

Out-of-Known-Safes Detection as a Benchmark for Runtime Monitors

Robbe De Groeve[✉], Laurens Devos[✉], and Mathias Verbeke[✉]

KU Leuven, Belgium

`{robbe.degroeve,laurens.devos,mathias.verbeke}@kuleuven.be`

Abstract. Runtime monitors safeguard ML models during deployment by deciding whether to pass a prediction to a downstream system or trigger a fallback, providing a defense against, e.g., dataset shift. Selecting the best monitor for an application requires setting up a benchmark. This paper identifies a bias in existing benchmarking frameworks rooted in how they handle covariate-shifted samples. These samples span a continuum from close to the training data to severe distortions that cause model failures. Existing benchmarks force monitors to either accept or reject covariate-shifted samples, incentivizing monitors to expand or contract their acceptance region, biasing monitor selection toward either higher false-accept rates, where dangerous predictions slip through, or excessive rejection of valid inputs. We introduce the Out-Of-Known-Safes (OOKS) benchmarking framework, a monitor evaluation methodology which excludes correctly classified covariate-shifted samples from evaluation rather than forcing an arbitrary accept or reject label on ambiguous cases. Experiments on CIFAR-10, CIFAR-100, and ImageNet across nine monitoring techniques demonstrate that benchmark design meaningfully changes monitor rankings.

Keywords: Model runtime monitoring · Benchmarking · Safety.

1 Introduction

Deep learning models are increasingly deployed in real-world systems where prediction errors carry operational, safety, or regulatory consequences. This challenge is particularly acute in safety-critical applications such as autonomous driving [40,4], medical diagnosis [37], and manufacturing [23], where models must continue operating safely even when deployment conditions differ from those seen during development [33,11,3,43].

A **runtime monitor** addresses this challenge by deciding for each prediction whether it should be used or withheld in favor of a fallback [31,11]. From a safety perspective, predictions should be accepted only when the model has been evaluated on that type of input and shown to behave safely [5]; predictions outside this known-safe operating region should instead be rejected.

Runtime monitors encounter three broad categories of inputs. *In-distribution (ID)* samples originate from the training distribution and split into correctly and incorrectly classified examples. *Semantically shifted (S-OOD)* samples lie outside the model’s intended task, while *covariate-shifted (C-OOD)* samples preserve task semantics but differ through changes such as blur, noise, or domain shift. The central challenge of runtime monitoring is deciding which of these samples should be accepted or rejected.

Historically, these categories have been studied separately [36]. Conventional *out-of-distribution* (OOD) detection focuses on distinguishing in-distribution data from out-of-distribution inputs, primarily targeting semantic shifts [21,24,43]. In contrast, *classification with a reject option (CRO)* [17] evaluates a classifier’s ability to abstain from uncertain predictions on in-distribution data [9,14,34].

Recent work has begun combining these perspectives. Examples include *full-spectrum OOD (FS-OOD)* [44], *G-ODIN* [24], *model-specific OOD (MS-OOD)* [2], and *selective classification under OOD data (SCOD)* [42,12]. While these frameworks all evaluate monitors across mixtures of ID, S-OOD, and C-OOD samples, they fundamentally disagree on how correctly classified covariate-shifted samples should be treated (Figure 1).

Covariate shifts form a continuum ranging from benign variations to severe corruptions, making it unrealistic for a runtime monitor to determine precisely where expected model generalization ends and unsafe behavior begins. Existing benchmarking frameworks nevertheless require these correctly classified C-OOD samples to be either accepted (MS-OOD) or rejected (G-ODIN). Both choices introduce optimization bias: rewarding acceptance encourages more permissive monitors with higher false-accept rates, while rewarding rejection favors overly conservative monitors with unnecessary false rejections.

This paper argues that correctly classified covariate-shifted samples should not influence runtime monitor evaluation. We introduce the Out-Of-Known-Safes (OOKS) benchmarking framework, which deliberately excludes these ambiguous samples from scoring (Figure 1, rightmost panel (g)). Instead of rewarding monitor behavior in this gray area, OOKS encourages deployment-

	(a) OOD	(b) CRO	(c) FS-OOD	(d) G-ODIN	(e) SCOD	(f) MS-OOD	(g) OOKS (ours)
Accept	ID ✓ ID ✗	ID ✓	ID ✓ ID C-OOD ✓ C-OOD ✗	ID ✓ ID ✗	ID ✓	ID ✓ C-OOD ✓	ID ✓ C-OOD ✓
Reject	S-OOD	ID ✗	S-OOD	C-OOD ✓ C-OOD ✗ S-OOD	ID ✗ S-OOD	ID ✗ C-OOD ✗ S-OOD	C-OOD ✓ ID ✗ C-OOD ✗ S-OOD

Fig. 1: Comparison of the accept/reject behavior across different runtime monitoring frameworks: conventional OOD, CRO [9], FS-OOD [44], G-ODIN [24], SCOD [42,12], MS-OOD [2] and OOKS. ID, S-OOD, and C-OOD denote in-distribution, semantically shifted, and covariate-shifted samples; ✓ and ✗ indicate correctly and incorrectly classified samples. The primary divergence lies in how frameworks label correctly and incorrectly classified covariate-shifted samples, C-OOD ✓ and C-OOD ✗ respectively. With our proposed OOKS framework, we argue that C-OOD ✓ should be excluded from monitor evaluation to prevent optimization biases that yield dangerous *false accepts*.

specific covariate variations to be incorporated directly into model development whenever they are expected during operation.

We make the following contributions:

1. We introduce the OOKS benchmarking framework to evaluate, tune, and compare runtime monitors.
2. We empirically show that benchmark design meaningfully affects monitor selection, and trace this bias to differing assumptions about correctly classified covariate-shifted samples.

2 Background & related work

This paper focuses on runtime monitors for image classification.¹ We consider classifiers f that map inputs $x \in X$ to labels $Y = \{1, \dots, C\}$, and runtime monitors g that score inputs based on f 's activations and outputs (e.g., softmax probabilities [21] or feature embeddings [28]). Inputs with $g(x, f) \leq \tau$ are *accepted* and passed through; inputs with $g(x, f) > \tau$ are *rejected* and routed to a fallback. A higher threshold τ is therefore more permissive.

Monitor evaluation arises when candidate monitors must be ranked and selected, whether for hyperparameter tuning or for choosing among entirely different monitoring strategies. This evaluation is performed on a dataset covering the range of inputs the monitor will encounter at deployment. Given a fixed

¹ The underlying ideas are not limited to classification and can be extended to other tasks such as regression or object detection and segmentation.

evaluation dataset, the final monitor ranking is entirely determined by what the benchmarking framework rewards; a benchmark that rewards the wrong behavior will consistently select a sub-optimal monitor.

2.1 Benchmarking monitors: choosing ground-truth monitor labels

A runtime monitor can be viewed as a binary classifier: it either accepts or rejects an incoming example. To evaluate a monitor, a benchmarking framework assigns ground-truth *monitor labels*. For each input x_i in an evaluation dataset, a *monitor label*² $z_i \in \{\text{accept}, \text{reject}\}$ represents the desired monitor action. Given a threshold τ , the monitor predicts $\hat{z}_i = \mathbb{1}[g(x, f) > \tau]$ (where $\hat{z}_i = 1$ denotes *reject*), and performance is measured by comparing $\hat{z}_i$ to z_i across the evaluation set using a chosen performance metric. The resulting ranking is clearly determined by the specific ground-truth rules a benchmarking framework enforces. Figure 1 gives an overview of how each framework labels the different categories of examples.

2.2 The safety perspective: maintaining known-safe operation

The goal of a runtime monitor is to decide whether a classifier’s prediction should be passed through to the downstream system or filtered out. From a safety-critical perspective, runtime monitoring can be understood as maintaining a *known-safe operating zone* around the monitored classifier.

Under this view, a prediction should be passed through only when it belongs to a scenario for which the model is known to behave safely. We call such a prediction *known safe*: the model’s performance on that type of scenario has been thoroughly evaluated, and the observed performance satisfies task-specific safety criteria. An ideal monitor g accepts known-safe cases and otherwise triggers:

$$g(x) = \begin{cases} 0 & \text{if } x \text{ belongs to a known-safe scenario,} \\ 1 & \text{otherwise.} \end{cases}$$

The corresponding trigger conditions cover both *known unsafe* scenarios (where the model is known to fail) and *unevaluated* scenarios for which no sufficient evidence about model behavior is available. Samples without such evidence fall outside the known-safe operating zone, even if the model might happen to guess correctly on some of them.

2.3 Relation to existing benchmarking frameworks

The growing number of runtime monitoring techniques has created a need for benchmarks that evaluate monitors across realistic deployment scenarios. Existing frameworks differ mainly in how they categorize deployment sample scenarios and how they expect the monitor to behave within those scenarios.

² This is called the model-specific acceptance/rejection (MS-A/R) in MS-OOD [2].

Deployment sample scenarios are commonly organized into in-distribution (ID) data, semantically-shifted out-of-distribution (S-OOD), and covariate-shifted out-of-distribution (C-OOD) samples [29,32]. In-distribution data matches the development layout, semantic shifts contain content outside the intended task scope, and covariate shifts preserve core task semantics while introducing feature corruptions like noise, blur, or illumination changes.

Conventional and Full-Spectrum OOD detection. Conventional out-of-distribution detection separates clean in-distribution data from out-of-distribution data. While well-suited to capturing semantic shifts, it does not directly capture a classifier’s mistakes on clean data, nor does it effectively handle the continuum of covariate corruptions [15]. Several works argue that runtime monitors should not reject covariate shifts because a robust model is expected to generalize to benign variations [44,1,22]. While the full-spectrum (FS-OOD) benchmarking framework [44] incorporates covariate-shifted data, its metrics remain primarily distribution-centric: they evaluate where a sample originates rather than whether the model is safe on it, creating a fundamental disconnect from the actual deployment objective.

For example, OpenOOD v1.5 [45] is a popular evaluation toolkit in this space. To maintain consistency with these established evaluation setups, our concrete OOKS evaluation setup uses the standard dataset selections and image corruption variants popularized by benchmarks like OpenOOD v1.5. However, the core novelty of OOKS lies entirely in its distinct evaluation methodology.

Classification with a reject option. Classification with a reject option, also known as selective classification, rejects predictions estimated to exceed a risk threshold while maximizing coverage [14,13]. However, classical formulations usually assume that the deployment data perfectly matches the clean development distribution. Extensions evaluating selective classification under distribution shifts exist [26], but they do not explicitly address the safety implications of diverse, unvetted covariate-shifted scenarios.

2.4 Resolving the covariate shift dilemma: MS-OOD and G-ODIN

Recent literature suggests that the ongoing dilemma regarding how to handle covariate-shifted examples can be resolved by evaluating correctness and distribution shifts simultaneously [2]. Specifically, combining out-of-distribution detection and correctness prediction ensures that misclassified covariate shifts (C-OOD ✗) are cleanly categorized as unsafe and targeted for rejection. However, there is no agreement on how to treat correctly-classified covariate shift (C-OOD ✓). There are two prominent options in existing literature:

- **MS-OOD (Model-Specific OOD)** argues that because the underlying model classifies C-OOD ✓ samples correctly, the monitor should be rewarded for accepting them [2]. The target monitor label for these samples is thus *accept*.
- **G-ODIN** conservatively rejects all out-of-distribution samples regardless of model accuracy; the target monitor label for C-OOD ✓ samples is thus *reject*.

3 Out-of-Known-Safes (OOKS) Benchmarking Framework

Depending on what a benchmarking framework rewards, different monitors will be selected as the best-performing option during hyperparameter tuning or architectural selection. This paper contends that correctly classified covariate-shifted samples reflect observed generalization rather than verified, systemic safety. Because forcing a hard accept or reject label on this gray zone introduces arbitrary optimization preferences, our proposed Out-Of-Known-Safes (OOKS) framework explicitly excludes these benign covariate shifts from scored evaluation. This design choice keeps monitor selection tightly focused on the operating boundaries where safety signals are unambiguous.

3.1 Why labeling correctly classified covariate shifts as *accept* biases monitor selection

Assigning a ground-truth label of $z_i = \text{accept}$ to correctly classified covariate-shifted samples, as the model-specific out-of-distribution (MS-OOD) framework does, creates a systematic bias toward monitors with higher false-accept rates. This occurs because the decision threshold τ is artificially driven toward more lenient values. For any runtime monitor g and threshold τ , the rates at which correctly classified covariate shifted (C-OOD ✓) samples and incorrectly classified covariate shifted (C-OOD ✗) samples are accepted are fundamentally driven by the same underlying score distribution. Incentivizing a high acceptance rate for C-OOD ✓ inevitably forces τ upward, dragging along the unintended acceptance of dangerous C-OOD ✗ failures. Consequently, a benchmarking framework that rewards high acceptance of C-OOD ✓ therefore systematically favors monitors that are more permissive on covariate-shifted data as a whole, including on the incorrectly classified samples that should be rejected.

OOKS operates on a foundational safety principle: *a monitor should accept a prediction only when the model has been rigorously evaluated on that specific type of input and its behavior has been verified as safe*. This translates into the acceptance of ID ✓ samples while ID ✗, C-OOD ✗ and S-OOD samples would need to be rejected.

Samples belonging to C-OOD ✓ occupy an ambiguous middle ground. Although the base model happens to classify them correctly, they originate from shifted environmental conditions whose safety has not necessarily been established. They represent opportunistic generalization rather than verified safety. Some may be mild, benign variations close to the in-distribution design domain, while others might sit perilously close to catastrophic model breakdowns or semantic anomalies. Because benchmarking frameworks provide no oracle boundary to separate safe generalization from impending failure, imposing a hard ground-truth label introduces severe trade-off biases:

- Labeling C-OOD ✓ as *accept* (the MS-OOD paradigm) is too lenient from a safety perspective. These inputs sit outside the verified operating zone

of the model. Rewarding a monitor for expanding its acceptance boundary over unvetted domains increases the probability of false accepts on highly corrupted, misclassified samples nearby.

- Conversely, labeling C-OOD ✓ as *reject* (the G-ODIN paradigm) is overly conservative. A model that successfully navigates mild environmental variations is executing exactly what robust generalization is intended to achieve. Treating these successes as failures penalizes system availability and favors overly restrictive monitors.

OOKS resolves these opposing tendencies by leaving C-OOD ✓ samples entirely unscored. While passing such an instance during deployment is not a failure in isolation (since the model output is correct), the benchmark should not actively penalize a monitor for filtering it out if a stricter stance is required to guarantee the rejection of nearby safety violations.

We illustrate this ambiguity and the resulting threshold bias with a Gaussian-based monitoring technique [28] fitted on CIFAR-10 [27]. We compare dataset-wide monitor score distributions over the full, clean in-distribution test set versus covariate-shifted test sets corrupted by Gaussian noise at two distinct severities. Figure 2 shows representative single-image examples of these visual corruptions, while Figure 3 tracks the macro-level dataset score distributions.



Fig. 2: CIFAR-10 examples under increasing Gaussian-noise corruption strength.

For each dataset score distribution, we can construct the mathematically optimal accept/reject threshold corresponding to the three different evaluation methodologies for handling correctly classified covariate-shifted samples: MS-OOD, G-ODIN, or our proposed OOKS.

As shown in Figure 3 (left), when corruptions are mild, optimizing for a distribution-centric framework like G-ODIN establishes a highly conservative threshold (τ_1) that rejects viable samples. In this regime, the optimal thresholds for MS-OOD and OOKS naturally align. However, under heavy corruptions (right), the MS-OOD metric skews the threshold drastically toward leniency (τ_2) to capture the heavily shifted C-OOD ✓ population, resulting in a dangerous surge of false accepts on failed model predictions. By omitting these ambiguous samples from the objective function, OOKS maintains a stable safety boundary that adapts to corruption without swinging toward dangerous permissiveness or crippling over-conservatism.

These opposing effects demonstrate why forcing a hard accept or reject label for correctly classified covariate-shifted samples introduces systematic bias.

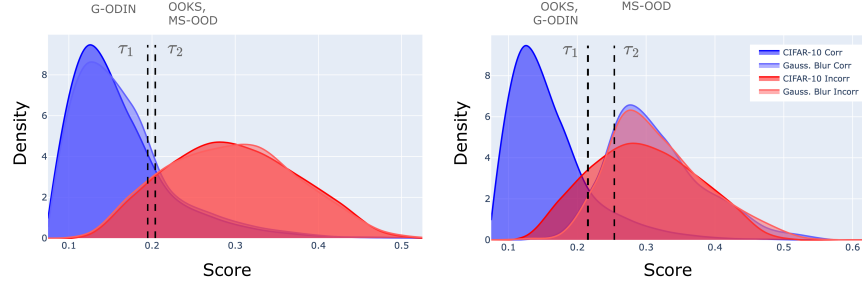


Fig. 3: Score distribution for ID ✓ (dark blue), ID ✗ (dark red), C-ODD ✓ (light blue), C-ODD ✗ (light red). Under mild corruption (left), G-ODIN yields a stricter threshold (τ_1) than MS-ODD and OOKS, whose optimal thresholds coincide. Under stronger corruption (right), the MS-ODD setting pulls the threshold in a lenient direction (τ_2), increasing false accepts relative to the two other settings. Threshold τ_1 is preferred by OOKS, while MS-ODD ranks τ_2 better, despite its higher false-accept rate and potential safety violations.

OOKS ignores these samples altogether, allowing the benchmark to avoid being too conservative or too permissive.

3.2 A ROC argument for excluding correctly classified covariate shifts

The bias can be made precise using a simple probabilistic argument. For any threshold, define the true accept rate as the fraction of correctly-classified samples that pass through, and the false accept rate as the fraction of incorrectly-classified ones that do. The curve tracing these two rates against each other as the threshold varies is a ROC curve, and its area has a natural interpretation: AUROC is the probability that the monitor assigns a more in-distribution score to a correctly-classified sample than to an incorrectly-classified one. Unless $\text{AUROC} = 1$, the score distributions overlap, meaning any threshold that accepts some correctly-classified C-ODD samples will inevitably also accept some incorrectly-classified ones.

This creates a structural problem for MS-ODD. By including C-ODD ✓ with an *accept* label, its objective rewards higher true accept rates in a way the OOKS objective does not. Since both the true and false accept rates are non-decreasing in the threshold, this pushes the MS-ODD-optimal threshold upward, and with it the false accept rate on unsafe C-ODD samples. The same logic applies to monitor selection: MS-ODD systematically favors monitors with higher true accept rates on C-ODD ✓, which by the C-ODD ROC structure implies higher false accept rates on C-ODD ✗. The magnitude of this bias scales directly with score overlap: it disappears only when $\text{AUROC} = 1$, and is largest when monitor is not predictive for C-ODD correctness.

4 Experiments

This paper answers the following experimental questions:

- Q1. Do the benchmarking frameworks rank different runtime monitors differently, resulting in a different selected monitor?
- Q2. Can a structural bias in failure and unavailability rates for the selected monitors be observed between MS-OOD, G-ODIN and OOKS?

4.1 Experimental setup

Datasets. We evaluate on three widely used vision datasets: CIFAR-10 [27], CIFAR-100 [27], and ImageNet1k [8]. For each benchmark setup, the original training and validation data define the in-distribution (ID) setting. We use the following additional datasets for covariate-shifted out-of-distribution (C-OOD) scenarios:

CIFAR-10 & CIFAR-100: The covariate-shifted scenarios use six corruption families: Gaussian & impulse noise, Gaussian & motion blur, and brightness & contrast shift. We follow the standard corruption-severity protocol from Hendrycks and Dietterich [20] to test monitor behavior under varying covariate-shift strengths.

ImageNet1k: The covariate-shifted scenarios are built from ImageNet-C, ImageNet-R [19], and ImageNet-V2 [35]. ImageNet-C uses the same corruptions as CIFAR, but combines them to test general corruption detection. ImageNet-R captures rendition and style shifts, while ImageNet-V2 captures distribution refresh and re-sampling effects.

Monitored classifiers. We monitor three classifiers, one per dataset. We base all three models on the same ResNet-50 [16] backbone. The model weights are obtained from Hugging Face [10]. The model cards report test accuracies of 94.65% on CIFAR-10 [6], 80.93% on CIFAR-100 [7], and ImageNet top-1/top-5 accuracies of 80.38%/94.6% at 224px evaluation resolution [39].

Runtime monitoring techniques. We evaluate post-hoc monitoring techniques that use confidence, logits, features, and gradients. Confidence-based monitors include entropy (Ent) and maximum softmax probability (Soft) [21]. Logit-based monitors include maximum logit (Logit) [18] and energy scoring (Energy) [30]. Feature-space monitors include KNN [38], Gaussian discriminant scoring (Gauss) [28], Virtual-logit Matching (ViM) [41], and GradNorm (Grad) [25].

All monitors are fit using only in-distribution training data. We split the original training set into ID_{Fit} and ID_{Tune} : ID_{Fit} is used to estimate monitor-specific quantities such as feature statistics or nearest-neighbor reference sets, while ID_{Tune} is used for threshold calibration. Thresholds are set to achieve $TNR(ID_{Tune} \checkmark) = 95\%$, ensuring a shared operating point across all monitors.

Evaluation metrics. For OOKS, the monitor labels are accept for ID ✓ and reject for ID ✗, C-ODD ✗, and S-ODD samples, while C-ODD ✓ samples are excluded from the metric. For the baseline benchmarking frameworks (MS-ODD and G-ODIN), we use the framework-specific accept/reject labels shown in Figure 1; this mirrors the model-specific accept/reject labeling idea used by MS-ODD [2], but changes the label mapping according to the framework under evaluation.

Let $A_\tau(S) = \{x \in S \mid g(x, f) \leq \tau\}$ denote accepted samples and $R_\tau(S) = \{x \in S \mid g(x, f) > \tau\}$ denote rejected samples for a dataset S . We treat rejection as the positive monitor decision and acceptance as the negative monitor decision. At the tuned threshold τ , we report:

$$\begin{aligned} \text{TNR}(\text{ID } \checkmark) &= \frac{|A_\tau(\text{ID } \checkmark)|}{|\text{ID } \checkmark|}, & \text{TPR}(\text{ID } \times) &= \frac{|R_\tau(\text{ID } \times)|}{|\text{ID } \times|}, \\ \text{TPR}(\text{C-ODD } \times) &= \frac{|R_\tau(\text{C-ODD } \times)|}{|\text{C-ODD } \times|}, & \text{TPR}(\text{S-ODD}) &= \frac{|R_\tau(\text{S-ODD})|}{|\text{S-ODD}|}. \end{aligned}$$

These quantities correspond respectively to true-negative rate on known-safe in-distribution samples and true-positive rates on known-unsafe or unevaluated samples. Equivalently, $1 - \text{TNR}(\text{ID } \checkmark)$ is the false-reject rate on known-safe samples, while $1 - \text{TPR}(\text{ID } \times)$, $1 - \text{TPR}(\text{C-ODD } \times)$, and $1 - \text{TPR}(\text{S-ODD})$ are false-accept rates on samples that should be filtered out under OOKS.

In the covariate-shift-focused experiments below, we do not separately evaluate monitoring performance on semantically shifted samples, because the benchmarking frameworks considered here differ only in how they treat covariate-shifted scenarios. A comprehensive monitor evaluation should nevertheless also include semantic shifts.

4.2 Results

Q1: Monitor ranking and technique selection. To answer Q1, we evaluate whether the choice of benchmarking framework (MS-ODD, G-ODIN, or OOKS), specifically how correctly classified covariate-shifted samples are handled, leads to different rankings between runtime monitors and alters which technique is ultimately selected.

The empirical results confirm that benchmarking framework choice affects monitor selection. As shown in Table 1, OOKS, MS-ODD, and G-ODIN disagree on the best monitor in 7 of the 15 covariate-shift cases.

Compared to OOKS, MS-ODD selects a different monitor in 5 covariate-shifted cases across CIFAR-10, CIFAR-100, and ImageNet1k, while G-ODIN diverges in 2. This demonstrates that a benchmarking framework’s monitor labels for covariate shifts materially affect monitor selection.

Q2: Differences in failure and unavailability rates between MS-ODD, G-ODIN, and OOKS. To address Q2, we analyze whether these different

Table 1: Runtime monitors selected by MS-ODD, G-ODIN, and OOKS (*ours*) for different covariate-shift cases. Cases highlighted with * or † have different selected runtime monitors for MS-ODD and G-ODIN compared to OOKS, respectively.

CIFAR-10				
C-ODD dataset	MS-ODD	G-ODIN	OOKS	
Brightness	Ent	Ent	Ent	
Contrast	Soft	Logit	Soft	†
Gaussian blur	VIM	VIM	VIM	
Gaussian noise	VIM	Gauss	Gauss	*
Impulse noise	VIM	Gauss	Gauss	*
Motion blur	KNN	VIM	VIM	*

CIFAR-100				
C-ODD dataset	MS-ODD	G-ODIN	OOKS	
Brightness	KNN	KNN	KNN	
Contrast	KNN	VIM	KNN	*
Gaussian blur	VIM	VIM	VIM	
Gaussian noise	VIM	VIM	VIM	
Impulse noise	VIM	VIM	VIM	
Motion blur	Gauss	VIM	VIM	*

ImageNet1k				
C-ODD dataset	MS-ODD	G-ODIN	OOKS	
ImageNet-C	Ent	Ent	Ent	
ImageNet-R	VIM	VIM	VIM	
ImageNetv2	GradNorm	Ent	Ent	†

rankings are indicative of an underlying bias toward higher failure rates (more false accepts) or higher system unavailability (more false rejects). The failure and unavailability rates of the runtime monitors selected by MS-ODD, G-ODIN, and OOKS are given in Table 2.

For all 5 cases where MS-ODD selected a different top monitor in Q1, we observe a shift in failure and unavailability rate: the MS-ODD-selected monitor achieves a lower unavailability rate at the direct cost of a higher failure rate. This matches theoretical expectations, as prioritizing the acceptance of C-ODD ✓ samples structurally biases the selection toward looser thresholds or monitors that favor throughput over safety. Notably, this lower unavailability rate factors in all correct predictions on the covariate-shifted dataset, which includes “lucky guesses” of the classifier alongside true generalization.

For the 2 cases where G-ODIN selected a different top monitor in Q1, the G-ODIN-selected monitoring technique has both a higher failure rate and a higher unavailability rate. In these two cases, penalizing the monitor for not

Table 2: Covariate-shift monitor-selection system-level consequences under alternative treatments of correctly classified covariate-shifted samples according to MS-OD, G-ODIN and OOKS (*ours*). For each covariate-shifted scenario, the table reports the resulting failure and unavailability rates of the selected monitors. Best (lowest) values are in bold.

CIFAR-10						
C-ODD dataset	Failure rate			Unavailability rate		
	MS-ODD	G-ODIN	OOKS	MS-ODD	G-ODIN	OOKS
Brightness	0.3988	0.3988	0.3988	0.0592	0.0592	0.0592
Contrast	0.378	0.3851	0.378	0.0682	0.0866	0.0682
Gauss. blur	0.1334	0.1334	0.1334	0.2438	0.2438	0.2438
Gauss. noise	0.1168	0.1083	0.1083	0.4171	0.4434	0.4434
Impulse noise	0.1652	0.1578	0.1578	0.347	0.361	0.361
Motion blur	0.409	0.4083	0.4083	0.3261	0.3319	0.3319

CIFAR-100						
C-ODD dataset	Failure rate			Unavailability rate		
	MS-ODD	G-ODIN	OOKS	MS-ODD	G-ODIN	OOKS
Brightness	0.4711	0.4711	0.4711	0.0838	0.0838	0.0838
Contrast	0.315	0.3257	0.315	0.1212	0.134	0.1212
Gauss. blur	0.29	0.29	0.29	0.1776	0.1776	0.1776
Gauss. noise	0.1634	0.1634	0.1634	0.3542	0.3542	0.3542
Impulse noise	0.1062	0.1062	0.1062	0.3731	0.3731	0.3731
Motion blur	0.3818	0.3676	0.3676	0.2925	0.3072	0.3072

ImageNet1k						
C-ODD Dataset	Failure rate			Unavailability rate		
	MS-ODD	G-ODIN	OOKS	MS-ODD	G-ODIN	OOKS
ImageNet-C	0.293	0.293	0.293	0.1366	0.1366	0.1366
ImageNet-R	0.2674	0.2674	0.2674	0.2575	0.2575	0.2575
ImageNetv2	0.5867	0.5662	0.5662	0.4133	0.4338	0.4338

accepting correctly classified covariate-shifted samples resulted in a clear Pareto suboptimal monitor selection.

OOKS consistently selects the monitoring technique with the lowest overall failure rate. By ignoring C-ODD ✓ samples, our benchmarking framework eliminates the trade-off, ensuring that monitor selection is driven purely by a safety-oriented criterion without penalizing benign deployment shifts.

5 Conclusion

Runtime monitors are only useful in safety-critical machine learning systems if their evaluation rewards the behavior that such systems actually require: accepting known-safe predictions while filtering out known-unsafe and unevalu-

ated ones. This paper argued that existing benchmarking frameworks’ handling of correctly classified covariate-shifted samples leads to biased runtime monitor comparison. We introduced OOKS as a safety-oriented benchmarking framework that avoids penalizing monitors for rejecting or accepting these types of samples.

Our empirical results show that this distinction matters in practice. Changing the benchmarking framework changes monitor rankings and changes which monitoring technique is selected. In particular, MS-ODD tends to favor monitors with lower unavailability but higher failure rates. OOKS instead consistently selects monitors with lower failure rates, giving a cleaner safety signal for deployment settings in which false accepts are more costly than unnecessary fallbacks. The G-ODIN-based selection results further show that benchmark assumptions can interact with monitor score distributions in non-intuitive ways, sometimes worsening both failure and unavailability.

Future work should study this interaction more explicitly through the severity of covariate shift. In this paper, we used mixed corruption severities to approximate unstructured real-world deployment conditions, where mild and severe shifts can occur together. However, separating mild from severe covariate shifts would make it possible to test the intuition underlying the different benchmarking framework choices: mild covariate shift may often resemble recoverable generalization, whereas severe covariate shift is more likely to fall outside the known-safe operating zone. Extending the Q1 and Q2 analyses with severity-stratified experiments could therefore clarify when MS-ODD, G-ODIN, and OOKS select different monitors, and why the G-ODIN cases in particular deviate from the expected safety-availability tradeoff.

References

1. Ahmed, F., Courville, A.: Detecting semantic anomalies. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 3154–3162 (2020)
2. Averly, R., Chao, W.L.: Unified out-of-distribution detection: A model-specific perspective. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1453–1463 (2023)
3. Bayram, F., Ahmed, B.S., Kassler, A.: From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems* **245**, 108632 (2022)
4. Chib, P.S., Singh, P.: Recent Advancements in End-to-End Autonomous Driving Using Deep Learning: A Survey. *IEEE Transactions on Intelligent Vehicles* **9**(1), 103–118 (2024)
5. Chow, C.K.: On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theory* **16**, 41–46 (1970)
6. Dadalto, E.: Resnet50_cifar10 (2024), https://huggingface.co/edadaltocg/resnet50_cifar10
7. Dadalto, E.: Resnet50_cifar100 (2024), https://huggingface.co/edadaltocg/resnet50_cifar100
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. Ieee (2009)
9. El-Yaniv, R., Wiener, Y.: On the foundations of noise-free selective classification. *Journal of Machine Learning Research* **11**(53), 1605–1641 (2010)
10. Face, H.: Hugging face hub (2021)
11. Ferreira, R.S., Guérin, J., Delmas, K., Guiochet, J., Waeselynck, H.: Safety monitoring of machine learning perception functions: A survey. *Computational Intelligence* **41**(2), e70032 (2025)
12. Franc, V., Paplham, J., Prusa, D.: Scod: From heuristics to theory. In: Computer Vision – ECCV 2024. pp. 424–441 (2025)
13. Galil, I., Dabbah, M., El-Yaniv, R.: What can we learn from the selective prediction and uncertainty estimation performance of 523 imagenet classifiers? In: ICLR (2023)
14. Geifman, Y., El-Yaniv, R.: Selective classification for deep neural networks. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
15. Guerin, J., Delmas, K., Ferreira, R., Guiochet, J.: Out-of-distribution detection is not all you need. Proceedings of the AAAI Conference on Artificial Intelligence **37**(12), 14829–14837 (2023)
16. 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)
17. Hendrickx, K., Perini, L., Van der Plas, D., Meert, W., Davis, J.: Machine learning with a reject option: A survey. *Machine Learning* **113**(5), 3073–3110 (2024)
18. Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Mostajabi, M., Steinhardt, J., Song, D.X.: Scaling out-of-distribution detection for real-world settings. In: International Conference on Machine Learning (2022)
19. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8340–8349 (2021)

20. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (2019)
21. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: International Conference on Learning Representations (2017)
22. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15262–15271 (2021)
23. Hesabi, H., Nourelfath, M., Hajji, A.: A deep learning predictive model for selective maintenance optimization. *Reliability Engineering & System Safety* **219**, 108191 (2022)
24. Hsu, Y.C., Shen, Y., Jin, H., Kira, Z.: Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
25. Huang, R., Geng, A., Li, Y.: On the Importance of Gradients for Detecting Distributional Shifts in the Wild. In: Advances in Neural Information Processing Systems (2021)
26. Kim, J., Koo, J., Hwang, S.: A unified benchmark for the unknown detection capability of deep neural networks. *Expert Systems with Applications* **229**, 120461 (2023)
27. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
28. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*. vol. 31. Curran Associates, Inc. (2018)
29. Liang, H., Peng, L., Sun, J.: Selective Classification Under Distribution Shifts. *Transactions on Machine Learning Research* (2024)
30. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based Out-of-distribution Detection. In: *Advances in Neural Information Processing Systems*. vol. 33, pp. 21464–21475. Curran Associates, Inc. (2020)
31. Machin, M., Guiochet, J., Waeselynck, H., Blanquart, J.P., Roy, M., Masson, L.: Smof: A safety monitoring framework for autonomous systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **48**(5), 702–715 (2018)
32. Mucsányi, B., Kirchhof, M., Oh, S.J.: Benchmarking uncertainty disentanglement: Specialized uncertainties for specialized tasks. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) *Advances in Neural Information Processing Systems*. vol. 37, pp. 50972–51038. Curran Associates, Inc. (2024)
33. Paterson, C., Hawkins, R., Picardi, C., Jia, Y., Calinescu, R., Habli, I.: Safety assurance of machine learning for autonomous systems. *Reliability Engineering & System Safety* **264**, 111311 (2025)
34. Pugnana, A., Ruggieri, S.: Auc-based selective classification. In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. vol. 206, pp. 2494–2514 (2023)
35. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do ImageNet classifiers generalize to ImageNet? In: *International Conference on Machine Learning*. pp. 5389–5400 (2019)

36. Salehi, M., Mirzaei, H., Hendrycks, D., Li, Y., Rohban, M.H., Sabokrou, M.: A unified survey on anomaly, novelty, open-set, and out of-distribution detection: Solutions and future challenges. *Transactions on Machine Learning Research* (2022)
37. Shen, D., Wu, G., Suk, H.I.: Deep Learning in Medical Image Analysis. *Annual Review of Biomedical Engineering* **19**(Volume 19, 2017), 221–248 (2017)
38. Sun, Y., Ming, Y., Zhu, X., Li, Y.: Out-of-distribution detection with deep nearest neighbors. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) *Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 162, pp. 20827–20840. PMLR (2022)
39. timm: resnet50.a1_in1k (2026), https://huggingface.co/timm/resnet50.a1_in1k
40. Ullrich, L., Buchholz, M., Dietmayer, K., Graichen, K.: Toward fully autonomous driving: Ai, challenges, opportunities, and needs. *IEEE Access* **14**, 17971–17997 (2026)
41. Wang, H., Li, Z., Feng, L., Zhang, W.: ViM: Out-Of-Distribution with Virtual-logit Matching. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4911–4920. IEEE (2022)
42. Xia, G., Bouganis, C.S.: Augmenting softmax information for selective classification with out-of-distribution data. In: *Proceedings of the Asian Conference on Computer Vision (ACCV)*. pp. 1995–2012 (2022)
43. Yang, J., Zhou, K., Li, Y., Liu, Z.: Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision* **132**(12), 5635–5662 (2024)
44. Yang, J., Zhou, K., Liu, Z.: Full-spectrum out-of-distribution detection. *International Journal of Computer Vision* **131**(10), 2607–2622 (2023)
45. Zhang, J., Yang, J., Wang, P., Wang, H., Lin, Y., Zhang, H., Sun, Y., Du, X., Zhou, K., Zhang, W., Li, Y., Liu, Z., Chen, Y., Li, H.: OpenOOD v1.5: Enhanced benchmark for out-of-distribution detection. In: *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models* (2024)