ALSA: Anchors in Logit Space for Out-of-Distribution Accuracy Estimation

Chenzhi Liu
chenzhi.liu@uq.edu.au
Mahsa Baktashmotlagh
m.baktashmotlagh@uq.edu.au
Yanran Tang
yanran.tang@uq.edu.au
Zi Huang
helen.huang@uq.edu.au
Ruihong Qiu
r.qiu@uq.edu.au

School of Electrical Engineering and Computer Science
The University of Queensland
Brisbane, Australia

Abstract

Estimating model accuracy on unseen, unlabeled datasets is crucial for real-world machine learning applications, especially under distribution shifts that can degrade performance. Existing methods often rely on predicted class probabilities (softmax scores) or data similarity metrics. While softmax-based approaches benefit from representing predictions on the standard simplex, compressing logits into probabilities leads to information loss. Meanwhile, similarity-based methods can be computationally expensive and domain-specific, limiting their broader applicability. In this paper, we introduce ALSA (Anchors in Logit Space for Accuracy estimation), a novel framework that preserves richer information by operating directly in the logit space. Building on theoretical insights and empirical observations, we demonstrate that the aggregation and distribution of logits exhibit a strong correlation with the predictive performance of the model. To exploit this property, ALSA employs an anchor-based modeling strategy: multiple learnable anchors are initialized in logit space, each assigned an influence function that captures subtle variations in the logits. This allows ALSA to provide robust and accurate performance estimates across a wide range of distribution shifts. Extensive experiments on vision, language, and graph benchmarks demonstrate ALSA's superiority over both softmax- and similarity-based baselines. Notably, ALSA's robustness under significant distribution shifts highlights its potential as a practical tool for reliable model evaluation. Code has been released at https://github.com/chenzhi-liu/ALSA.

1 Introduction

In real-world machine learning model deployment, the trustworthy and reliable performance of well-trained models largely depends on the in-distribution (ID) assumption. However, the user interactions in online environments frequently introduce diverse and dynamically changing data streams, where the target (test) data may differ significantly from the source (train) data, resulting in substantial model performance degradation [12, 27, 40]. Consequently, accurately estimating model performance in these out-of-distribution (OOD) scenarios is of critical importance.

Performance estimation methods mainly fall into two categories. (1) Using the softmax scores of models. For instance, the Average Confidence (AC) method [16] uses the average of maximum softmax scores to estimate the error, while Average Thresholded Confidence (ATC) [10] further refines the estimation by learning a threshold to better map softmax scores to error predictions. Similarly, Confidence Optimal Transport (COT) [34] employs optimal transport between softmax scores under the assumption of a label distribution shift. (2) Relying on data similarity. These approaches suggest that a model's accuracy is strongly associated with the similarity between the source and target datasets, and quantifying this similarity can lead to accurate performance estimation [5, 6].

Although data similarity-based methods hold promise for model evaluation, they are computationally intensive, as their effectiveness often necessitates the use of multiple models or extensive data augmentation. Furthermore, these approaches are specifically designed for certain data types, including images [5, 6] and graphs [52], making them less suitable as a general framework for evaluating models under distribution shifts. Softmax-based methods are also limited as softmax compresses logits [14], resulting in the loss of magnitude differences that are critical for accurate performance estimation. By operating in logit space instead of softmax outputs a greater amount of information is retained, thereby enhancing accuracy estimation.

Additionally, existing methods overlook class-wise imbalances, which can adversely impact predictions when training distributions are skewed [19, 43]. For example, metrics like average confidence in AC [10] and thresholds in ATC [10] can be misleading when minority classes are scarce in training but prevalent in the target set. Thus, analyzing logits distribution and class-wise performance are both crucial for improving *OOD performance estimation*.

This paper proposes *Anchors in Logit Space for Accuracy estimation* (ALSA), a novel framework for estimating the accuracy of machine learning models on unlabeled datasets under distribution shift. Unlike existing methods that rely on softmax probabilities, ALSA operates directly in the logit space to preserve richer information for more accurate performance estimation. The approach is motivated by the observation that logits tend to aggregate near a specific hyperplane, and certain regions in logit space are more predictive of correct classification outcomes.

To formalize this insight, *anchors* are introduced as reference points in logit space that act as confidence indicators. A probabilistic model is then constructed, where each anchor contributes to the likelihood that a logit vector corresponds to a correct prediction. These contributions are aggregated and passed through a sigmoid function to yield the final probability estimate. ALSA is evaluated on a diverse set of benchmarks, including vision, language, and graph datasets, and demonstrates improved accuracy estimation compared to existing methods. In addition to providing useful insights into model behavior under distribution shifts, it shows strong robustness in scenarios where the marginal distribution of the target variable differs across datasets, which is common in real-world applications.

- Identification and analysis of information loss caused by softmax compression, emphasizing the importance of logit distribution in accuracy estimation.
- Development of ALSA, a novel framework that leverages logit-space distributions to estimate accuracy under distribution shifts without the need for labeled target data.

- Extensive evaluation across vision, language, and graph datasets, as well as diverse model architectures, demonstrating ALSA's superior performance, robustness, and generalization ability in real-world applications.

Prior Work. Our research is related to but different from work in OOD detection [8, 22, 31] and generalization estimation [2, 10, 35, 40, 52]. Prior approaches use softmax scores [10, 22], calibration [31], or dataset similarity [5], often at the cost of reduced generality or informativeness. ALSA addresses these issues by leveraging richer structures in logit space. Detailed related work can be found in Appendix A.

2 Preliminaries and Motivations

2.1 Problem Definition

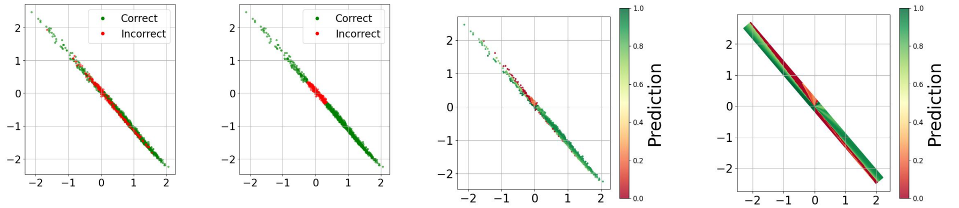
Consider a multi-class classification problem with input space $\mathcal{X} \in \mathbb{R}^d$ and label space $\mathcal{Y} = \{1, 2, \ldots, c\}$. The source and target datasets over $\mathcal{X} \times \mathcal{Y}$ are denoted by D^S and D^T, with corresponding distributions P_S and P_T. A classifier $f : \mathcal{X} \to \mathcal{Z}$ maps inputs to the logit space $\mathcal{Z} \in \mathbb{R}^c$, without softmax normalization. The training and validation sets $\{(\mathbf{x}_{\text{train}}^{(i)}, y_{\text{train}}^{(i)})\}$ and $\{(\mathbf{x}_{\text{val}}^{(i)}, y_{\text{val}}^{(i)})\}$ are drawn from P_S and used to train and validate f. Given an unlabelled test set $D^T = \{\mathbf{x}_{\text{test}}^{(i)}\}_{i=1}^n$ sampled from a different distribution $P_T \neq P_S$, the goal is to estimate the accuracy of f on D^T, denoted by $\text{Acc}(f, D^T)$.

2.2 Band Width of Logit Distribution

Classification models generally follow a similar overarching architecture. The encoder $\text{En}(\cdot)$ transforms an input sample $\mathbf{x}$ into an embedding vector $\mathbf{h} \in \mathbb{R}^m$, which is then mapped to logits $\mathbf{z} \in \mathbb{R}^c$ via a linear transformation. While different models vary in their encoder design, the logits they produce are constrained to a narrow band around a $(c-1)$-dimensional hyperplane in the logit space. The extent of this fluctuation, termed band width, quantifies how far logits deviate from the hyperplane. The following proposition formalizes this constraint:

Proposition 1. *Given a weight matrix $W \in \mathbb{R}^{c \times m}$ with entries $W_{ij} \sim N(0, \sigma_W^2)$, and an embedding vector $\mathbf{h}$ whose components are independent with mean 0 and variance σ_h^2, suppose $\mathbf{b} = 0$ and Xavier initialization is applied, ensuring the variance condition of $\mathbf{h}$. Then, the bandwidth of the logits satisfies:* $\text{Bandwidth} < 4z\sqrt{\frac{1}{m+c}}$, *where z is the z-score of a Gaussian distribution.*

The final linear transformation, given by $\mathbf{z} = W\mathbf{h} + \mathbf{b}$, projects the embedding vector $\mathbf{h}$ from an m-dimensional space to the c-dimensional logit space. These logits are then normalized by the softmax function to yield a probability vector $\mathbf{z}'$ over the classes. A key observation is that the softmax function is invariant to constant shifts; that is, $\text{softmax}(\mathbf{z}) = \text{softmax}(\mathbf{z} + t\mathbf{1})$ for any $t \in \mathbb{R}$, where $\mathbf{1} = (1, 1, \ldots, 1) \in \mathbb{R}^c$. This invariance implies that any gradient component along the direction of $\mathbf{1}$ does not affect the output distribution $\mathbf{z}'$. Consequently, during training with gradient-descent-based optimizers, updates to $\mathbf{z}$ occur only in directions orthogonal to $\mathbf{1}$, leaving its projection onto $\mathbf{1}$ unchanged. As a result, the variation of the logits is confined to the $(c-1)$-dimensional subspace orthogonal to $\mathbf{1}$, effectively restricting them to lie within a narrow band around that hyperplane. The proposition above precisely quantifies the upper bound of this bandwidth. The formal proof is deferred to Appendix B.



(a) True pred. (b) ATC pred. (c) ALSA pred. scatter (d) ALSA pred. surface

Figure 1: Logit distributions and accuracy estimates on the Waterbirds-WILDS validation set using ResNet18. (a) Ground-truth correctness. (b) ATC predictions. (c) ALSA-estimated correctness probabilities. (d) ALSA's prediction surface over the logit region. A 3D example for three classes is shown in Figure 5 in appendix.

2.3 Model Performance against Logit Distribution

According to the proposition stated in Section 2.2, the logits should closely follow a linear structure, as clearly illustrated in Figure 1. In Figure 1(a), each point represents a logit output, with colors indicating whether the predictions of a well-trained model are correct. The likelihood of correct predictions varies depending on the logit's position. For Class 0 (horizontal axis), incorrect samples (in red) are farther from the $y = x$ decision boundary, indicating that even with high confidence after softmax, the probability of incorrect predictions remains elevated. In contrast, for Class 1 (vertical axis), this uncertainty diminishes more rapidly, suggesting that Class 1 requires a lower confidence threshold for nearly perfect classification. This highlights the non-uniformity in the model's ability to distinguish between different classes. Additionally, the distribution of incorrect samples suggests that logits carry information along the all-ones vector. For Class 0, incorrect samples are concentrated at the lower end of the band, whereas for Class 1, they are mostly at the upper end.

Figure 1(b) visualizes the predictions of ATC, which applies a unified softmax score threshold to classify all samples below it as incorrect. However, this approach overlooks variations in the probability of correct predictions across different regions in logit space and fails to account for class-wise differences in classification difficulty.

A similar trend holds in a three-class synthetic scenario (detailed in Appendix C), where logits cluster near a plane in 3D space. This further supports the hypothesis that the position of logits provides additional insight into the probability of correct predictions. To explore this, ALSA models logits' structure and its relationship with correctness.

3 Methodology

3.1 Definition of Anchors

Since all the logits aggregate close to a hyperplane perpendicular to the all-ones vector, it can be expected that when a vector is far from the all-ones vector, certain elements of the vector will dominate. Such a vector will have high confidence after normalization by the softmax function. However, a high-confidence predicted probability does not always indicate a correct prediction. Additionally, the probability of a correct prediction is expected to decay when no element in the predicted probability dominates, indicating that the model is more uncertain among possible classes. Therefore, logits at different positions on the hyperplane correspond to different probabilities of the model making a correct prediction. A set of predefined anchors is used to capture this variation in probability.

An *anchor* is defined as a triplet consisting of a position vector $\mathbf{a} \in \mathbb{R}^c$, a peak influence value $p \in \mathbb{R}$, and a variance $v \in \mathbb{R}$. The set of k anchors is denoted by $\mathcal{A} = \{(\mathbf{a}_i, p_i, v_i)\}_{i=1}^k$, where a_i, p_i, and v_i are three learnable parameters. Each anchor exerts influence on other logits, and the cumulative influences are transformed into a probability of correct prediction using a sigmoid function. For logits with high positive influences, a high probability of correct prediction is expected. Similarly, for those with negative influences, it is highly probable they will result in incorrect predictions.

Anchors Initialization. The k anchor positions are initialized by randomly sampling k logits from the validation set. Peak values are set to a positive or negative constant, depending on whether the corresponding logit represents a correct or incorrect prediction. Variances are drawn from a Gaussian distribution with moderate mean and variance to allow sufficient flexibility. ALSA is not sensitive to the specific choice of initialization, as its objective is to learn anchors that align with the underlying logit distribution. Since anchor parameters are continuously refined during training, different initializations tend to converge to similar distribution given the same data. A comparison between random initialization and sampling anchor positions from the validation set, presented in Appendix D, shows nearly identical prediction surfaces, supporting the robustness of ALSA to initialization choices.

3.2 Influence Function of Anchors

Let the logit vector $\mathbf{z}_i$ from the classifier f correspond to input sample $\mathbf{x}_i$, that is, $\mathbf{z}_i = f(\mathbf{x}_i)$.

In order to quantify the influence that a set of anchors $\mathcal{A}$ exerts on the logit vector $\mathbf{z}_i$, an influence function $\mathrm{Infl}(\cdot,\cdot)$ is required. Ideally, this function should:

- Exert an influence that diminishes as the distance between the logit vector and the anchor increases.
- Have parameters to control the maximum influence and distance an anchor can exert.
- Be differentiable to facilitate optimization.

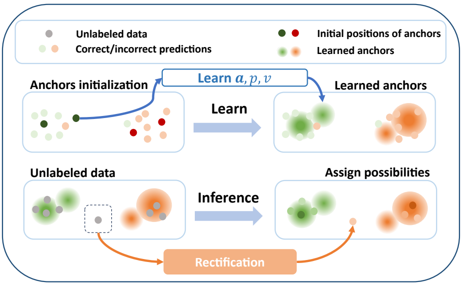


Figure 2: The workflow of ALSA: Anchors are learned from the validation set. During the testing phase, the probability of correctness is assigned to each logit. For regions where the anchor cannot generalize well, rectification is performed.

Given these criteria, a Gaussian-like influence function is proposed for a specific anchor $(\mathbf{a}_j, p_j, v_j)$:

$$\mathrm{Infl}(\mathbf{z}_i, (\mathbf{a}_j, p_j, v_j)) = p_j \exp\left(-v_j^2 \,\mathrm{dist}^2(\mathbf{z}_i, \mathbf{a}_j)\right). \tag{1}$$

The magnitude $|p_j|$ controls the maximum influence an anchor can exert on a logit, and its sign determines the nature of the influence (positive or negative). As the squared distance $\mathrm{dist}^2(\mathbf{z}_i, \mathbf{a}_j)$ increases, the influence decreases rapidly. The variance v_j controls the decay rate of the influence; a smaller v_j value allows the influence to propagate over a greater distance.

The total influence that a logit $\mathbf{z}_i$ receives from the set of anchors $\mathcal{A}$ is defined as:

$$\mathrm{Infl}(\mathbf{z}_i, \mathcal{A}) = \sum_{j=1}^{k} p_j \exp\left(-v_j^2 \,\mathrm{dist}^2(\mathbf{z}_i, \mathbf{a}_j)\right), \tag{2}$$

where the distance function dist is the cosine distance between two vectors, defined as

$$\mathrm{dist}(\mathbf{z}_i, \mathbf{a}_j) = 1 - \frac{\mathbf{z}_i \cdot \mathbf{a}_j}{\|\mathbf{z}_i\|\|\mathbf{a}_j\|}. \tag{3}$$

The bounded nature of cosine distance within [0, 2] helps maintain parameter stability by preventing extreme values, thereby promoting a more stable training. Finally, the probability that the model will give a correct prediction when the logits are $\mathbf{z}_i$ is given by:

$$p_{\text{true}}(\mathbf{z}_i) = \mathrm{Sigmoid}\left(\mathrm{Infl}(\mathbf{z}_i, \mathcal{A})\right). \tag{4}$$

Although the Gaussian-like function satisfies the desired properties, the choice of influence function is not unique. An alternative approach is to use an exponential influence function, which is defined as:

$$\mathrm{Infl}(\mathbf{z}_i, (\mathbf{a}_j, p_j, v_j)) = p_j \exp\left(-v_j^2 \,\mathrm{dist}(\mathbf{z}_i, \mathbf{a}_j)\right). \tag{5}$$

Unlike the Gaussian-like function, which decays quadratically with distance, the exponential function decreases more rapidly when the distance is small but more gradually when the distance is large. To examine its effect, this alternative formulation is also evaluated in the experiments.

3.3 Model Inference

For an unlabeled target dataset with n samples $D^T = \{\mathbf{x}_i\}_{i=1}^n$, the corresponding logits set is denoted as $S^T = \{\mathbf{z}_i\}_{i=1}^n$. The initial predicted accuracy of the evaluated model f on target dataset D^T is provided as the arithmetic mean of the probability of correct prediction for all samples:

$$\hat{Acc}_{\text{train}}(f, D^T) = \frac{1}{n}\sum_{i=1}^{n} p_{\text{true}}(\mathbf{z}_i). \tag{6}$$

Estimation Rectification. Some logits may receive minimal influence from all anchors, indicating they are located far from known regions in logit space. In such cases, the total influence $\mathrm{Infl}(\mathbf{z}_i, \mathcal{A})$ approaches zero, leading to an estimated probability near 50% after applying the sigmoid function. However, this is unreliable, as it reflects high uncertainty rather than an informed prediction. To address this, a simple rectification is applied: if a sample's logits receive insufficient influence, the prediction is set to $p_{\text{true}}(\mathbf{z}_i) = \frac{1}{c}$, where c is the number of classes.

A logit is deemed negligibly influenced if it falls outside a pre-specified confidence interval of an anchor's influence peak, controlled by a confidence interval $\alpha \in (0,1)$. Specifically, for a Gaussian-like influence function, the cut-off is given by $t = pe^{-\left(\mathrm{erf}^{-1}(\alpha)\right)^2}$, where $\mathrm{erf}^{-1}(\cdot)$ is the inverse error function, and for an exponential influence function, it is $t = p(1-\alpha)$. This cut-off, derived from the confidence interval α, is independent of variance, serving as a unified threshold for all anchor influences in the two proposed influence functions. Proof is in Appendix E.

Finally, the predicted accuracy is:

$$\hat{Acc}(f, D^T) = \frac{1}{n} \sum_{i=1}^{n} p_{\text{rec}}(\mathbf{z}_i), \tag{7}$$

where $p_{\text{rec}}(\mathbf{z}_i)$ is defined as:

$$\begin{cases} \text{Sigmoid}\left(\text{Infl}(\mathbf{z}_i, \mathcal{A})\right) & |\text{Infl}(\mathbf{z}_i, \mathcal{A})| \geq t, \\ \frac{1}{c} & \text{otherwise}, \end{cases} \tag{8}$$

where c is the number of classes.

3.4 Learning Objective

The source validation set is denoted as $D = \{(\mathbf{x}_{\text{val}}^{(i)}, y_{\text{val}}^{(i)})\}_{i=1}^{n}$ with n samples. Using the classifier f, the logits for each sample are obtained, resulting in the set $\{(\mathbf{z}_{\text{val}}^{(i)}, y_{\text{val}}^{(i)})\}_{i=1}^{n}$, where $\mathbf{z}_{\text{val}}^{(i)} = f(\mathbf{x}_{\text{val}}^{(i)})$. The predicted label is defined as $\hat{y}_i = \arg\max_j z_{\text{val},j}^{(i)}$, where $z_{\text{val},j}^{(i)}$ is the j-th element of $\mathbf{z}_{\text{val}}^{(i)}$. Additionally, $\delta_i = \mathbb{I}(\hat{y}_i = y_{\text{val}}^{(i)})$ is defined. Thus, the set $\{(\mathbf{z}_{\text{val}}^{(i)}, \delta_i)\}_{i=1}^{n}$ is obtained. The set of anchors with k anchors is $\mathcal{A} = \{(\mathbf{a}_j, p_j, v_j)\}_{j=1}^{k}$.

For each $\mathbf{z}_{\text{val}}^{(i)}$, $p_{\text{true}}(\mathbf{z}_{\text{val}}^{(i)})$ is defined as the probability that the model will make a correct prediction. The Binary Cross Entropy Loss is used as the loss function. By minimizing

$$\mathcal{L} = -\sum_{i=1}^{n} \left[\delta_i \cdot \log\left(p_{\text{true}}(\mathbf{z}_{\text{val}}^{(i)})\right) + (1 - \delta_i) \cdot \log\left(1 - p_{\text{true}}(\mathbf{z}_{\text{val}}^{(i)})\right) \right], \tag{9}$$

where the set of anchors can be learned that reflects the distribution of logits and the probability that the model will give correct predictions when logits are in different regions.

The stopping criteria for the training process must also be considered. The model's true accuracy on D is given by $Acc(f, D) = \frac{1}{n}\sum_{i=1}^{n} \delta_i$. After each training epoch, the predicted accuracy is calculated as $\hat{Acc}_{\text{train}}(f, D) = \frac{1}{n}\sum_{i=1}^{n} p_{\text{true}}(\mathbf{z}_{\text{val}}^{(i)})$. The training is stopped when

$$\left| \hat{Acc}_{\text{train}}(f, D) - Acc(f, D) \right| < \varepsilon. \tag{10}$$

Here, ε is a very small positive number in the range $(0, 1]$; in practice, $\varepsilon = 10^{-5}$ is used for all experiments.

4 Experiments

Datasets. Multiple benchmark datasets across different types of distributions, including vision, language, and graph datasets, are considered in this study. **Natural shift:** Non-simulated shifts from data collection variations are evaluated using CIFAR-10.1 [39], CIFAR-10.2 [33], and ImageNetV2 [40]. **Subpopulation shift:** Subpopulation changes are examined using Waterbirds-WILDS (W-WILDS) [41] for vision tasks and Amazon-WILDS (A-WILDS) [28] for language tasks. **Synthetic shift:** Deliberate corruptions are tested using CIFAR-10C [21], CIFAR-100C [21], and MNIST-M [51]. **Domain shift:** Domain adaptation is evaluated using the Office-31 dataset [29], with data from three domains. **Temporal**

Table 1: MAE of accuracy estimates on vision and language datasets, averaged over three seeds. Bold: best; Italics: second-best among ours. Full table with std in Appendix I.

Dataset	Variant	AC	DoC	IM	ATC	COT	ALSA-E	ALSA-G
CIFAR10	CIFAR-10.1	4.82	4.69	5.70	1.27	3.02	2.83	**1.22**
	CIFAR-10.2	7.97	7.84	8.74	4.59	6.20	**0.61**	*4.19*
	CIFAR-10C	6.57	6.44	7.60	**2.28**	3.18	5.09	4.18
CIFAR100	CIFAR-100C	9.46	7.80	9.02	3.85	**2.41**	7.26	6.56
A-WILD	Subpop1	3.70	2.16	2.14	5.90	3.39	*0.99*	**0.97**
	Subpop2	4.19	2.65	2.65	5.82	4.18	*1.43*	**1.17**
W-WILDS	Subpop	0.78	0.79	0.74	1.55	0.97	**0.32**	*0.54*
ImageNet	ImageNetv2	3.63	2.68	2.85	**0.75**	9.57	4.07	3.22
MNIST	MNIST-M	13.27	13.44	16.29	13.63	4.96	**2.82**	*3.09*
Office-31	domain	38.67	34.47	32.90	41.55	11.40	*8.35*	**7.77**

Table 2: MAE between estimated and true accuracy on ogb-arxiv, averaged over three seeds. AVG. denotes the mean across backbones. Full results with standard deviations are in Appendix I.

Model	AC	DoC	IM	GNNEval	ATC	COT	ALSA-E	ALSA-G
GCN	1.61	2.94	3.89	10.59	19.76	6.64	*1.13*	**0.99**
SAGE	1.31	1.72	2.07	10.42	25.31	5.85	2.21	**1.08**
GAT	3.24	4.16	4.60	13.34	23.89	5.06	*1.73*	**1.01**
GIN	30.09	5.86	7.17	9.54	32.28	30.79	*2.18*	**2.14**
AVG.	9.06	3.67	4.43	10.97	25.31	12.09	*1.81*	**1.30**

shift: Temporal changes are assessed using the ogbn-arxiv dataset [23], focusing on time-evolving data. Further dataset setup details are in Appendix F.

Baselines. The proposed method, ALSA, is compared with the baseline methods. **Average Confidence (AC)** [10] estimates error by averaging maximum softmax confidence. **Difference of Confidence (DoC)** [16] uses the difference in confidence between source and target data. **Importance Re-weighting (IM)** re-weights classifier error using confidence-based slices [3]. **Average Thresholded Confidence (ATC)** [10] estimates error by thresholding softmax confidence values. **Confidence Optimal Transport (COT)** [34] uses optimal transport cost to quantify distributional shifts. **GNNEvaluator (GNNEval)** [52] predicts GNN performance using simulated graph data. ALSA with an exponential influence function is denoted as ALSA-E, while ALSA with a Gaussian-like influence function is denoted as ALSA-G or simply ALSA. Detailed explanations for baselines are in Appendix G and the details of experimental setup can be found in Appendix H.

4.1 Results

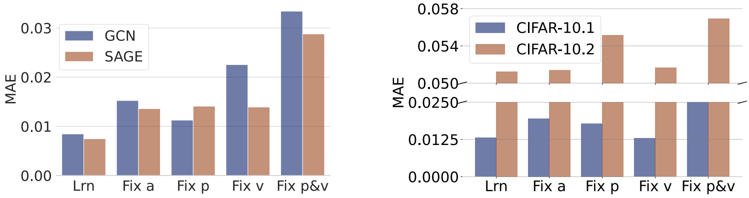
Results across Modalities. Tables 1 and 2 summarize MAE results on vision/language and graph datasets, respectively. The proposed method consistently ranks among the top performers across diverse distribution shifts. It significantly outperforms all baselines on subpopulation benchmarks (e.g., Amazon-WILDS, Waterbirds-WILDS) and achieves strong results under natural shifts (e.g., CIFAR10.1, CIFAR10.2). On synthetic corruptions (e.g., CIFAR-10C, CIFAR-100C), it remains robust across a wide severity range, with Appendix I showing accurate estimates even when true accuracy drops by over 30%. As no method can guarantee correctness under all conditions due to the nature of the task [10], it is notable that while baselines exceed 10% error on at least one dataset, our method consistently remains below this threshold. In the graph domain (ogbn-arxiv), it delivers clear gains across all GNN backbones under temporal shifts.

Correlation with True Accuracy. Table 3 reports R^2 and Pearson correlation (ρ) values on ogbn-arxiv across years. The proposed method consistently achieves the high correlation scores across models, demonstrating its ability to track true performance under distribution shift.

Table 3: R^2 and Pearson correlation (ρ) between estimated and true accuracy. R^2 reflects fit to $y = x$; ρ measures linear correlation. Higher is better. Best results are highlighted.

Model	AC		DoC		IM		GNNEval		ATC		COT		ALSA	
	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ
GCN	0.84	0.96	0.52	0.96	0.25	0.96	-29.28	-0.23	-17.42	0.95	-1.12	0.96	**0.94**	**0.97**
SAGE	0.85	0.97	0.74	0.98	0.61	0.98	-29.10	-0.29	-48.32	**0.98**	-2.13	0.96	**0.87**	**0.98**
GAT	-0.13	0.85	-0.50	0.84	-0.73	0.85	-44.46	-0.25	-29.58	0.55	-0.74	0.75	**0.91**	**0.99**
GIN	-162.85	0.85	-6.40	**0.85**	-8.69	0.76	-19.58	-0.16	-189.22	0.82	-170.22	0.83	**-0.32**	0.46
AVG.	-40.32	0.91	-1.41	0.91	-2.14	0.89	-30.60	-0.23	-71.13	0.83	-43.55	0.88	**0.60**	**0.85**

Additional analyses are provided in the appendix to support the robustness and interpretability of ALSA. Appendix J examines the effects of key hyperparameters, including the confidence interval α and the number of anchors. The analysis shows that while extreme values of the confidence interval α can lead to increased estimation error, ALSA remains robust across a broad and practical range of settings. Performance also stabilizes once a sufficient number of anchors is reached. Appendix K presents a complexity analysis, showing that inference time scales linearly with the number of samples. Inference and training time comparisons are also reported, demonstrating a favorable trade-off between accuracy and efficiency.



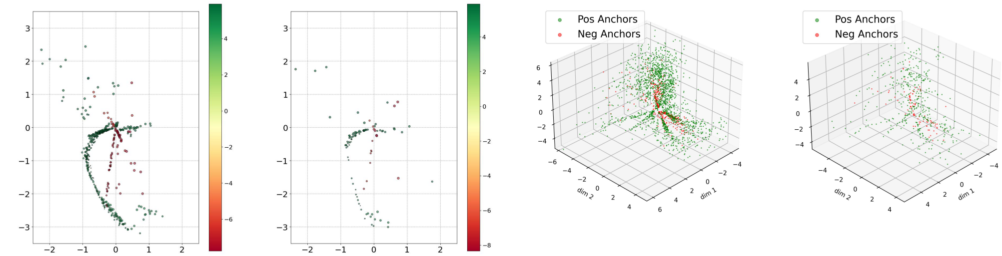
(a) GCN and SAGE for ogbn-arxiv (b) CIFAR-10.1 and CIFAR-10.2

Figure 3: (a) MAE is reported to evaluate the GCN and SAGE models on the ogbn-arxiv dataset by ablating different modules. (b) MAE is reported to evaluate the ResNet18 model on CIFAR-10.1 and CIFAR-10.2 by ablating different modules.

4.2 Ablation Study

The ablation study evaluated the effectiveness of the learnable parameters in ALSA with 1. anchor position (a), representing each anchor's position in logit space; 2. peak influence value (p), indicating the maximum positive or minimum negative influence an anchor can exert; and 3. variance (v), determining the extent of the influence. To test different configurations, these parameters are either fixed after initialization or updated during training. Five experiments are conducted: 1. `Lrn`: all parameters (a, p, and v) are learnable; 2. `Fix a`: anchor positions are fixed; 3. `Fix p`: peak values are fixed; 4. `Fix v`: variances are fixed; and 5. `Fix p&v`: both p and v are fixed.

MAE is utilised as the metric of ablation study. As shown in Figure 3, ablating any component leads to an increase in error, whereas the fully learnable configuration consistently attains the lowest error. Although the severity of the degradation varies, the learnable configuration consistently outperforms the others. The anchor position defines the region being modeled, and making it non-learnable (`Fix a`) restricts the model to focusing on the probability distribution of only specific regions. Fixing any of the peak influence value (`Fix p`), the variance (`Fix v`), and both (`Fix p&v`) hinders the anchors to provide a more fine-grained influence decay, which results in worse generalization of probability distribution.



(a) 500 anchors (b) 100 anchors (c) 2000 anchors (d) 500 anchors

Figure 4: Comparison of learnt anchors from ResNet18 on the Waterbirds-WILDS with 500 vs. 100 anchors (a and b), and on CIFAR-10 remapped to a three-class problem with 2000 vs. 500 anchors (c and d). The color represents the peak values, while the point size reflects the extent of influence. Note that in our experiments, the influence decays according to the cosine distance rather than the Euclidean distance, as might be intuitive from the image.

4.3 Visualization of Anchors

This experiment illustrates the learnt anchors on two settings: the two-class case on Waterbirds-WILDS and the synthetic three-class case, which is remapped to three classes on CIFAR-10, as shown in Figure 4 (a, b) and (c, d), respectively. The anchor positions, their peak values (indicated by color), and variance values (indicated by point size) are shown.

A clear aggregation pattern emerges, where positive and negative anchors (anchors with positive or negative peak values) cluster in distinct regions of logit space. This pattern suggests that logits with similar predictive performance tend to group together. Moreover, the consistent distribution of anchors across different anchor counts indicates that they consistently align with the same underlying distribution in the training data, further verifying the validity of our method.

5 Conclusion

In this work, ALSA, a framework designed to better understand the distribution and probability of correct predictions within the logit space, was proposed. The approach is motivated by the observation that models often exhibit imbalanced prediction performance across different classes—an issue overlooked by existing methods. By analyzing the properties of logits distributions, it is demonstrated that this information can be captured using only a limited number of anchors. Extensive experiments across diverse datasets and domains, including vision, language, and graph tasks, validate the effectiveness of ALSA in accurately estimating model performance. The results show that ALSA not only provide a more robust method for accuracy estimation but also offer deeper insights into model behavior, making them highly applicable for real-world machine learning deployments.

Acknowledgement

This research has been partially supported by Australian Research Council Discovery Projects (DP230101196, DE250100919, CE200100025, and FT230100426).

References

[1] L.C. Andrews. *Special Functions of Mathematics for Engineers*. Online access with subscription: SPIE Digital Library. SPIE Optical Engineering Press, 1998. ISBN 9780819426161. URL https://books.google.com.au/books?id=2CAqsF-RebgC.

[2] Christina Baek, Yiding Jiang, Aditi Raghunathan, and J. Zico Kolter. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. In *NeurIPS*, 2022.

[3] Mayee F. Chen, Karan Goel, Nimit Sharad Sohoni, Fait Poms, Kayvon Fatahalian, and Christopher Ré. Mandoline: Model evaluation under distribution shift. In *ICML*, 2021.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[5] Weijian Deng and Liang Zheng. Are labels always necessary for classifier accuracy evaluation? In *CVPR*, 2021.

[6] Weijian Deng, Stephen Gould, and Liang Zheng. What does rotation prediction tell us about classifier accuracy under varying testing environments? In *ICML*, 2021.

[7] Weijian Deng, Yumin Suh, Stephen Gould, and Liang Zheng. Confidence and dispersity speak: Characterizing prediction matrix for unsupervised accuracy estimation. In *ICML*, 2023.

[8] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *NeurIPS*, 2019.

[9] Sven Elflein, Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. On out-of-distribution detection with energy-based models. *CoRR*, abs/2107.08785, 2021.

[10] Saurabh Garg, Sivaraman Balakrishnan, Zachary Chase Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *ICLR*, 2022.

[11] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *NIPS*, 2017.

[12] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019.

[13] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[14] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep learning*. MIT Press, 2016.

[15] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *ICLR*, 2020.

[16] Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *ICCV*, pages 1134–1144, 2021.

[17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

[18] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[19] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *TKDE*, 2009.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *CoRR*, 2019.

[22] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2016.

[23] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.

[24] Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya R. Gupta. To trust or not to trust a classifier. In *NeurIPS*, 2018.

[25] Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J. Zico Kolter. Assessing generalization of sgd via disagreement. In *ICLR*, 2022.

[26] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[27] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran S. Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

[28] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran S. Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

[29] Piotr Koniusz, Yusuf Tas, and Fatih Porikli. Domain adaptation by mixture of alignments of second-or higher-order scatter tensors. In *CVPR*, 2017.

[30] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[31] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[33] Shangyun Lu, Bradley Nott, Aaron Olson, Alberto Todeschini, Hossein Vahabi, Yair Carmon, and Ludwig Schmidt. Harder or different? a closer look at distribution shift in dataset reproduction. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, volume 5, page 15, 2020.

[34] Yuzhe Lu, Yilong Qin, Runtian Zhai, Andrew Shen, Ketong Chen, Zhenlin Wang, Soheil Kolouri, Simon Stepputtis, Joseph Campbell, and Katia P. Sycara. Characterizing out-of-distribution error via optimal transport. *NeurIPS*, 2023.

[35] John Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *ICML*, 2021.

[36] Preetum Nakkiran and Yamini Bansal. Distributional generalization: A new kind of generalization. *CoRR*, abs/2009.08092, 2020.

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.

[38] PyTorch. Torchvision models. https://pytorch.org/vision/main/models.html, 2024. Accessed: 2024-09-21.

[39] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *CoRR*, 2018.

[40] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.

[41] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *CoRR*, 2019.

[42] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *CoRR*, 2019.

[43] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020.

[44] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[45] Danny Wang, Ruihong Qiu, Guangdong Bai, and Zi Huang. GOLD: graph out-of-distribution detection via implicit adversarial latent generation. In *ICLR*, 2025.

[46] Danny Wang, Ruihong Qiu, Guangdong Bai, and Zi Huang. Text meets topology: Rethinking out-of-distribution detection in text-rich networks. *CoRR*, abs/2508.17690, 2025.

[47] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2018.

[48] Yaodong Yu, Zitong Yang, Alexander Wei, Yi Ma, and Jacob Steinhardt. Predicting out-of-distribution error with the projection norm. In *ICML*, 2022.

[49] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *ICML*, 2016.

[50] Hongjie Zhang, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. In *ECCV*, 2020.

[51] Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E. Gonzalez, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, and Kurt Keutzer. A review of single-source deep unsupervised visual domain adaptation. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):473–493, 2022.

[52] Xin Zheng, Miao Zhang, Chunyang Chen, Soheila Molaei, Chuan Zhou, and Shirui Pan. Gnnevaluator: Evaluating GNN performance on unseen graphs without labels. In *NeurIPS*, 2023.