables 2 and 3, we find that even in higher dimensions and more complex extrapolation scenarios (e.g., Gaps and Tails, as discussed in Section 4), PAGER is able to produce $> 50\%$ reduction in FN and FP metrics over the baselines. Furthermore, PAGER significantly reduces the amount of overlap (C_{low} and C_{high}) between the risk regimes. The baselines on the other hand produce higher confusion scores demonstrating their limitations in risk stratification. This observation persists even on the Skillcraft dataset containing real-world distribution shifts (Figure 4). Finally, despite the increased computational complexity, $\texttt{Score}_2$ leads to lower confusion scores compared to $\texttt{Score}_1$ while producing comparable FP and FN metrics.

5.3. Results on Imaging Benchmarks

Our analysis in Figure 5 reveals that PAGER achieves lower FN, FP, and confusion scores compared to the baselines, even when confronted with challenging extrapolation regimes in imaging datasets. This demonstrates the effectiveness of our approach in handling diverse modalities of data. Additionally, we provide sample images that were accurately identified as high risk by PAGER in Appendix A.4. Notably, these examples correspond to regimes that were not encountered during training.

Chair Angle Prediction

Observed $y \in [0-30, 60-90]$ Unobserved $y \in [30-60]$

GAPS

Method	FN	FP	C_{low}	C_{high}
DEUP	9.9	12.46	13.1	15.34
Ours (Score 1)	6.8	8.67	6.9	8.85
Ours (Score 2)	5.6	8	6.78	9.1

TAILS

Method	FN	FP	C_{low}	C_{high}
DEUP	3.1	12.96	26.6	20.9
Ours (Score 1)	1.6	9.33	19.8	11.2
Ours (Score 2)	2.4	9	18.63	7.72

Cell Count Estimation

Observed $y \in [0-50, 150-200]$ Unobserved $y \in [50-150]$

GAPS

Method	FN	FP	C_{low}	C_{high}
DEUP	18.88	13.39	8.83	11.54
Ours (Score 1)	10.40	9.33	3.28	5.34
Ours (Score 2)	12.03	10.33	3.04	4.42

Observed $y \in [50-150]$ Unobserved $y \in [0-50, 150-200]$

TAILS

Method	FN	FP	C_{low}	C_{high}
DEUP	4.25	15.49	10.33	9.98
Ours (Score 1)	4.82	11.36	5.01	6.23
Ours (Score 2)	4.82	10.01	4.86	3.68

CIFAR-10 Rotation Angle Prediction

Observed $y \in [0-45, 70-90]$ Unobserved $y \in [45-70]$

GAPS

Method	FN	FP	C_{low}	C_{high}
DEUP	14.90	15.22	18.81	27.50
Ours (Score 1)	3.34	7.86	3.28	5.34
Ours (Score 2)	3.83	9.14	2.85	3.19

Observed $y \in [25-70]$ Unobserved $y \in [0-25, 70-90]$

TAILS

Method	FN	FP	C_{low}	C_{high}
DEUP	19.24	20.04	29.37	47.15
Ours (Score 1)	6.71	8.11	11.73	4.06
Ours (Score 2)	5.05	10.38	6.58	2.09

Figure 5. **Efficacy of PAGER on Image Regression Benchmarks.** We can observe that in comparison to the state-of-the art baseline DEUP, PAGER effectively minimizes the FN, FP and confusion metrics even under challenging extrapolation scenarios. We find that PAGER can consistently flag samples from the unobserved regimes which corresponds to highly erroneous predictions.

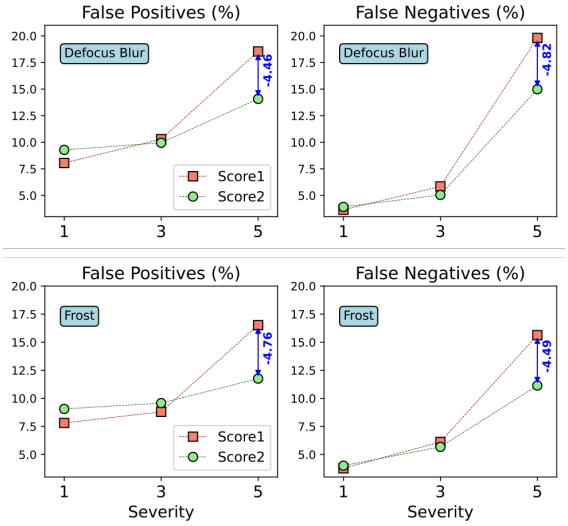


Figure 6. **Benefits of $\texttt{Score}_2$.** Under test time image corruptions such as defocus blur and frost, $\texttt{Score}_2$ can better resolve these risk regimes and reduce both FP and FN metrics over $\texttt{Score}_1$.

5.4. Analysis

In this section, we provide deeper empirical insights into the behavior of the proposed approach.

A. Do anchored models compromise performance? An important question that pertains the use of anchored models for failure characterization is whether these models compromise predictive performance. Anchored training is a general protocol applicable to any architecture and typically leads to improved generalization, particularly under distribution shifts. Conceptually, as showed in (Thiagarajan et al., 2022), centering a dataset using different constant inputs will lead to different solutions, due to inherent lack of shift invariance in neural tangent kernels induced by commonly adopted neural networks. Building upon this principle, we use different anchors for the same sample across different epochs with the goal of marginalizing out the effect of anchor choice at inference time. Through this process, anchoring implicitly enables the training process to explore a richer class of hypotheses, and often produces improved predictive performance when compared to standard model training. To demonstrate this, we computed the test performance (R^2 statistic) in both observed (range of y values exposed during training) and unobserved (range of y values unseen during training) regimes for the three image regression benchmarks. As shown in Table 4, anchoring performs competitively and sometimes even outperforms standard training.

B. When should one use $\texttt{Score}_2$ in PAGER? As shown above, $\texttt{Score}_2$ often demonstrates noteworthy improvements in confusion scores (C_{low} and C_{high}). This is valuable in scenarios where users need to flag samples with the highest risk, and ensure that high-risk samples are not misclassified as moderate risk. Another scenario where $\texttt{Score}_2$ is beneficial is when the test samples are drawn from a different distribution compared to training (referred to as covariate shifts). To demonstrate this, we repeated the CIFAR-10 rotation angle prediction experiment by applying natural image corruptions (defocus blur and frost) at varying severity levels (Figure 6). Interestingly, we observed significant improvements in both FP and FN scores with $\texttt{Score}_2$ as the severity increased. In summary, while $\texttt{Score}_1$ excels in scalability and is well-suited for online evaluation, $\texttt{Score}_2$ effectively addresses challenging testing scenarios.

C. Implementing PAGER with an anchored regression head. Regarding the application of PAGER to models

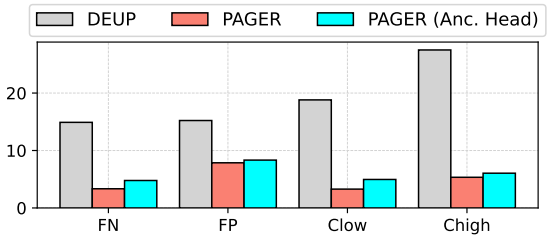


Figure 7. **PAGER can be implemented for a model trained without anchoring.** By training only a regression head via anchoring, one can implement PAGER with a pre-trained feature extractor backbone. Using the CIFAR-10 rotation angle prediction experiment, we show that this variant produces improvements over the DEUP baseline, similar to PAGER with a fully anchored model.

Table 5. **Impact of post-hoc calibration on PAGER.** When additional calibration data becomes available, PAGER can leverage it to both recalibrate the uncertainty estimates as well as improve the non-conformity scores. This leads to consistently superior metrics over the PAGER variant without additional calibration data.

Dataset	Method	Calibration?	FP↓	FN↓	C_{low}↓	C_{high}↓
DTI	DEUP	No	17.34	19.23	48.25	39.94
		Yes	13.08	12.95	33.10	19.26
	PAGER	No	9.09	6.67	19.94	13.35
		Yes	4.27	3.88	17.55	10.76
CIFAR-10	DEUP	No	15.22	14.90	18.81	27.50
		Yes	6.08	4.64	6.36	7.75
	PAGER	No	7.86	3.34	3.28	5.34
		Yes	4.81	2.59	1.70	4.48

trained without anchoring, we first emphasize that anchoring does not necessitate any adjustments to the optimizer, loss function, or training protocols. However, in situations involving pre-trained models, it is feasible to train solely an anchoring-based regression head attached to an existing feature extractor. Optionally, one may fine-tune the feature extractor concurrently with the anchored regression head in an end-to-end manner, adhering to standard practices. As an illustration, in the experiment on CIFAR-10 rotation angle prediction under the Gaps setting with $\texttt{Score}_1$, we considered a variant, where we trained an anchored head while keeping the feature extractor frozen. Note, the feature extractor was obtained through standard training on the same dataset. Our findings in Figure 7 reveal that even with this approach, the performance is comparable to PAGER implemented with a fully anchored model.

D. Does post-hoc calibration help PAGER? In real-world applications, obtaining access to calibration data representing the distribution shifts is not always practical. Hence, one of our objectives was to ensure that that PAGER can still meaningfully organize samples into different risk regimes, even without post-hoc calibration. However, it is common practice to adopt post-hoc calibration, when additional data becomes available. In such scenarios, PAGER can leverage the data to (i) recalibrate the uncertainty estimates from forward anchoring for guaranteed coverage, and (ii) improve the quality of the non-conformity score estimates. To demonstrate this, we performed an additional experiment, where we calibrated the uncertainty estimates (90% coverage), and reimplemented the non-conformity score computation by obtaining the max discrepancy over the union of training ($\mathcal{D}$) and calibration sets ($\mathcal{D}_c$). From Table 5, we find that the performance of both the baseline and PAGER improve by including additional calibration data, and more importantly, the benefits of PAGER persist.

Table 6. **Ablation study.** Using only uncertainties or the proposed non-conformity scores leads to sub-par performance in risk characterization on the CIFAR-10 benchmark.

Method	FP↓	FN↓	C_{low}↓	C_{high}↓
UQ-only	12.54	13.45	14.75	9.15
MNC-only	13.08	13.24	12.91	11.38
PAGER	5.05	10.38	6.58	2.09

E. Ablation study. In order demonstrate the value of combining uncertainty and manifold non-conformity scores in PAGER, we performed an ablation study on the CIFAR-10 rotation angle benchmark. In theory, all extreme rotations should be picked by large uncertainties – however, in practice, they tend to produce both false positives and false negatives, which can be attributed to insufficiency of epistemic uncertainties (see Figure 1) as well as their poor calibration under distribution shifts. On the other hand, while non-conformity can identify discrepancies arising due to feature inconsistency, it has the inherent challenge that it only measures the relative change in the target space or distances in the input space. Since these scores are unnormalized, they can vary vastly across different uncertainty regimes. Consequently, as we illustrate in Table 6, combining both scores leads to significantly improved performance.

6. Conclusions

In this paper, we proposed PAGER, a framework for failure characterization in deep regression models. It leverages the principle of anchoring to integrate epistemic uncertainties and novel non-conformity scores, enabling the organization of samples into different risk regimes and facilitating a comprehensive analysis of model errors. We identify two key impacts. First, PAGER can enhance the safety of AI model deployment by preemptively detecting failure cases in real-world applications. This can prevent costly errors and mitigate risks associated with inaccurate predictions. Second, PAGER contributes to advancing research in failure characterization for deep regression.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344. Supported by the LDRD Program under project 22-SI-004. LLNL-CONF-850978.

References

Ailerons datsets. https://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html. Accessed: 2023-05-11.

Virtual library of simulation experiments. https://www.sfu.ca/~ssurjano/index.html. Accessed: 2023-05-01.

Boston housing. https://scikit-learn.org/1.0/modules/generated/sklearn.datasets.load_boston.html. Accessed: 2023-05-11.

Delve datasets. https://www.cs.toronto.edu/~delve/data/datasets.html. Accessed: 2023-05-11.

Amini, A., Schwarting, W., Soleimany, A., and Rus, D. Deep evidential regression. *Advances in Neural Information Processing Systems*, 33:14927–14937, 2020.

Baek, C., Jiang, Y., Raghunathan, A., and Kolter, J. Z. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. *Advances in Neural Information Processing Systems*, 35:19274–19289, 2022.

Bishop, C. M. and Nasrabadi, N. M. *Pattern Recognition and Machine Learning*. *J. Electronic Imaging*, 16(4): 049901, 2007.

Chen, M., Goel, K., Sohoni, N. S., Poms, F., Fatahalian, K., and Ré, C. Mandoline: Model evaluation under distribution shift. In *International Conference on Machine Learning*, pp. 1617–1629. PMLR, 2021.

Deng, W. and Zheng, L. Are labels always necessary for classifier accuracy evaluation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15069–15078, 2021.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, pp. 1–77, 2023.

Guillory, D., Shankar, V., Ebrahimi, S., Darrell, T., and Schmidt, L. Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1134–1144, 2021.

He, B., Lakshminarayanan, B., and Teh, Y. W. Bayesian deep ensembles via the neural tangent kernel. *Advances in Neural Information Processing Systems*, 33:1010–1022, 2020.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Jiang, Y., Krishnan, D., Mobahi, H., and Bengio, S. Predicting the generalization gap in deep networks with margin distributions. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJlQfnCqKX.

Jiang, Y., Nagarajan, V., Baek, C., and Kolter, J. Z. Assessing generalization of SGD via disagreement. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=WvOGCEAQhxl.

Lahlou, S., Jain, M., Nekoei, H., Butoi, V. I., Bertin, P., Rector-Brooks, J., Korablyov, M., and Bengio, Y. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=eGLdVRvvfQ.

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

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Liu, J. Z., Lin, Z., Padhy, S., Tran, D., Bedrax-Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *arXiv preprint arXiv:2006.10108*, 2020.

Luo, R., Sun, L., Xia, Y., Qin, T., Zhang, S., Poon, H., and Liu, T.-Y. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6), 2022.

Narayanaswamy, V., Anirudh, R., Kim, I., Mubarka, Y., Spanias, A., and Thiagarajan, J. J. Predicting the generalization gap in deep models using anchoring. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4393–4397. IEEE, 2022.

Netanyahu, A., Gupta, A., Simchowitz, M., Zhang, K., and Agrawal, P. Learning to extrapolate: A transductive approach. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lid14UkLPd4.

Ng, N., Cho, K., Hulkund, N., and Ghassemi, M. Predicting out-of-domain generalization with local manifold smoothness. *arXiv preprint arXiv:2207.02093*, 2022.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Seedat, N., Crabbé, J., and van der Schaar, M. Data-SUITE: Data-centric identification of in-distribution incongruous examples. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 19467–19496, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/seedat22a.html.

Teng, J., Wen, C., Zhang, D., Bengio, Y., Gao, Y., and Yuan, Y. Predictive inference with feature conformal prediction. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0uRm1YmFTu.

Thiagarajan, J. J., Anirudh, R., Narayanaswamy, V., and timo Bremer, P. Single model uncertainty estimation via stochastic data centering. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=j0J9upqN5va.

Tibshirani, R. J., Foygel Barber, R., Candes, E., and Ramdas, A. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pp. 9690–9700. PMLR, 2020.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Yang, J., Zhou, K., Li, Y., and Liu, Z. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

Yao, H., Wang, Y., Zhang, L., Zou, J. Y., and Finn, C. C-mixup: Improving generalization in regression. *Advances in Neural Information Processing Systems*, 35: 3361–3376, 2022.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

A. Appendix

A.1. Detailed Description of Anchoring in PAGER

PAGER expands on the recent successes in anchoring (Thiagarajan et al., 2022; Netanyahu et al., 2023) by building upon the Δ-UQ methodology introduced in (Thiagarajan et al., 2022). This methodology is used to estimate prediction uncertainties, which play a vital role in characterizing model risk regimes, as depicted in Figure 2 of the main paper. With that context, we now provide a concise overview of $\Delta-$UQ, its training and uncertainty estimation.

Overview: $\Delta-$UQ, short for $\Delta-$Uncertainty Quantification, is a highly efficient strategy for estimating predictive uncertainties that leverages anchoring. It belongs to the category of methods that estimate uncertainties using a single model (Van Amersfoort et al., 2020; Liu et al., 2020). $\Delta-$UQ has been demonstrated to be an improved and scalable alternative to Deep Ensembles (Lakshminarayanan et al., 2017), eliminating the need to train multiple independent models for estimating uncertainties. The core idea behind $\Delta-$UQ is based on the observation that the injection of constant biases (anchors) to the input dataset produces different model predictions as a function of the bias. To that end, models trained using the same dataset but shifted by respective biases generates diverse predictions. This phenomenon arises from the fact that the neural tangent kernel (NTK)(Jacot et al., 2018) induced in deep models lacks invariance to input data shifts (Bishop & Nasrabadi, 2007). Consequently, the variance among these models *a.k.a anchored ensembles* serves as a strong indicator of predictive uncertainty. Based on this observation, $\Delta-$UQ follows a simple strategy to consolidate the anchored ensembles into a single model training, where the input is reparameterized as an anchored tuple, as described in Section 2 of the main paper. It is important to note that $\Delta-$UQ performs anchoring in the input space for both vector-valued and image data.

Training: In this phase, for every training pair $\{\mathrm{x}, \mathrm{y}\}$ drawn from the dataset $\mathcal{D}$, a random anchor r_k is selected from the same training dataset. Both the input x and the anchor r_k are transformed into a tuple given as $[\mathrm{r}_k, \mathrm{x} - \mathrm{r}_k]$. Importantly, this reparameterization does not alter the original predictive task, but instead of using only x, the tuple $[\mathrm{r}_k, \mathrm{x} - \mathrm{r}_k]$ is mapped to the target y. For vector-valued data, $\Delta-$UQ constructs the tuples by concatenating the anchor r_k and the residual along the dimension axis. In the case of images, the tuples are created by appending along the channel axis, resulting in a $6-$channel tensor for each 3-channel image. These tuples are organized into batches and used to train the models. Throughout the training process, in expectation, every sample x is anchored with every other sample in the dataset. The goal here is that the predictions for every x should remain consistent regardless of the chosen anchor. The training objective is given by:

$$\theta^* = \arg\min_{\theta} \quad \mathcal{L}(\mathrm{y}, F_\theta([\mathrm{r}_k, \mathrm{x} - \mathrm{r}_k]), \tag{4}$$

where $\mathcal{L}(.)$ is a loss function such as MAE or MSE. In effect, the $\Delta-$UQ training enforces that for every input sample x, $F_\theta([\mathrm{r}_1, \mathrm{x} - \mathrm{r}_1]) = F_\theta([\mathrm{r}_2, \mathrm{x} - \mathrm{r}_2]) = \cdots = F_\theta([\mathrm{r}_k, \mathrm{x} - \mathrm{r}_k])$, where F_θ is the underlying model that operates on the tuple $([\mathrm{r}_k, \mathrm{x} - \mathrm{r}_k])$ to predict y.

Uncertainty Estimation: During the inference phase, using the trained model with weights θ^*, we compute the prediction y_t for any test sample x_t. This is performed by averaging the predictions across K randomly selected anchors drawn from the training dataset. The standard deviation of these predictions is then used as the estimate for predictive uncertainty. The equations for calculating the mean prediction and uncertainty around a sample can be found in Equation 1 of the main paper.

A.2. Algorithm Listings for PAGER

Algorithms 1,2 and 3 provide the details for estimating predictive uncertainty, non-conformity scores - $\texttt{Score}_1$ and $\texttt{Score}_2$ respectively in PAGER.

A.3. Description of our Training Protocols

In the case of Cell Count and Chair Angle benchmarks, we train an anchored $40-2$ WideResnet model. The training is performed with a batch size of 128 for 100 epochs. We utilize the ADAM optimizer with momentum parameters of $(0.9, 0.999)$ and a fixed learning rate of $1e-4$. To train the anchored auto-encoder for computing $\texttt{Score}_2$, we employ a convolutional architecture with an encoder-decoder structure. The encoder consists of two convolutional layers with kernel sizes of $(3, 3)$ and appropriate padding, as well as MaxPooling operations. The decoder comprises two transposed convolutional layers with stride 2 to reconstruct the input images. We train the anchored auto-encoder using a batch size of 128 for 100 epochs. The ADAM optimizer with momentum parameters $(0.9, 0.999)$, and a fixed learning rate of $1e-3$, is used for training. As mentioned in the main paper, for the case of CIFAR-10, we train a ResNet-34 model with the

Algorithm 1 PAGER: Predictive Uncertainty Estimation

Input: Input test samples $\{\mathrm{x}_i^t\}_{i=1}^N$, Pre-trained anchored model F_{θ^*}, Anchors $\{\mathrm{r}_k\}_{k=1}^K$ drawn from the training dataset $\mathcal{D}$
Output: Predictive Uncertainties (`Unc`) for $\{\mathrm{x}_i^t\}_{i=1}^N$
Initialize: $\texttt{Unc} = \texttt{list}()$
for i in 1 to N **do**
  $\mu(y_i^t|\mathrm{x}_i^t) = \frac{1}{K}\sum_{k=1}^K F_{\theta^*}([\mathrm{r}_k, \mathrm{x}_i^t - \mathrm{r}_k])$;
  $\sigma(y_i^t|\mathrm{x}_i^t) = \sqrt{\frac{1}{K-1}\sum_{k=1}^K (F_{\theta^*}([\mathrm{r}_k, \mathrm{x}_i^t - \mathrm{r}_k]) - \mu(y_i^t|\mathrm{x}_i^t))^2}$;
  $\texttt{Unc[i]} = \sigma(y_i^t|\mathrm{x}_i^t)$
end for
return: `Unc`

Algorithm 2 PAGER: $\texttt{Score}_1$ Computation

Input: Input test samples $\{\mathrm{x}_i^t\}_{i=1}^N$, Pre-trained anchored model F_{θ^*}, Train data subset $\{\mathrm{r}_k, \mathrm{y}_k\}_{k=1}^K$
Output: $\texttt{Score}_1$ for $\{\mathrm{x}_i^t\}_{i=1}^N$
Initialize: $\texttt{Score}_1 = \texttt{list}()$
for i in 1 to N **do**
  $\mathtt{s} = \max_k \left\| \mathrm{y}_k - F_{\theta^*}([\mathrm{x}_i^t, \mathrm{r}_k - \mathrm{x}_i^t]) \right\|_1 \quad \forall k \in \{1 \cdots K\}$;
  $\texttt{Score}_1\texttt{[i]} = \mathtt{s}$
end for
return: $\texttt{Score}_1$

Algorithm 3 PAGER: $\texttt{Score}_2$ Computation

Input: Input test samples $\{\mathrm{x}_i^t\}_{i=1}^N$, Pre-trained anchored model F_{θ^*}, Pre-trained anchored auto-encoder A, Train data subset $\{\mathrm{r}_k, \mathrm{y}_k\}_{k=1}^K$, Learning rate η, Weighing Factor λ, No. of iterations T
Output: $\texttt{Score}_2$ for $\{\mathrm{x}_i^t\}_{i=1}^N$
Initialize: $\texttt{Score}_2 = \texttt{list}()$
for i in 1 to N **do**
  Initialize: $\bar{\mathrm{x}} \leftarrow \mathrm{x}_i^t$
  for $iter$ in 1 to T **do**
    Compute $\mathcal{R}(\bar{\mathrm{x}}) = \left\| \bar{\mathrm{x}} - A([\mathrm{x}_i^t, \bar{\mathrm{x}} - \mathrm{x}_i^t]) \right\|_2 + \left\| \mathrm{x}_i^t - A([\bar{\mathrm{x}}, \mathrm{x}_i^t - \bar{\mathrm{x}}]) \right\|_2$.
    Compute $L = \frac{1}{K}\sum_k \|\mathrm{y}_k - F_{\theta^*}([\bar{\mathrm{x}}, \mathrm{r}_k - \bar{\mathrm{x}}])\|_1 + \lambda\mathcal{R}(\bar{\mathrm{x}})$
    Update $\bar{\mathrm{x}} \leftarrow \bar{\mathrm{x}} - \eta\nabla_{\bar{\mathrm{x}}} L$
  end for
  $\texttt{Score}_2\texttt{[i]} = \|\mathrm{x}_i^t - \bar{\mathrm{x}}\|_2$
end for
return: $\texttt{Score}_2$

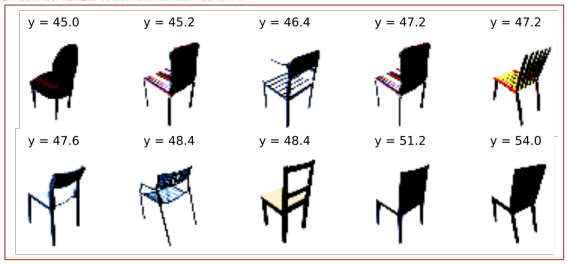
standard training configurations. For the other regression benchmarks, we used a standard MLP with 5 hidden layers, ReLU activation and batchnorm. They were all trained for 5000 epochs with learning rate $5e-5$ and ADAM optimizer.

A.4. Additional Results

Image regression experiment. For the cell count and chair angle prediction benchmarks from the main paper, we provide examples of high-risk sample as detected by PAGER. Please refer to Figure 8 for the examples. Notably, these samples correspond to regimes that were not encountered during training.

Observed $y \in [0-30, 60-90]$ Unobserved $y \in [30-60]$ Predicted Failure Cases

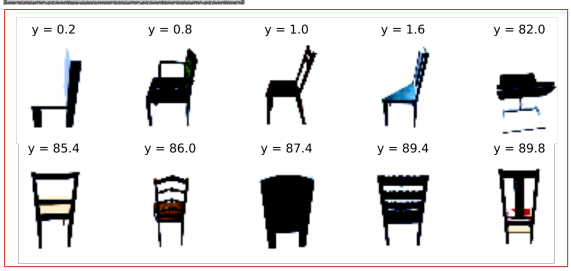
Method	FN	FP	C_{low}	C_{high}
DEUP	9.9	12.46	13.1	15.34
Ours (Score 1)	6.8	8.67	6.9	8.85
Ours (Score 2)	5.6	8	6.78	9.1



(a) Chair Angles (Gap)

Observed $y \in [15-75]$ Unobserved $y \in [0-15, 75-90]$ Predicted Failure Cases

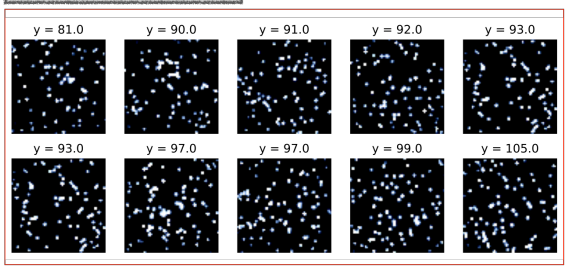
Method	FN	FP	C_{low}	C_{high}
DEUP	3.1	12.96	26.6	20.9
Ours (Score 1)	1.6	9.33	19.8	11.2
Ours (Score 2)	2.4	9	18.63	7.72



(b) Chair Angles (Tails)

Observed $y \in [0-50, 150-200]$ Unobserved $y \in [50-150]$ Predicted Failure Cases

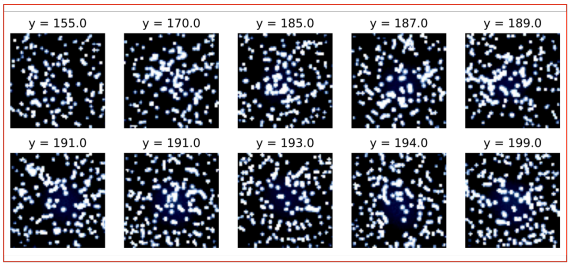
Method	FN	FP	C_{low}	C_{high}
DEUP	18.88	13.39	8.83	11.54
Ours (Score 1)	10.4	9.33	3.28	5.34
Ours (Score 2)	12	10.33	3.04	4.42



(c) Cells Count (Gap)

Observed $y \in [50-150]$ Unobserved $y \in [0-50, 150-200]$ Predicted Failure Cases

Method	FN	FP	C_{low}	C_{high}
DEUP	4.2	15.49	10.33	9.98
Ours (Score 1)	4.8	11.3	5.01	6.23
Ours (Score 2)	4.8	10	4.86	3.68



(d) Cells Count (Tails)

Figure 8. **Efficacy of PAGER on Image Regression Benchmarks.** We can observe that in comparison to the state-of-the art baseline DEUP, PAGER effectively minimizes the FN, FP and confusion metrics even under challenging extrapolation scenarios. We find that PAGER can consistently flag samples from the unobserved regimes which corresponds to highly erroneous predictions.