# CALI: Failure Prediction via Cramer-Wold Latent Inference and KDE

Nathan Nowakowski, Diana Nurbakova, Előd Egyed-Zsigmond, Sylvie Calabretto

# CALI: Failure Prediction via Cramer-Wold Latent Inference and KDE

Nathan Nowakowski[1] (✉), Diana Nurbakova[1], Előd Egyed-Zsigmond[1], and Sylvie Calabretto[1]

INSA Lyon, CNRS, Universite Claude Bernard Lyon 1, LIRIS, UMR5205, 69621 Villeurbanne, France {nathan.nowakowski,diana.nurbakova, elod.egyed-zsigmond,sylvie.calabretto}@insa-lyon.fr

**Abstract.** Reliable confidence estimation is essential for deploying deep neural networks in safety-critical applications. A major issue in failure prediction is the lack of score polarisation; preventing a clear separation between success and failure distributions. While this phenomenon is often noted, it is rarely analysed in depth. We introduce CALI, a framework that uses class-specific Cramer-Wold Auto-Encoders with a novel learning strategy, and Kernel Density Estimation to model the resulting discriminative latent manifold, enabling robust confidence assessment in classification. We utilise the Wasserstein distance to formally quantify score distribution separation. Furthermore, we introduce a new metric, the Average Risk Increase (ARI), which measures the loss in failure prediction precision when calibrated thresholds are subjected to variations. ARI could serve as a key indicator for industrial reliability in evolving environments. Experiments across diverse architectures and datasets show that our method reaches state-of-the-art performance in FPR@95TPR and AUPR-error. Our approach consistently achieves the best results in Wasserstein distance and ARI, demonstrating superior score separability and practical robustness for real-world deployment.

**Keywords:** Failure Prediction · Discriminative Representation Learning · Score Separation.

## 1 Introduction

The rapid integration of Artificial Intelligence (AI) systems into critical areas such as medical diagnosis, autonomous vehicles, or defence [41, 8] necessitates reliability that goes beyond mere accuracy. Users must be able to trust each and every decision made by the system, even in uncertain or adverse conditions [25]. However, these complex systems are plagued by a critical vulnerability: their lack of transparency and propensity for overconfidence pose a significant systemic risk [12]. This is why transparency has become a clear legislative requirement, as shown by the EU AI Act [9], which imposes human control mechanisms for high-risk systems, requiring machines to account for their own errors.

However, for such human control to be operational, the system must be able to accurately characterise the origin of its doubts. This need for discernment

leads us to rigorously distinguish between out-of-distribution (OOD) detection and failure prediction (FP) [17]. While OOD focuses on identifying unknown data that the model has never encountered during training, FP acts as an internal moderator for known domain data. The challenge, illustrated in Figure 1, is more subtle: it is not about detecting a foreign intrusion, but rather identifying a judgement error on familiar data.

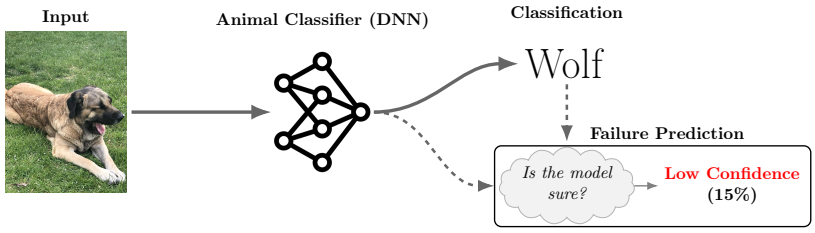


**Fig. 1.** Conceptual Overview of the Failure Prediction Paradigm

While AUROC remains a standard metric for AI evaluation, complementary metrics such as AUPR-Error, FPR@95TPR, AURC, and e-AURC have been proposed to address context-specific error costs [14, 4, 11, 46]. However, these metrics overlook threshold stability: when confidence distributions of successes and failures overlap, identifying a robust decision threshold becomes impractical. We argue that operational confidence scores must be polarised, minimising risk while ensuring clear distributional separation for resilient error rejection.

To address this, we explicitly model the regions of correct and misclassified examples per class, drawing inspiration from human intuition: confidence grows with repeated positive outcomes, and vice-versa. Our framework provides interpretable confidence scores by ranking predictions based on their similarity to prior outcomes, analogous to human decision-making. We achieve this by leveraging Kernel Density Estimation (KDE) [3] to quantify the statistical proximity of new samples to historical distributions of *good* and *bad* decisions, identifying a safety zone for decision-making.

Our approach is flexible, accommodating internal representations from any layer of a pre-trained DNN classifier. However, high-dimensional neural activations introduce challenges: the curse of dimensionality and structural noise that obscure success-failure boundaries. We mitigate these issues by compressing the data into a discriminative latent space using a modified Cramer-Wold Auto-Encoder (CWAE) [20]. This not only reduces dimensionality but also preserves data topology while disentangling correct and incorrect classifications, enabling more accurate KDE-based density estimation.

Since our framework awaits one or a combination of input layers, it raises the concerns of the optimal source of uncertainty signals within DNN. While some approaches suggest combining multiple layers to leverage the multi-scale richness of representations [34, 24], others, such as Trust Score [18] or ConfidNet [4], rely

only on the penultimate layer. We will therefore conduct an analysis to determine which strategy best captures the *error signature* within our latent manifold.

Our main contributions, validated by extensive experiments, are as follows:

- A dimensionality reduction method preserving the geometric structure necessary for score separation.
- The introduction of distribution polarisation as a performance metric, surpassing the ambiguity of current scores.
- A directly interpretable score, enabling effective human control and also aligning with AI certification requirements.

## 2 Definitions and Notation

Failure prediction, also known as misclassification detection [14] or ordinal ranking [28], aims to identify incorrect predictions by comparing their confidence level to that of correct predictions. If misclassified samples consistently have lower confidence than correctly classified ones, it is possible to predict model errors at inference time. Formally, an optimal ordinal ranking model must satisfy for every two samples $(x_i, y_i)$ and $(x_j, y_j)$:

$$\kappa(x_i, y_i) > \kappa(x_j, y_j) \quad \forall (x_i, x_j) \text{ s.t. } \hat{y}_i = y_i \text{ and } \hat{y}_j \neq y_j \tag{1}$$

where:

- $x_i \in \mathcal{X}$ denotes an input representations (e.g., raw sample or embedding),
- $y_i \in \mathcal{Y}$ is the corresponding true label,
- $\hat{y}_i$ is the model's predicted label for $x_i$,
- $\kappa(x, y)$ quantifies the confidence in the prediction $(x, y)$.

Then, with a predefined threshold $\tau$, the users can reject the incorrect classification results based on the following decision function $g$:

$$g(x, \tau) = \begin{cases} \text{accept} & \text{if } \kappa(x, \hat{y}) \geq \tau \\ \text{reject} & \text{if } \kappa(x, \hat{y}) < \tau \end{cases} \tag{2}$$

## 3 Related Work

Generating reliable confidence scores is challenging as DNNs are notoriously overconfident, assigning high probabilities to incorrect predictions [12]. This issue is particularly pronounced with the standard baseline: Maximum Softmax Probability (MSP) [14]. To mitigate this, research has branched into **post-hoc scoring** and **confidence learning**.

Alternatives like the Energy Score leverage unnormalised logits [26], while Trust Score [18] utilises topological distances in the feature space. However, the latter's efficacy is often limited by the curse of dimensionality in high-dimensional neural embeddings.

Frameworks rooted in Bayesian inference, such as MC-Dropout [10] and Deep Ensembles [22], quantify uncertainty through predictive entropy. Despite their robustness, their high computational footprint, requiring multiple forward passes or memory-intensive model sets, often precludes deployment in real-time [27].

The confidence learning paradigm consists in learning confidence scores. Specialised architectures like ConfidNet [4] utilise an auxiliary head to regress true class probabilities, though they depend on the head's ability to capture failure modes missed by the backbone. To reduce overhead, FMFP [46] integrates confidence into the training objective via flat minima seeking. However, this induces a trade-off between classification accuracy and failure detection performance. Finally, Outlier Exposure techniques like OpenMix [47] train on non-target samples to improve uncertainty boundaries, but their efficacy is sensitive to the overlap between outliers and in-distribution failure modes.

To bridge the gap between complex confidence-learning models and computationally intensive Bayesian sampling, we propose *CALI* (CrAmer-wold Latent Inference and KDE). CALI treats FP as a constrained density estimation problem within a structured latent manifold. Unlike Mahalanobis-based methods that assume Gaussianity [24], CALI utilises a tailored CWAE to project hierarchical representations into a low-dimensional space ($d \leq 10$) specifically optimised to disentangle success and failure distributions. By applying KDEs on this manifold, we achieve robust failure detection with significantly lower inference overhead than Bayesian techniques and reduced training complexity compared to auxiliary networks like ConfidNet. Finally, by focusing strictly on misclassification detection, our approach eliminates the requirement for outlier exposure.

***Scope Boundaries.*** While our work focuses on predictive trust, it is situated at the intersection of several neighbouring paradigms. OOD detection aims to identify samples from unseen classes ; in contrast in FP we identify misclassifications on in-distribution (ID) data [24, 43]. Furthermore, Conformal Prediction provides model-agnostic prediction sets with guaranteed coverage for a chosen confidence level, offering formal guarantees but answering *which labels are plausible?* rather than giving a confidence score to the top label [36, 1]. Similarly, eXplainable AI (XAI) provide insights into model decision-making, but do not certify per-instance correctness and can inadvertently increase user overconfidence if explanations are plausible despite incorrect predictions [35, 39].

***What about Calibration?*** While uncertainty estimation often focuses on probability calibration, misclassification detection fundamentally relies on ordinal ranking rather than probabilistic interpretation. Enforcing calibration can degrade the separability between correct and incorrect predictions [46]. Consequently, our framework prioritises maximising the margin between these two distributions rather than aligning scores with frequentist accuracies.

# 4 The Proposed Framework - CALI

## 4.1 The Fundamental Idea: Localisation of Characteristics in Space

Our approach to quantifying confidence is based on a simple but powerful intuition: if the representation of an instance for a predicted class $c$ is located in a region with a high density of correctly classified examples for that class, this indicates high confidence. Conversely, if it lies in a region with a high density of misclassified examples, this suggests low confidence. The goal is therefore to model these underlying distributions to evaluate the typicality of a new example relative to the model's past behaviour. To do this, we construct two density estimators for each predicted class $c$:

$\widehat{f}_{\text{good},c}(x)$ : estimated density on the correctly classified examples of class $c$

$\widehat{f}_{\text{bad},c}(x)$ : estimated density on the incorrectly classified examples of class $c$

While parametric methods like Gaussian distributions or Gaussian Mixture Models (GMMs) are common [24, 43], they present challenges when applied to FP tasks. First, FP requires a discriminative comparison between two distinct density manifolds, correct and incorrect, rather than a one-class Gaussian membership test [5]. Second, strict parametric assumptions fail to capture the topological complexity of latent spaces, where misclassified samples often form fragmented, non-Gaussian clusters [29].

In contrast, KDE offers a non-parametric solution that converges to any density shape without prior knowledge of the distribution's form [3]. To the best of our knowledge, despite being a cornerstone of statistics [37, 3], KDE has not yet been applied specifically to FP. We therefore adopt KDE to model the complex success-failure boundaries within our latent space.

## 4.2 Bayesian Framing

To formalise the computation of our confidence score, we adopt a Bayesian approach. Although our ultimate goal for our failure detection system is to establish an ordinal ranking $\kappa$, we conceptualise our mathematics in a probabilistic domain to ensure mathematical coherence.

Let $c$ denote the model's predicted class for an input $x$, where $x$ corresponds to any layer of the representation space of the DNN. We define $C_{\text{good}}$ (resp. $C_{\text{bad}}$) as the event that the prediction for class $c$ is correct (resp. incorrect). Theoretically, the ideal Bayesian confidence score for classifying samples would be the posterior probability of correctness:

$$P(C_{\text{good}} \mid x, c) = \frac{f_{\text{good},c}(x)\, P(C_{\text{good}} \mid c)}{f_{\text{good},c}(x)\, P(C_{\text{good}} \mid c) + f_{\text{bad},c}(x)\, P(C_{\text{bad}} \mid c)} \tag{3}$$

where:

- $f_{\text{good},c}(x)$ and $f_{\text{bad},c}(x)$ are the class-conditional densities of $x$ for correct and incorrect predictions,

- $P(C_{\text{good}} \mid c)$ is the prior probability that a prediction for class $c$ is correct (i.e., the model's class-conditional accuracy for class $c$),
- $P(C_{\text{bad}} \mid c) = 1 - P(C_{\text{good}} \mid c)$.

**From Posterior Probabilities to Ordinal Ranking** While the Bayesian formulation provides a solid probabilistic foundation, directly estimating $P(C_{\text{good}} \mid c)$ in real-world scenarios is often counterproductive for FP. The extreme scarcity of errors ($P(C_{\text{bad}} \mid c) \ll P(C_{\text{good}} \mid c)$) tends to saturate the posterior scores toward one [4].

To bypass this imbalance, we deliberately shift from absolute probability estimation to relative density comparison. By adopting the established practice of assuming equal priors ($P(C_{\text{bad}} \mid c) = P(C_{\text{good}} \mid c) = 0.5$) [23, 15], we decouple the confidence score from the global accuracy of the model and refocus it on the local distinguishability of the sample. This transformation yields a more expressive ordinal ranking function:

$$\kappa(x, c) = \frac{\widehat{f}_{\text{good},c}(x)}{\widehat{f}_{\text{good},c}(x) + \widehat{f}_{\text{bad},c}(x)} \tag{4}$$

This formulation is made possible by estimating the underlying densities using KDE. Specifically, our approach involves fitting class-conditional KDEs on the DNN representation of training examples. For each predicted class $c$, we construct two separate KDEs: one using correctly classified examples to estimate $f_{\text{good},c}$, and another using misclassified examples to estimate $f_{\text{bad},c}$.

### 4.3 Dimensionality Reduction: Beyond Linear Projections

Following hierarchical approaches like Mahalanobis-based or DkNN [24, 34], we leverage internal representations to capture the model's reasoning process across various abstraction stages. Much like evaluating a student's logic rather than just a final answer, examining intermediate features allows us to detect inconsistencies that emerge long before the output layer. However, applying KDE directly to these high-dimensional embeddings (e.g., $d \geq 1024$ for DenseNet blocks) triggers the curse of dimensionality, where density metrics lose discriminative power. While techniques such as PCA or Gaussian approximations are common [18, 43], they fail to preserve non-linear topological structures and tend to over-smooth the fragmented regions characteristic of model failures [40, 29].

To address this, we introduce a compression scheme utilising a Cramer-Wold Auto-Encoder (CWAE) [20]. Beyond minimising reconstruction error, our CWAE serves as a supervised manifold shaping tool: it regularises the latent space (e.g., $d = 10$) to preserve discriminative features of model failure. By actively separating the manifolds of correct and incorrect predictions, this geometric transformation, illustrated in Figure 2, ensures that the subsequent KDE remains numerically stable and highly discriminative.

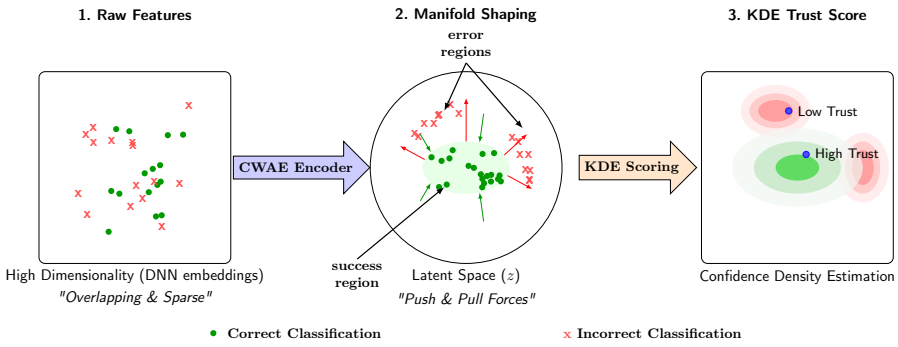


**Fig. 2.** Manifold Shaping Per Class for KDE Scoring

### 4.4 Latent Representation Learning via CWAE

To map high-dimensional DNN embeddings $\mathcal{F}$ into a compact latent space $\mathcal{Z} \in \mathbb{R}^d$, we adopt the Auto-Encoder (AE) framework [2]. By encoding activations $x \in \mathcal{F}$ (sampled from $P_x$) into lower-dimensional codes $z \in \mathcal{Z}$ before decoding them back, AEs learn to compress information through a reconstruction loss. This loss explicitly penalises discrepancies between the original input and its reconstructed output, ensuring the latent space preserves essential structural features. Concretely, we use an encoder $G_\phi : \mathcal{F} \to \mathcal{Z}$ and a decoder $G_\theta : \mathcal{Z} \to \mathcal{F}$, and minimise:

$$\mathcal{L}_{rec} = \mathbb{E}_{x \sim P_x}[\|x - G_\theta(G_\phi(x))\|^2] \tag{5}$$

This non-linear bottleneck yields a compact representation, enabling feasible density estimation while preserving critical structural information.

For effective FP, the latent space must exhibit a **discriminative topology** where successful and failed classifications form distinct, non-overlapping distributions. We identify the CWAE as the optimal framework for this structured manifold. While Variational Auto-Encoders (VAEs) are a common alternative, they are ill-suited for our geometric requirements; VAEs impose restrictive, per-sample KL-divergence penalties that introduce stochastic blurring and over-smoothing latent structures and merging success-failure boundaries [6]. In contrast, CWAEs preserve fine-grained geometric details and have been shown to outperform VAEs in capturing complex latent topologies [20]. We further enhance it by decoupling the learning signals based on the backbone's correctness, ensuring the territory of success is statistically predictable while the fringe of failure is geometrically isolated:

***Prior-Matching for Success (Correct Samples)*** For correctly classified samples ($\hat{y} = y$), we align the aggregated posterior with a standard Gaussian prior $P_Z = \mathcal{N}(0, I)$ using the Cramer-Wold (CW) distance [20]. As the closed-form expression of the Wasserstein Auto-Encoder loss [42], CW provides a way

to mold correct instances into a stable reference manifold. The CW loss is:

$$\mathcal{L}_{\text{CW}} = \frac{1}{N_{\text{pos}}^2} \sum_{i,j \in \mathcal{I}_{\text{pos}}} \phi_d(\|z_i - z_j\|^2) + \frac{1}{N_{\text{pos}}} \sum_{i \in \mathcal{I}_{\text{pos}}} \zeta_d(\|z_i\|^2) + C \tag{6}$$

where $\mathcal{I}_{\text{pos}}$ are the indices of correct predictions, and $\phi_d, \zeta_d$ are kernels based on confluent hypergeometric functions [20]. This clustering creates a dense, predictable success core, providing the necessary contrast for density estimation.

***Margin-based Repulsion (Incorrect Samples)*** To decouple incorrect predictions ($\hat{y} \neq y$) from the compact success manifold, we introduce an asymmetric **Push Loss** ($\mathcal{L}_{push}$). Unlike contrastive learning methods that define repulsion through relative distances between samples, our approach directly penalises misclassifications that fall within a critical radius $\mathcal{R}_{crit}$ from the manifold's center, ensuring a clear discriminative margin:

$$\mathcal{L}_{push} = \frac{1}{|z_{inc}|} \sum_{z \in z_{inc}} \text{ReLU}(\mathcal{R}_{crit} - \|z\|_2)^2 \tag{7}$$

where $z_{inc}$ denotes the set of incorrect predictions in the batch. Following the concentration of measure properties of the Gaussian prior used for correct instances, we set $\mathcal{R}_{crit} = \alpha\sqrt{d}$. The factor $\alpha > 1$ acts as a slack factor to prevent incorrect samples from clustering too close to the success boundary. This safety margin accounts for the discrepancy between theoretical Gaussian concentration and empirical latent structures, thereby avoiding sharp transition zones that would otherwise degrade the reliability of the downstream KDEs.

One CWAE is trained per class, with $\mathcal{L}_{total}$ integrating reconstruction and separation terms between positive and negative samples:

$$\mathcal{L}_{total} = \mathcal{L}_{rec} + \lambda \mathbb{1}_{correct} \mathcal{L}_{CW} + \gamma \mathbb{1}_{incorrect} \mathcal{L}_{push} \tag{8}$$

Figure 3 illustrates the interaction of these three training objectives. By jointly optimising for reconstruction fidelity, distributional cohesion for correct samples, and metric separation for failures, our framework actively sculpts a latent manifold where the density contrast between success and failure is maximised. Providing a topological substrate for the KDE-based trust scoring.

To evaluate the impact of the Push Loss on both detection performance and representational integrity, we conducted an ablation study. The results confirm that $\mathcal{L}_{push}$ is essential for establishing the discriminative margin required for reliable trust estimation. This objective enhances failure detection without compromising reconstruction quality. A analysis is provided in Appendix A.

Figure 4 resumes the complete pipeline, tracing the transition from raw input to the final confidence score, including feature extraction, latent manifold shaping, and KDE-based trust estimation. CALI's implementation is shared online[1].

[1] https://anonymous.4open.science/r/CALI

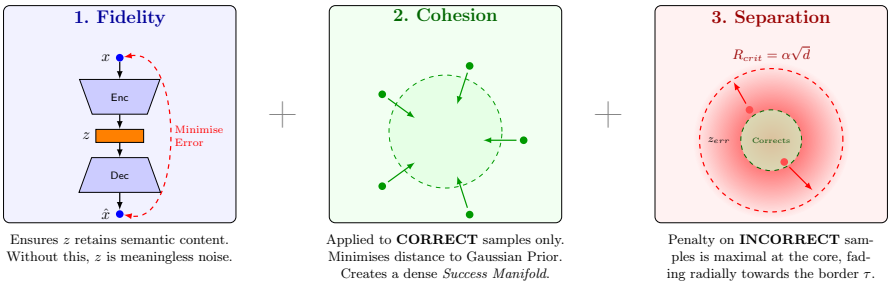


**Fig. 3. Discriminative Manifold Shaping with Class-Specific CWAE.** Our framework partitions the latent space through three complementary objectives: (1) $\mathcal{L}_{rec}$ preserves structural features via input reconstruction; (2) $\mathcal{L}_{CW}$ regularises successful predictions (green) toward a compact Gaussian success core; and (3) $\mathcal{L}_{push}$ repels misclassifications (red) beyond a safety margin $\alpha\sqrt{d}$. This geometric decoupling prevents density overlap, ensuring a discriminative topology for downstream trust estimation.

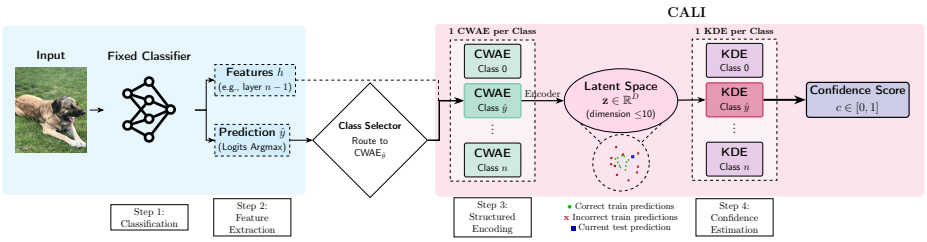


**Fig. 4.** CALI Framework: Step-by-Step Process

## 5 Experiments

### 5.1 Expertimental Setup

We evaluate our framework across 8 architecture-dataset pairs, of varying complexity: MNIST [7], FashionMNIST [44], PneumoniaMNIST [19], CIFAR-10 and 100 [21] and SVHN [31]. These are paired with diverse backbones ranging from standard CNN and VGG-16 [38] to ResNet-18/50 [13], DenseNet-121 [16] and WideResNet-50 [45]. Testing across these varied configurations ensures the framework's backbone-agnostic robustness.

For the tailored CWAE, we adopted a hyperparameter configuration through iterative optimisation across multiple architectures and datasets, ensuring the global applicability of our method: $d = 10$, $\gamma = 0.1$, $\lambda = 10$ and $\alpha = 2.5$. See Appendix B for details. For KDE configuration, we conducted an ablation study to identify a backbone-agnostic setup that provides the best empirical compromise: **Exponential Kernel**, **Euclidean Distance**, and **Silverman's rule** for bandwidth (see Appendix C). Layer-wise, in line with existing literature, our analysis in Appendix D confirms that the penultimate layer yields the most

discriminative features for success-failure separation. We thus use this layer as the default input for all subsequent experiments.

### 5.2 Evaluation Metrics

To assess detection quality, we follow the standard evaluation protocol used in prior works: AUROC, FPR@95TPR, AUPR-Success, AUPR-Error, e-AURC (normalised version of AURC) [14, 17, 4]. However, these metrics are inherently *threshold-free*: they evaluate ranking quality, they do not reflect the suitability of confidence scores for deployment with a fixed decision threshold.

To address this, we use the Wasserstein distance between the confidence distributions of correct and incorrect predictions. Unlike ranking metrics, it captures the global separation and polarisation of scores. A larger distance indicates that incorrect predictions are concentrated near 0 and correct ones near 1, facilitating the selection of a robust operating threshold for real-world deployment [33]. This property illustrated in Figure 5, is crucial in deployment, where a single decision threshold, estimated on held-out calibration data, must remain reliable under inevitable distribution shifts. Compared to existing metrics,


**Fig. 5.** Wasserstein Distance on Confidence Score Distributions

Wasserstein distance explicitly evaluates how well a confidence scoring function supports a stable, human-interpretable threshold for FP, which is a key requirement in many industrial decision pipelines.

Complementing distributional evaluation, we introduce the **Average Risk Increase** (ARI) to quantify operational robustness under threshold sensitivity. In practice, a FP framework relies on a decision threshold $\tau$ applied to the confidence score $\kappa(x, \hat{y})$ to reject potential errors (cf. Eq 2). The optimal threshold ($\tau$) is selected via Youden's point on the ROC curve, though other methods exist [30]. Let $R(\tau)$ denote the conditional risk (i.e. the error rate) among predictions accepted at threshold $\tau$. To measure robustness, we analyse the stability of $R(\tau)$ under local perturbations within a bounded interval $[\tau - \delta, \tau + \delta]$. We intermediary define the **Risk Increase** as the absolute increase in error rate relative to the optimal operating point:

$$\Delta R(\tau) = |R(\tau) - R(\tau \pm \delta)| \tag{9}$$

We then define the ARI as the expected accuracy cost incurred by threshold miscalibration, as the area under the said Risk Increase:

$$\mathrm{ARI} = \frac{1}{2\delta} \int_{\tau^-\delta}^{\tau^+\delta} \Delta R(t) dt \tag{10}$$

Unlike standard aggregate metrics, the ARI provides a tangible operational guarantee: it represents the average percentage points of accuracy lost when the operating point deviates from its optimum. A low ARI characterises a method that maintains a safe performance plateau, whereas a high ARI reveals a brittle sensitivity where minor threshold shifts lead to sharp increases in the error rate.

In our main analysis, we report a focused subset of metrics: FPR@95TPR, AUPR-Error, e-AURC, and our proposed evaluation via Wasserstein distance, to provide a concise yet comprehensive overview of performance. We omit AUROC and AUPR-Success from the main results as these metrics consistently reached near-optimal values, offering limited discriminative insight; however, the full results remain available in Appendix E. The ARI is then analysed independently to highlight its unique contribution to operational robustness.

We compare our approach against the primary paradigms discussed in Section 3: (i) standard baselines like MSP, Energy Score. (ii) Bayesian approximations such as MC-Dropout and Deep Ensembles, (iii) geometric score like Trust Score, and (iv) specialised FP model like ConfidNet.

### 5.3 Experimental Results

Figure 6 presents a comparison across 8 architecture-dataset pairs and 4 key metrics. We highlight several observations:

CALI achieves **1st on Wasserstein distance in all 8 configurations**. This confirms that our tailored class-conditional CWAE and KDE scoring effectively learn and leverage the structured latent representations where correct and incorrect predictions occupy separated regions. Producing polarised scores, facilitating human interpretability and robust decision-making.

CALI ranks **1st on 5/8 systems and 2nd on 3/8 systems** for AUPR-Error. On FPR@95TPR, CALI achieves **1st on 5/8 systems and top-3 on all 8 systems**. This demonstrates that our confidence estimation provides low false alarm rates even at high recall thresholds.

Performance on e-AURC presents a more nuanced picture. CALI ranks **1st on the PneumoniaMNIST medical dataset**, but achieves ranks 3–7 on the remaining systems. We hypothesise this is due to the nature of our confidence scores: strong latent space separation (high Wasserstein) produces polarised confidence distributions rather than one highly peaked one. Figure 7 highlights this: SOTA methods produce a sharp peak for one population, either correct or incorrect predictions, depending on the specific baseline. However, their confidence for the opposing population remains *diffuse*, failing to form a distinct, well-separated counter-peak. This asymmetry artificially favours e-AURC; any highly concentrated distribution provides an easy threshold for a selective prediction mechanism to reject or accept samples, thereby reducing the cumulative risk. However,

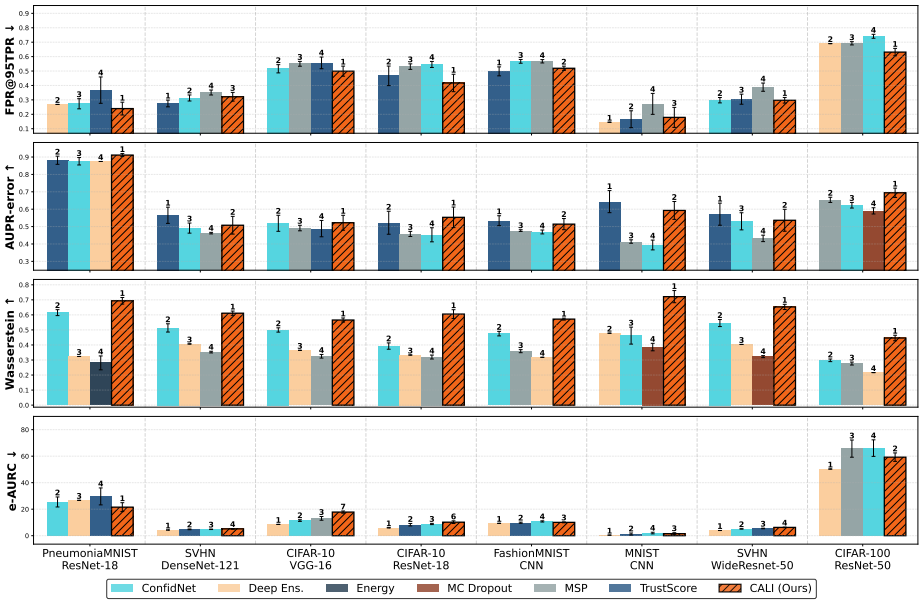


**Fig. 6. Comparative Benchmark on Failure Prediction Performance.** For visual clarity, we show only the top-3 performing methods per metric-system pair, plus CALI (ours) if not already in the top-3. Numbers above bars indicate ranking (1 = best). Error bars represent standard deviation across 5 seeds. Deep Ensemble has no error bars as our 5 model instances form a single ensemble.

this gain is deceptive, as it hides **poor distributional separation** and often leads to **higher FPR@95TPR** (lower is better) compared to our framework. In contrast, CALI maintains a more balanced and physically meaningful polarisation, ensuring that both success and error populations are treated as distinct, identifiable clusters rather than a single peak emerging from a background of uncertainty.

To quantify the operational significance of the distributional separation observed in Figure 7, we therefore introduced a robustness-test based on threshold sensitivity, represented by the ARI (see. Eq. 10). Before displaying the ARIs, we visualise the Risk Increase (see. Eq. 9), in Figure 8. Then Figure 9 illustrates the ARI evolution across increasing perturbation ranges, up to 20% of jitter. The CALI framework consistently outperforms state-of-the-art methods, maintaining superior stability even under significant threshold jitter.

Finally, complexity analysis and runtime benchmarks are provided in Appendix G, showing that CALI achieves competitive training time and negligible inference overhead compared to methods such as ConfidNet or MC-Dropout.

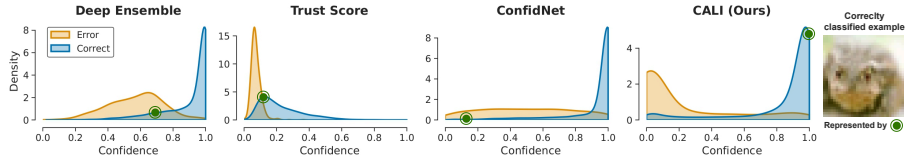


**Fig. 7. Confidence Score Distributions for Success vs. Error.** While top-3 SOTA methods on e-AURC (CIFAR-10/ResNet-18) exhibit sharp but overlapping confidence distributions for correct and incorrect predictions, limiting their discriminative power. CALI, however, achieves superior polarisation and higher Wasserstein distance by clearly separating the two populations. This effect is illustrated by a correctly classified image, where CALI's score reflects an unambiguous decision boundary.

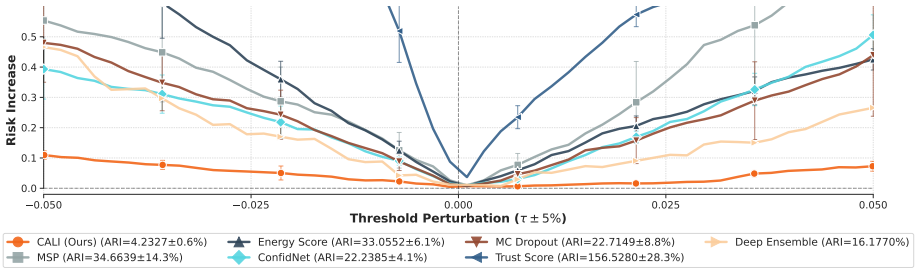


**Fig. 8. Error rate stability under ±5% threshold shifts (CIFAR-10/ResNet-18)**. CALI remains robust (ARI<5%), while baselines degrade sharply (e.g., 16% for the second-best), highlighting poor distributional separation (see Fig. 7).

## 6 Limitations

Our class-wise density estimation faces two constraints: (i) sample availability, as KDE requires both correct and incorrect instances **per class** to model distributions, and (ii) convergence, as non-parametric methods typically scale with sample size. However, our experiments show that compressing the latent space to $d = 10$ effectively mitigates these requirements, ensuring reliable scoring even with sparse failure data. Detailed sample statistics are provided in Appendix F.

## 7 Conclusion

This paper introduces CALI, a class-conditional failure prediction framework using tailored CWAEs. Our approach reshapes the embedding manifold via a custom loss function, improving separation between correct and incorrect predictions. The scoring function, derived from class-conditional KDEs, provides a reliable likelihood of correctness that achieves on-par with SOTA baselines in FPR@95TPR and AUPR-Error. Furthermore, our approach excels in distributional separation, providing an interpretable score and operational robustness,

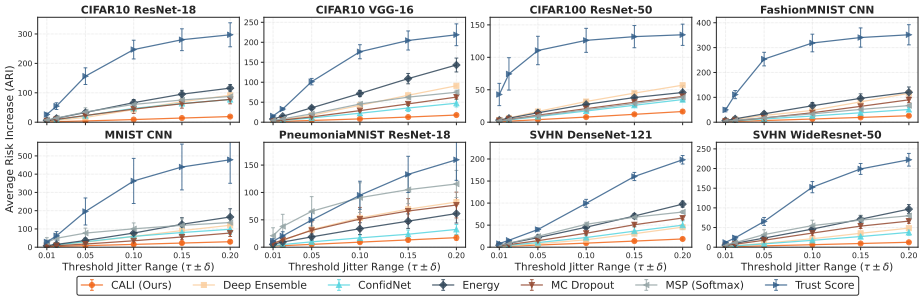


**Fig. 9. Resilience to Threshold Perturbations.** Evolution of the Average Risk Increase (ARI) across different datasets (lower is better). CALI significantly outperforms baselines by maintaining near-constant performance despite threshold jitter, whereas traditional scoring methods show high sensitivity to minor operational shifts.

demonstrated through the ARI metric. We show that our framework maintains stable performance even with threshold variability, offering a safer and more reliable solution for real-world deployment of deep classifiers. Finally, its computational overhead remains lower than that of most baseline methods.

Future work will explore extending this framework to other architectures like transformers. Additionally, we aim to investigate its synergy with Conformal Prediction to refine prediction sets and its integration into Active Learning to optimise sample selection through superior failure discrimination.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Angelopoulos, A.N., Bates, S.: Conformal prediction: A gentle introduction. Foundations and Trends in Machine Learning **16**(4), 494–591 (2023)
2. Bank, D., Koenigstein, N., Giryes, R.: Autoencoders. Machine learning for data science handbook: data mining and knowledge discovery handbook pp. 353–374 (2023)
3. Chen, Y.C.: A tutorial on kernel density estimation and recent advances. Biostatistics & Epidemiology **1**(1), 161–187 (2017)
4. Corbière, C., Thome, N., Bar-Hen, A., Cord, M., Pérez, P.: Addressing failure prediction by learning model confidence. Advances in neural information processing systems **32** (2019)
5. Dadalto, E., Romanelli, M., Pichler, G., Piantanida, P.: A data-driven measure of relative uncertainty for misclassification detection. arXiv preprint arXiv:2306.01710 (2023)
6. Dai, B., Wang, Z., Wipf, D.: The usual suspects? reassessing blame for vae posterior collapse. In: International conference on machine learning. pp. 2313–2322. PMLR (2020)

7. Deng, L.: The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE signal processing magazine **29**(6), 141–142 (2012)
8. European Defence Agency: Trustworthiness for AI in defence (2025), https://eda.europa.eu/docs/default-source/brochures/taid-white-paper-final-09052025.pdf
9. European Parliament and Council: Regulation (EU) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence (artificial intelligence act). Official Journal of the European Union L 2024/1689 (2024), https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng
10. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059. PMLR (2016)
11. Geifman, Y., Uziel, G., El-Yaniv, R.: Bias-reduced uncertainty estimation for deep neural classifiers. arXiv preprint arXiv:1805.08206 (2018)
12. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International conference on machine learning. pp. 1321–1330. PMLR (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136 (2016)
15. Hofmann, C., Schmid, S., Lehner, B., Klotz, D., Hochreiter, S.: Energy-based hopfield boosting for out-of-distribution detection. Advances in Neural Information Processing Systems **37**, 131859–131919 (2024)
16. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
17. Jaeger, P.F., Lüth, C.T., Klein, L., Bungert, T.J.: A call to reflect on evaluation practices for failure detection in image classification. arXiv preprint arXiv:2211.15259 (2022)
18. Jiang, H., Kim, B., Guan, M., Gupta, M.: To trust or not to trust a classifier. Advances in neural information processing systems **31** (2018)
19. Kermany, D.S., Goldbaum, M., Cai, W., Valentim, C.C., Liang, H., Baxter, S.L., McKeown, A., Yang, G., Wu, X., Yan, F., et al.: Identifying medical diagnoses and treatable diseases by image-based deep learning. cell **172**(5), 1122–1131 (2018)
20. Knop, S., Tabor, J., Podolak, I., Mazur, M., et al.: Cramer-wold auto-encoder. Journal of Machine Learning Research **21**(164), 1–28 (2020)
21. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images.(2009) (2009)
22. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in neural information processing systems **30** (2017)
23. Lazarow, J., Jin, L., Tu, Z.: Introspective neural networks for generative modeling. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2774–2783 (2017)
24. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. Advances in neural information processing systems **31** (2018)

25. Li, J., Zhou, Y., Yao, J., Liu, X.: An empirical investigation of trust in ai in a chinese petrochemical enterprise based on institutional theory. Scientific reports **11**(1), 13564 (2021)
26. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based out-of-distribution detection. Advances in neural information processing systems **33**, 21464–21475 (2020)
27. Meding, I., Bodin, A., Tonderski, A., Johnander, J., Petersson, C., Svensson, L.: You can have your ensemble and run it too-deep ensembles spread over time. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4020–4029 (2023)
28. Moon, J., Kim, J., Shin, Y., Hwang, S.: Confidence-aware learning for deep neural networks. In: international conference on machine learning. pp. 7034–7044. PMLR (2020)
29. Mukhoti, J., Kirsch, A., Van Amersfoort, J., Torr, P.H., Gal, Y.: Deep deterministic uncertainty: A new simple baseline. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24384–24394 (2023)
30. Nahm, F.S.: Receiver operating characteristic curve: overview and practical use for clinicians. Korean journal of anesthesiology **75**(1), 25–36 (2022)
31. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y., et al.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning. vol. 2011, p. 4. Granada (2011)
32. Ozaki, Y., Watanabe, S., Yanase, T.: OptunaHub: A platform for black-box optimization. arXiv preprint arXiv:2510.02798 (2025)
33. Panaretos, V.M., Zemel, Y.: Statistical aspects of wasserstein distances. Annual review of statistics and its application **6**(1), 405–431 (2019)
34. Papernot, N., McDaniel, P.: Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. arXiv preprint arXiv:1803.04765 (2018)
35. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144 (2016)
36. Shafer, G., Vovk, V.: A tutorial on conformal prediction. Journal of Machine Learning Research **9**(3) (2008)
37. Silverman, B.W.: Density estimation for statistics and data analysis. Routledge (2018)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
39. Spitzer, P., Holstein, J., Morrison, K., Holstein, K., Satzger, G., Kühl, N.: Don't be fooled: the misinformation effect of explanations in human–ai collaboration. International Journal of Human–Computer Interaction pp. 1–29 (2025)
40. Tenenbaum, J.B., Silva, V.d., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. science **290**(5500), 2319–2323 (2000)
41. The White House: America's AI action plan (2025)
42. Tolstikhin, I., Bousquet, O., Gelly, S., Schoelkopf, B.: Wasserstein auto-encoders. arXiv preprint arXiv:1711.01558 (2017)
43. Venkataramanan, A., Benbihi, A., Laviale, M., Pradalier, C.: Gaussian latent representations for uncertainty estimation using mahalanobis distance in deep classifiers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4488–4497 (2023)
44. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
45. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)

46. Zhu, F., Cheng, Z., Zhang, X.Y., Liu, C.L.: Rethinking confidence calibration for failure prediction. In: European conference on computer vision. pp. 518–536. Springer (2022)
47. Zhu, F., Cheng, Z., Zhang, X.Y., Liu, C.L.: Openmix: Exploring outlier samples for misclassification detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12074–12083 (2023)

# A Impact of Geometric Push Loss on Failure Prediction

## A.1 Experimental Protocol

To validate the contribution of the proposed loss component, we conduct an ablation study across four distinct dataset-architecture pairs: FashionMNIST (CNN), CIFAR-10 (ResNet-18 and VGG-16), and SVHN (DenseNet-121). This selection ensures that our findings are consistent across varying complexities and architectural designs. For each experiment, we utilise a subset of 60% of the original training data, to reduce the computational burden of ablation studies. We then adopt an 80/20 split on this subset for training and internal validation. All results are averaged over 5 independent seeds, totalling 60 experimental runs to ensure statistical significance. Consistent with the literature, all latent features are extracted from the penultimate layer $(n-1)$.

**Loss Variants and Objectives** We evaluate three configurations of our latent space optimisation:

1. **Baseline (CWAE)**: Standard Cramer-Wold Autoencoder without failure-specific constraints.
2. **Push Loss ($\mathcal{L}_{push}$)**: The proposed objective defined as:

$$\mathcal{L}_{push} = \frac{1}{|Z_{inc}|} \sum_{z \in Z_{inc}} \text{ReLU}(\alpha\tau - \|z\|_2)^2 \tag{11}$$

3. **Push-Pull Loss ($\mathcal{L}_{pp}$)**: An extension adding a *pull* constraint on correct samples to enforce tighter concentration within the theoretical radius:

$$\mathcal{L}_{pp} = \mathcal{L}_{push} + \frac{1}{|Z_{corr}|} \sum_{z \in Z_{corr}} \text{ReLU}(\|z\|_2 - \tau)^2 \tag{12}$$

Where $\tau = \sqrt{D}$ is the theoretical radius of a $D$-dimensional Gaussian and $\alpha = 2.5$ serves as the safety margin.

**Evaluation Metrics** We assess performance using a dual-perspective approach:

- **Failure Prediction**: Evaluated via FPR@95TPR, AUPR-error, and e-AURC, focusing on the model's ability to rank misclassifications.
- **Distributional Separability**: Measured via the Wasserstein distance between the latent norms of correct and incorrect samples.
- **Reconstruction Fidelity**: Monitored through MSE, ensuring that failure-specific constraints do not distort the latent space to the point of compromising feature representational integrity.

**Results** The ablation study resumed in Table 1 highlights a clear distinction in the behaviour of the two proposed objectives. On one hand, the Push Loss yields a consistent and robust improvement over the baseline, achieving a 28% average gain across all evaluation criteria with a relatively low dispersion ($\sigma = 10.0$). On the other hand, while the Push-Pull Loss occasionally reaches higher peak performances, its adoption is precluded by two critical factors. First, it induces a significant performance degradation on e-AURC, representing a detrimental trade-off that compromises the overall reliability of the failure prediction. Second, this variant exhibits more instability, as evidenced by a higher standard deviation ($\sigma = 17.0$ compared to 10.0 for the Push Loss). These results suggest that over-constraining the latent space leads to stochastic sensitivity, making the simpler Push Loss the most reliable candidate for real-world deployment. Regarding reconstruction fidelity, the Push Loss maintains the latent space's integrity with an MSE increase below 30%. While this relative change seems high, it remains negligible in absolute terms, order of $10^{-2}$ to $10^{-1}$, ensuring that feature representation is preserved. We therefore adopt the **Push-Loss as additional loss** in our tailored CWAE.

**Table 1.** Absolute improvement over baseline (%) for each configuration.

| Dataset | Architecture | Config | FPR@95TPR | AUPR-Error | Wasserstein | e-AURC | Recon MSE |
|---|---|---|---|---|---|---|---|
| CIFAR-10 | ResNet-18 | Push | **31.04 ± 9.45** | 36.80 ± 20.73 | **39.89 ± 8.57** | **37.23 ± 8.99** | 9.63 ± 0.56 |
| | | Push+Pull | 30.30 ± 7.96 | **50.13 ± 12.19** | 9.24 ± 11.00 | -26.32 ± 27.67 | 17.62 ± 1.75 |
| | VGG | Push | 5.13 ± 6.06 | 15.40 ± 4.95 | **103.38 ± 6.66** | **27.53 ± 7.22** | 21.10 ± 3.50 |
| | | Push+Pull | **16.24 ± 5.39** | **24.32 ± 7.07** | 48.19 ± 14.74 | -80.99 ± 41.34 | 89.65 ± 6.32 |
| FashionMNIST | CNN | Push | 7.04 ± 3.10 | 18.61 ± 5.69 | 46.72 ± 3.07 | 22.70 ± 3.31 | 21.10 ± 1.51 |
| | | Push+Pull | **28.81 ± 3.79** | **59.50 ± 12.54** | **60.20 ± 2.52** | **27.18 ± 5.41** | 67.23 ± 2.94 |
| SVHN | DenseNet | Push | 7.35 ± 7.39 | 7.46 ± 5.85 | **27.17 ± 3.95** | **17.55 ± 18.01** | 27.54 ± 0.91 |
| | | Push+Pull | **27.16 ± 9.24** | **45.61 ± 15.63** | 20.35 ± 7.61 | -2.51 ± 22.42 | 75.91 ± 4.69 |

## B Hyperparameter Optimization

To identify the optimal configuration for our adapted CWAE, we conducted a three-stage optimisation process using the Optuna framework [32]. The search focused on four key hyperparameters: the **latent dimension** ($z$), the **Cramer-Wold loss weight** ($\lambda$), the **Push Loss weight** ($\gamma$), and the **geometric margin factor** ($\alpha$). As in Appendix A, the optimisation was performed across 4 pairs : 3 datasets $\times$ 4 architectures (CIFAR-10 VGG-16 and ResNet-18, FashionMNIST CNN, SVHN DenseNet-121). The three stages were:

1. **Exploration**: A broad search over a wide range of values to identify regions of high performance. See Table 2.
2. **Refinement**: A narrowed search around the preliminary optima to fine-tune the parameters. See Table 3.

3. **Consensus**: We analysed the frequency of optimal values across all configurations, to conclude on the best hyperparameter values. See Table 4.

In the first phase, showed in Table 2, a broad exploration revealed trends across most configurations, with latent dimensions generally $z \geq 8$, loss push weights $\gamma < 0.5$, and $\lambda$ around 1 or 10. The geometric margin factor $\alpha$ stabilised between 2 and 4, though the CNN architecture consistently deviated from these patterns, likely due to its simpler structure and data characteristics.

**Table 2.** Phase 1: Initial Exploration Ranges and Local Optima per Pair

| | | **FashionMNIST** | **CIFAR-10** | | **SVHN** |
|---|---|---|---|---|---|
| **HP** | **Search Range** | CNN | ResNet-18 | VGG-16 | DenseNet-121 |
| $z$ | $[2, 5, 8, 10]$ | 2 | 8 | 10 | 8 |
| $\gamma$ | $[0.1, 0.5, 1.0, 5.0, 10.0]$ | 0.5 | 0.1 | 0.1 | 0.1 |
| $\lambda$ | $[0.1, 0.5, 1.0, 5.0, 10.0]$ | 0.5 | 1.0 | 10.0 | 1.0 |
| $\alpha$ | $[1.5, 4.0]_{step=0.5}$ | 2.0 | 4.0 | 2.4 | 3.0 |

Building on these insights, the second phase resumed in Table 3, narrowed the search intervals to refine the optimal ranges. For $z$, values clustered around 9 to 11, with $z$=10 appearing most frequently. The push loss weight converged to 0.1 in most cases, while the Cramer-Wold weight stabilised at 10 for complex architectures, though the CNN favours $\lambda$=5. The geometric margin factor $\alpha$ showed small variability, with 2.5 emerging as a common value. This refinement highlighted a regularisation effect, balancing model complexity and generalisation.

**Table 3.** Phase 2: Refined Search Ranges based on Phase 1 Modes

| | | **FashionMNIST** | **CIFAR-10** | | **SVHN** |
|---|---|---|---|---|---|
| **HP** | **Search Range** | CNN | ResNet-18 | VGG-16 | DenseNet-121 |
| $z$ | $[8, 11]_{step=1}$ | 9 | 11 | 10 | 10 |
| $\gamma$ | $[0.1, 0.3]_{step=0.1}$ | 0.1 | 0.3 | 0.1 | 0.1 |
| $\lambda$ | $[1.0, 5.0, 8.0, 10.0]$ | 5.0 | 10.0 | 10.0 | 10.0 |
| $\alpha$ | $[1.5, 3.5]_{step=0.5}$ | 1.5 | 2.5 | 2.5 | 3.0 |

In the final consensus phase in Table 4, we selected **hyperparameters that are architecture- and dataset-agnostic** based on their frequency across all configurations. The resulting configuration, $z = 10$, $\gamma = 0.1$, $\lambda = 10$, and $\alpha = 2.5$, not only aligned with the best-performing setup for VGG-16 on CIFAR-10 but also consistently outperformed individual optima on other validation sets. This suggests that prioritising cross-architecture generalisation over local optima mitigates overfitting to specific validation set noise, enabling convergence toward

a structurally stable region of the parameter space. Such robustness is critical for real-world applications where optimal configurations are often unknown. By validating this approach across diverse architectures and datasets, our work underscores the importance of hyperparameter tuning strategies that balance performance and generalisation, paving the way for scalable and reliable CWAE implementations.

**Table 4.** Phase 3: Validation Performance Comparison

| Dataset | Architecture | Config | FPR@95TPR (↓) | AUPR-Error (↑) | Wasserstein (↑) | e-AURC (↓) | Recon MSE (↓) |
|---|---|---|---|---|---|---|---|
| CIFAR-10 | ResNet-18 | Best Loc. | **0.30** | 0.54 | 0.69 | 6.66 | 0.16 |
| | | Glob. Cons. | 0.31 | **0.55** | 0.69 | **6.01** | 0.16 |
| | VGG | Best Loc. | 0.42 | 0.42 | 0.64 | 9.80 | 0.00 |
| | | Glob. Cons.(=) | 0.42 | 0.42 | 0.64 | 9.80 | 0.00 |
| FashionMNIST | CNN | Best Loc. | 0.62 | 0.36 | 0.61 | 9.17 | 0.07 |
| | | Glob. Cons. | **0.61** | **0.37** | **0.68** | **8.00** | 0.08 |
| SVHN | DenseNet | Best Loc. | 0.33 | 0.38 | 0.70 | 5.87 | 0.02 |
| | | Glob. Cons. | **0.32** | **0.41** | 0.70 | **4.57** | 0.02 |

## C Ablation on KDE Hyperparameters

This appendix presents the complete numerical results of the hyperparameter ablation study to determine the optimal configuration for our KDE-based confidence score. It has been conducted on the same dataset-architecture pairs than for the previous studies A and B: CIFAR-10 VGG-16 and ResNet-18, FashionMNIST CNN, SVHN DenseNet-121. The performance of each configuration was evaluated on the penultimate layer, as before, as it is known to contain the most information for confidence estimation. The study was conducted over the following hyperparameters:

- **Bandwidth**: Scott's rule, Silverman's rule, Grid Search (10 values log-spaced from 0.1 to 1)
- **Distance**: Euclidean, Manhattan, Mahalanobis
- **Kernel**: Gaussian, Exponential

To identify the optimal configuration, we rank the candidates based on their **aggregated mean score across the four evaluation metrics**: FPR@95TPR, AUPR-Error, Wassertein and e-AURC. Specifically, we compute the average of the normalised e-AURC values for each combination; this composite index ensures a balanced compromise across all failure prediction criteria. The resulting rankings, summarised in Table 5, prioritise configurations that offer the most stable performance across the entire metric suite.

Our analysis, detailed in Table 5 identifies a dominant hyperparameter configuration that emerged as the top performer across three out of four benchmarks: **Silverman, Mahalanobis, Exponential**. While a different combination ranked first for the outlier pair (SVHN-DensNet-121), this local optimum

**Table 5.** Top 5 Hyperparameter Combinations Ranked by Failure Detection Performance.

| Rank | F-MNIST (CNN) | CIFAR-10 (ResNet-18) | CIFAR-10 (VGG-16) | SVHN (DenseNet-121) |
|---|---|---|---|---|
| 1st | **Silver./Maha./Exp.**$^\star$ | **Silver./Maha./Exp.**$^\star$ | **Silver./Maha./Exp.**$^\star$ | Silver./Eucl./Exp. |
| 2nd | Scott/Maha./Exp. | Scott/Maha./Exp. | Scott/Maha./Exp. | Scott/Eucl./Exp. |
| 3rd | G.S./Maha./Exp. | G.S./Maha./Exp. | G.S./Maha./Exp. | G.S./Eucl./Exp. |
| 4th | G.S./Maha./Gaussian | G.S./Maha./Gaussian | G.S./Maha./Gaussian | **Silver./Maha./Exp.**$^\star$ |
| 5th | Scott/Maha./Gaussian | Scott/Maha./Gaussian | Scott/Maha./Gaussian | Scott/Man./Exp. |

$^\star$Selected Consensus Configuration: [Kernel Exponential, Distance Metric Mahalanobis, Bandwidth Silverman]

failed to reach the top 5 on other datasets, indicating poor generalisation. In contrast, our consensus configuration demonstrated resilience, maintaining a fourth-place ranking even on the outlier pair. By prioritising this cross-dataset-architecture stability over unstable local peaks, we ensure a robust, backbone-agnostic framework that eliminates the need for per-task hyperparameter tuning.

## D Layer Selection Analysis

We conduct an ablation study to determine which internal layer(s) of the classifier provide the most informative features for failure prediction. For four representative architecture-dataset pairs, as in studies A, B and C, we evaluate all possible layer combinations on a held-out validation split of the training set and select the combination that maximises the global score averaging four metrics: FPR@95TPR, AUPR-Error, Wasserstein distance, and e-AURC.

For each candidate layer $\ell$, we apply our complete failure prediction framework:

1. Extract features $h_\ell$ from layer $\ell$ for each sample
2. Train a class-conditional WAE on these features to obtain latent representations $z_\ell$
3. Fit a KDE model per class on the training latent space
4. Compute confidence scores $c_\ell \in [0, 1]$ for validation samples using the KDE

For multi-layer combinations $\mathcal{L} = \{\ell_1, \ldots, \ell_k\}$, we obtain a final confidence score by **averaging the individual confidence scores**:

$$c_{\mathcal{L}} = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} c_\ell \tag{13}$$

We then compute the **layer importance** $\Delta_\ell$ for each layer $\ell$ as:

$$\Delta_\ell = \mathbb{E}[\text{score} \mid \ell \in \mathcal{L}] - \mathbb{E}[\text{score} \mid \ell \notin \mathcal{L}] \tag{14}$$

where the expectation is taken over all tested combinations. A positive $\Delta_\ell$ indicates that including layer $\ell$'s confidence scores in the average improves performance, while a negative value suggests it introduces noise or redundancy.

Figure 10 presents the best layer configuration for each architecture-dataset pair. Across all tested pairs, **single-layer confidence scores outperform averaged multi-layer combinations**, suggesting that combining confidence estimates from multiple depths dilutes the signal rather than enriching it.

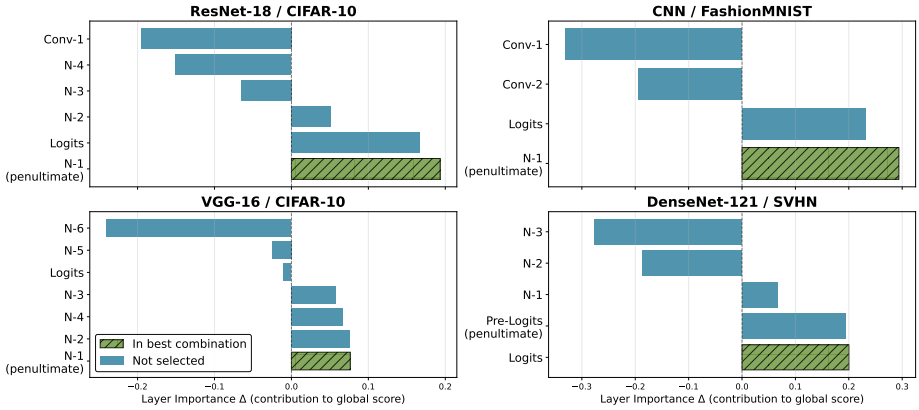


**Fig. 10.** Best layer selection per architecture-dataset pair.

Three out of four pairs achieve optimal performance using the penultimate layer, while DenseNet-121 on SVHN favors the logits layer. However, the performance gap between these two layers is minimal across all configurations ($< 1.2\%$ relative difference). To ensure a **unified, architecture-agnostic configuration** that simplifies deployment and enhances reproducibility, we adopt **the penultimate layer as the universal feature extraction point** for all experiments in the main paper.

## E Main Results

This appendix provides comprehensive results for all methods across all experimental configurations. Tables 6–13 present detailed performance metrics for each architecture-dataset pair, reporting mean and standard deviation across 5 independent runs with different random seeds. The method Deep Ensemble does not have a standard deviation since 5 runs forms a single result. **Bold values** indicate the best-performing method per metric (within a tolerance of $\pm 0.001$ to account for numerical precision).

## F Experience Samples

Table 14 details the per-class sample distribution for each dataset. It reports the average and minimum number of correct and incorrect predictions available to fit the failure prediction methods, conditioned on the classifier's performance.

**Table 6.** Complete results for CIFAR10 - ResNet-18.

| Method | FPR@95TPR (↓) | AUPR-error (↑) | Wasserstein (↑) | e-AURC (↓) | AUROC (↑) | AUPR-success (↑) |
|---|---|---|---|---|---|---|
| **CALI (Ours)** | **0.417 ± 0.060** | **0.553 ± 0.059** | **0.605 ± 0.029** | 10.200 ± 1.005 | 0.913 ± 0.010 | 0.989 ± 0.001 |
| Deep Ensemble | 0.567 ± 0.000 | 0.356 ± 0.000 | 0.333 ± 0.000 | **5.971 ± 0.000** | 0.914 ± 0.000 | **0.994 ± 0.000** |
| ConfidNet | 0.546 ± 0.021 | 0.453 ± 0.040 | 0.393 ± 0.021 | 8.834 ± 0.459 | 0.911 ± 0.006 | 0.991 ± 0.000 |
| TrustScore | 0.467 ± 0.068 | 0.522 ± 0.066 | 0.118 ± 0.023 | 8.299 ± 0.926 | **0.918 ± 0.012** | 0.991 ± 0.001 |
| MC Dropout | 0.571 ± 0.026 | 0.435 ± 0.009 | 0.293 ± 0.013 | 9.775 ± 0.845 | 0.904 ± 0.004 | 0.989 ± 0.001 |
| Energy Score | 0.693 ± 0.017 | 0.326 ± 0.004 | 0.154 ± 0.008 | 17.612 ± 1.773 | 0.843 ± 0.008 | 0.981 ± 0.002 |
| MSP | 0.531 ± 0.019 | 0.458 ± 0.015 | 0.320 ± 0.013 | 9.503 ± 0.795 | 0.907 ± 0.003 | 0.990 ± 0.001 |

**Table 7.** Complete results for CIFAR10 - VGG-16.

| Method | FPR@95TPR (↓) | AUPR-error (↑) | Wasserstein (↑) | e-AURC (↓) | AUROC (↑) | AUPR-success (↑) |
|---|---|---|---|---|---|---|
| **CALI (Ours)** | **0.499 ± 0.035** | **0.522 ± 0.043** | **0.566 ± 0.014** | 17.826 ± 0.814 | 0.885 ± 0.007 | 0.980 ± 0.001 |
| Deep Ensemble | 0.565 ± 0.000 | 0.403 ± 0.000 | 0.364 ± 0.000 | **8.527 ± 0.000** | 0.906 ± 0.000 | **0.991 ± 0.000** |
| ConfidNet | 0.516 ± 0.029 | 0.519 ± 0.046 | 0.500 ± 0.014 | 11.424 ± 0.619 | **0.909 ± 0.003** | 0.987 ± 0.001 |
| TrustScore | 0.557 ± 0.040 | 0.488 ± 0.047 | 0.140 ± 0.010 | 13.272 ± 0.545 | 0.896 ± 0.005 | 0.985 ± 0.001 |
| MC Dropout | 0.582 ± 0.020 | 0.471 ± 0.015 | 0.315 ± 0.010 | 13.428 ± 1.404 | 0.895 ± 0.004 | 0.985 ± 0.002 |
| Energy Score | 0.656 ± 0.012 | 0.412 ± 0.015 | 0.188 ± 0.010 | 17.187 ± 1.813 | 0.869 ± 0.004 | 0.981 ± 0.002 |
| MSP | 0.550 ± 0.017 | 0.492 ± 0.016 | 0.325 ± 0.012 | 13.089 ± 1.411 | 0.898 ± 0.004 | 0.986 ± 0.002 |

## G Computational Complexity Analysis

In this section, we formalise the computational complexity of our proposed framework, CALI, and compare it against standard failure prediction baselines.

Let $\mathcal{O}(C_{bb})$ denote the cost of a single backbone forward pass, $D$ the dimension of extracted features, and $d$ the CWAE latent dimension ($d \ll D$, typically $d \leq 10$). For distance-based and non-parametric methods, let $M$ represent the number of reference samples stored per class, and $C$ the total number of classes.

*Training Complexity.* CALI trains one lightweight CWAE per class on frozen backbone features. For a batch of size $B$, the forward/backward pass costs $\mathcal{O}(B \cdot D \cdot d)$, the Cramer-Wold regularisation (only on correct samples) $\mathcal{O}(B^2 \cdot d)$ and the Push Loss (only on incorrect samples) $\mathcal{O}(B \cdot d)$. Since each CWAE models a single class in low-dimensional space, training converges quickly and requires few epochs ($E_{CWAE}$). After training, a KDE is fitted per class on $M$ samples in $\mathbb{R}^d$.

*Inference Complexity.* At test time, inference involves three sequential operations: (1) backbone feature extraction $\mathcal{O}(C_{bb})$, (2) encoding via the predicted class's CWAE encoder $\mathcal{O}(D \cdot d)$, and (3) KDE scoring against that class's support set $\mathcal{O}(M \cdot d)$. By avoiding computations across all classes, our inference complexity is strictly bounded by $\mathcal{O}(C_{bb} + D \cdot d + M \cdot d)$.

A summary of theoretical complexities for both training and inference is provided in Table 15.

### G.1 Comparative Analysis against Baselines

While post-hoc methods like MSP and Energy Score offer zero-overhead inference $\mathcal{O}(C_{bb})$, as shown in 5.3, they lack the structural reliability required for robust

**Table 8.** Complete results for CIFAR100 - ResNet-50.

| Method | FPR@95TPR (↓) | AUPR-error (↑) | Wasserstein (↑) | e-AURC (↓) | AUROC (↑) | AUPR-success (↑) |
|---|---|---|---|---|---|---|
| **CALI (Ours)** | **0.631 ± 0.026** | **0.694 ± 0.027** | **0.446 ± 0.018** | 59.277 ± 3.186 | **0.847 ± 0.008** | 0.921 ± 0.005 |
| Deep Ensemble | 0.690 ± 0.000 | 0.572 ± 0.000 | 0.216 ± 0.000 | **50.289 ± 0.000** | 0.825 ± 0.000 | **0.938 ± 0.000** |
| ConfidNet | 0.741 ± 0.014 | 0.623 ± 0.016 | 0.298 ± 0.009 | 66.095 ± 6.238 | 0.821 ± 0.008 | 0.913 ± 0.010 |
| TrustScore | 0.782 ± 0.012 | 0.585 ± 0.017 | 0.034 ± 0.011 | 70.730 ± 5.729 | 0.799 ± 0.007 | 0.908 ± 0.009 |
| MC Dropout | 0.755 ± 0.016 | 0.590 ± 0.018 | 0.197 ± 0.007 | 83.524 ± 8.049 | 0.784 ± 0.011 | 0.890 ± 0.012 |
| Energy Score | 0.861 ± 0.016 | 0.465 ± 0.020 | 0.073 ± 0.005 | 126.105 ± 10.393 | 0.695 ± 0.015 | 0.837 ± 0.015 |
| MSP | 0.692 ± 0.011 | 0.654 ± 0.015 | 0.276 ± 0.010 | 65.712 ± 6.487 | 0.826 ± 0.009 | 0.913 ± 0.010 |

**Table 9.** Complete results for FashionMNIST - CNN.

| Method | FPR@95TPR (↓) | AUPR-error (↑) | Wasserstein (↑) | e-AURC (↓) | AUROC (↑) | AUPR-success (↑) |
|---|---|---|---|---|---|---|
| **CALI (Ours)** | 0.519 ± 0.015 | 0.514 ± 0.032 | **0.572 ± 0.006** | 10.068 ± 0.323 | 0.912 ± 0.002 | **0.989 ± 0.000** |
| Deep Ensemble | 0.601 ± 0.000 | 0.441 ± 0.000 | 0.318 ± 0.000 | **9.317 ± 0.000** | 0.906 ± 0.000 | **0.990 ± 0.000** |
| ConfidNet | 0.567 ± 0.013 | 0.469 ± 0.011 | 0.475 ± 0.015 | 10.772 ± 0.455 | 0.905 ± 0.004 | 0.988 ± 0.001 |
| TrustScore | **0.498 ± 0.031** | **0.534 ± 0.028** | 0.073 ± 0.004 | 9.799 ± 0.433 | **0.914 ± 0.004** | **0.989 ± 0.000** |
| MC Dropout | 0.612 ± 0.018 | 0.451 ± 0.009 | 0.317 ± 0.013 | 11.680 ± 0.655 | 0.898 ± 0.003 | 0.987 ± 0.001 |
| Energy Score | 0.695 ± 0.011 | 0.358 ± 0.010 | 0.173 ± 0.011 | 18.371 ± 1.061 | 0.849 ± 0.006 | 0.980 ± 0.001 |
| MSP | 0.568 ± 0.012 | 0.477 ± 0.005 | 0.359 ± 0.010 | 10.923 ± 0.585 | 0.904 ± 0.003 | 0.988 ± 0.001 |

failure prediction. Conversely, methods that improve reliability often introduce prohibitive computational bottlenecks:

- **Bayesian Approximations:** Methods relying on multiple stochastic passes (e.g., MC-Dropout with $T$ passes) or Deep Ensembles (with $K$ independent models) scale the inference cost linearly by a factor of $T$ or $K$. This makes them largely unsuitable for latency-sensitive deployment.
- **Auxiliary Networks:** ConfidNet requires joint or sequential training with the backbone over the entire dataset. Tasked with the complex objective of global true-class probability regression, it suffers from slow convergence (typically requiring $E_{confidnet} \approx 200$ epochs). This significantly inflates the overall training complexity compared to our class-wise CWAE, which reaches optimal manifold shaping in a fraction of the time ($E_{CWAE} \approx 40$ epochs).

We provide empirical runtime measurements to validate the theoretical complexities reported in Table 15. All experiments are conducted on CIFAR-10 (50,000 training / 10,000 test images) using a ResNet-18 backbone. All computations were performed on a single Nvidia Tesla V100 GPU (32GB). Reported times are reported on the full training and test sets.

CALI trains in less than half the time of ConfidNet, while inference remains negligible at under 10 seconds for 10,000 samples ( 0.6ms per sample).

**Table 10.** Complete results for MNIST - CNN.

| Method | FPR@95TPR (↓) | AUPR-error (↑) | Wasserstein (↑) | e-AURC (↓) | AUROC (↑) | AUPR-success (↑) |
|---|---|---|---|---|---|---|
| **CALI (Ours)** | 0.179 ± 0.069 | 0.593 ± 0.051 | **0.721 ± 0.040** | 1.518 ± 0.743 | 0.963 ± 0.009 | 0.998 ± 0.001 |
| Deep Ensemble | **0.145 ± 0.000** | 0.290 ± 0.000 | 0.478 ± 0.000 | **0.434 ± 0.000** | **0.969 ± 0.000** | **1.000 ± 0.000** |
| ConfidNet | 0.302 ± 0.048 | 0.394 ± 0.028 | 0.462 ± 0.056 | 2.003 ± 0.605 | 0.945 ± 0.005 | 0.998 ± 0.001 |
| TrustScore | 0.166 ± 0.059 | **0.643 ± 0.063** | 0.137 ± 0.039 | 1.268 ± 0.759 | 0.967 ± 0.010 | **0.999 ± 0.001** |
| MC Dropout | 0.318 ± 0.083 | 0.362 ± 0.011 | 0.386 ± 0.025 | 2.229 ± 0.903 | 0.942 ± 0.009 | 0.998 ± 0.001 |
| Energy Score | 0.440 ± 0.070 | 0.306 ± 0.025 | 0.195 ± 0.037 | 3.979 ± 2.388 | 0.905 ± 0.028 | 0.996 ± 0.002 |
| MSP | 0.272 ± 0.073 | 0.414 ± 0.010 | 0.331 ± 0.023 | 2.080 ± 0.947 | 0.946 ± 0.010 | 0.998 ± 0.001 |

**Table 11.** Complete results for PneumoniaMNIST - ResNet-18.

| Method | FPR@95TPR (↓) | AUPR-error (↑) | Wasserstein (↑) | e-AURC (↓) | AUROC (↑) | AUPR-success (↑) |
|---|---|---|---|---|---|---|
| **CALI (Ours)** | **0.240 ± 0.044** | **0.911 ± 0.008** | **0.694 ± 0.022** | **21.543 ± 3.358** | **0.947 ± 0.007** | **0.968 ± 0.005** |
| Deep Ensemble | 0.268 ± 0.000 | 0.874 ± 0.000 | 0.324 ± 0.000 | 26.671 ± 0.000 | 0.939 ± 0.000 | 0.960 ± 0.000 |
| ConfidNet | 0.273 ± 0.035 | 0.876 ± 0.021 | 0.615 ± 0.020 | 25.452 ± 3.779 | 0.938 ± 0.007 | 0.963 ± 0.006 |
| TrustScore | 0.368 ± 0.091 | 0.882 ± 0.023 | 0.207 ± 0.040 | 29.643 ± 6.393 | 0.928 ± 0.014 | 0.957 ± 0.009 |
| MC Dropout | 0.470 ± 0.161 | 0.792 ± 0.059 | 0.266 ± 0.060 | 46.464 ± 8.256 | 0.897 ± 0.016 | 0.933 ± 0.012 |
| Energy Score | 0.465 ± 0.116 | 0.791 ± 0.046 | 0.281 ± 0.046 | 49.383 ± 12.081 | 0.891 ± 0.019 | 0.928 ± 0.018 |
| MSP | 0.470 ± 0.161 | 0.792 ± 0.059 | 0.146 ± 0.048 | 46.464 ± 8.256 | 0.897 ± 0.016 | 0.933 ± 0.012 |

**Table 12.** Complete results for SVHN - DenseNet-121.

| Method | FPR@95TPR (↓) | AUPR-error (↑) | Wasserstein (↑) | e-AURC (↓) | AUROC (↑) | AUPR-success (↑) |
|---|---|---|---|---|---|---|
| **CALI (Ours)** | 0.322 ± 0.030 | 0.508 ± 0.051 | **0.611 ± 0.014** | 5.161 ± 0.254 | 0.926 ± 0.006 | **0.995 ± 0.000** |
| Deep Ensemble | 0.360 ± 0.000 | 0.390 ± 0.000 | 0.407 ± 0.000 | **4.294 ± 0.000** | 0.920 ± 0.000 | **0.996 ± 0.000** |
| ConfidNet | 0.313 ± 0.021 | 0.491 ± 0.029 | 0.513 ± 0.027 | 4.818 ± 0.174 | 0.932 ± 0.003 | **0.995 ± 0.000** |
| TrustScore | **0.275 ± 0.023** | **0.565 ± 0.047** | 0.172 ± 0.013 | 4.798 ± 0.280 | **0.933 ± 0.004** | **0.995 ± 0.000** |
| MC Dropout | 0.376 ± 0.023 | 0.443 ± 0.008 | 0.347 ± 0.006 | 5.694 ± 0.641 | 0.921 ± 0.003 | 0.994 ± 0.001 |
| Energy Score | 0.477 ± 0.014 | 0.384 ± 0.013 | 0.236 ± 0.007 | 7.191 ± 0.707 | 0.897 ± 0.004 | 0.992 ± 0.001 |
| MSP | 0.351 ± 0.017 | 0.462 ± 0.004 | 0.353 ± 0.006 | 5.587 ± 0.633 | 0.923 ± 0.003 | 0.994 ± 0.001 |

**Table 13.** Complete results for SVHN - WideResNet-50.

| Method | FPR@95TPR (↓) | AUPR-error (↑) | Wasserstein (↑) | e-AURC (↓) | AUROC (↑) | AUPR-success (↑) |
|---|---|---|---|---|---|---|
| **CALI (Ours)** | **0.297 ± 0.020** | 0.536 ± 0.063 | **0.653 ± 0.017** | 6.266 ± 0.207 | 0.922 ± 0.008 | 0.993 ± 0.000 |
| Deep Ensemble | 0.389 ± 0.000 | 0.352 ± 0.000 | 0.404 ± 0.000 | **3.909 ± 0.000** | 0.912 ± 0.000 | **0.996 ± 0.000** |
| ConfidNet | **0.298 ± 0.019** | 0.530 ± 0.049 | 0.546 ± 0.023 | 5.128 ± 0.142 | **0.931 ± 0.005** | 0.995 ± 0.000 |
| TrustScore | 0.305 ± 0.035 | **0.571 ± 0.064** | 0.134 ± 0.015 | 5.463 ± 0.212 | 0.927 ± 0.008 | 0.994 ± 0.000 |
| MC Dropout | 0.413 ± 0.036 | 0.407 ± 0.025 | 0.322 ± 0.007 | 6.772 ± 0.589 | 0.908 ± 0.003 | 0.993 ± 0.001 |
| Energy Score | 0.509 ± 0.038 | 0.351 ± 0.023 | 0.212 ± 0.025 | 8.489 ± 1.105 | 0.884 ± 0.005 | 0.991 ± 0.001 |
| MSP | 0.388 ± 0.029 | 0.432 ± 0.019 | 0.321 ± 0.007 | 6.618 ± 0.565 | 0.911 ± 0.002 | 0.993 ± 0.001 |

**Table 14.** Detailed sample statistics (↓), grouped by dataset and architecture.

| Dataset | Architecture | $\text{Samples}_{\text{mean}}$ | $\text{Correct}_{\text{mean}}$ | $\text{Correct}_{\text{min}}$ | $\text{Incorrect}_{\text{mean}}$ | $\text{Incorrect}_{\text{min}}$ |
|---|---|---|---|---|---|---|
| CIFAR-10 | ResNet-18 | 5000.0 | 4651.7 | 4072 | 348.3 | 43 |
| CIFAR-10 | VGG-16 | 5000.0 | 4650.8 | 4137 | 349.2 | 130 |
| CIFAR-100 | ResNet-50 | 500.0 | 314.8 | 126 | 185.2 | 27 |
| FashionMNIST | CNN | 6000.0 | 5497.1 | 4200 | 502.9 | 41 |
| MNIST | CNN | 6000.0 | 5707.6 | 5053 | 292.4 | 52 |
| PneumoniaMNIST | ResNet-18 | 2354.0 | 1816.0 | 142 | 538.0 | 4 |
| SVHN | DenseNet-121 | 7325.7 | 7073.5 | 4403 | 252.2 | 61 |
| SVHN | WideResnet-50 | 7325.7 | 7076.6 | 4442 | 249.1 | 93 |

**Table 15. Theoretical Complexity Comparison for Failure Prediction Methods.** $C_{bb}$: Backbone forward pass; $D$: Feature dimension; $d$: Latent dimension ($d \ll D$); $T$: MC-Dropout passes; $K$: Ensemble size; $M$: Support samples per class.

| Method | Training Complexity | Inference Complexity (per sample) |
|---|---|---|
| MSP / Energy Score | $\mathcal{O}(0)$ | $\mathcal{O}(C_{bb})$ |
| ConfidNet | $\mathcal{O}(E_{confidnet} \cdot N \cdot (C_{bb} + C_{head}))$ | $\mathcal{O}(C_{bb} + C_{head})$ |
| MC-Dropout | $\mathcal{O}(E \cdot N \cdot C_{bb})$ | $\mathcal{O}(T \cdot C_{bb})$ |
| Deep Ensembles | $\mathcal{O}(K \cdot E \cdot N \cdot C_{bb})$ | $\mathcal{O}(K \cdot C_{bb})$ |
| Trust Score | $\mathcal{O}(N \cdot D \cdot log\frac{N}{C})$ | $\mathcal{O}(C_{bb} + C \cdot D \cdot logM)$ |
| **CALI (Ours)** | $\mathcal{O}(E_{CWAE} \cdot N \cdot D \cdot d)$ | $\mathcal{O}(C_{bb} + D \cdot d + M \cdot d)$ |

**Table 16.** Empirical runtimes on CIFAR-10 with ResNet-18.

| Method | Training Time | Inference Time |
|---|---|---|
| MSP / Energy | — | 0.03 s |
| ConfidNet | 252 s | 0.09 s |
| MC-Dropout | — | 83.63 s |
| Deep Ensembles | 430.92 s | 0.15 s |
| Trust Score | 1.11 s | 3.51 s |
| **CALI (Ours)** | 100.11 s | 6.63 s |

*Deep Ensembles training time reflects the cost of training four additional models; inference time is measured over all five models. MC-Dropout inference uses 50 forward passes.*