Post-hoc Selective Classification for Reliable Synthetic Image Detection

Kaixiang Zheng*
University of Waterloo
k56zheng@uwaterloo.ca

Jacob H. Seidman
Reality Defender
jacob@realitydefender.ai

Abstract

As synthetic images become increasingly realistic, reliable synthetic image detection techniques are of pressing need to prevent their misuse. Despite satisfactory in-distribution performance, deep neural network-based synthetic image detectors (SIDs) lack reliability in deployment and often fail in the presence of common covariate shifts, resulting in poor detection accuracy. To avoid the risk caused by potential errors, we adopt a selective classification (SC) strategy by allowing SIDs to abstain from making low confidence predictions. For practicality, we focus on post-hoc methods which perform confidence estimation on a given SID without retraining. However, we show that conventional logit-based confidence score functions (CSFs) exhibit pathological behavior under covariate shifts, leading to SC performance close to or even worse than random guessing. To address this, we propose a simple yet effective SC framework for **Re**liable **S**ynthetic **I**mage **De**tection (ReSIDe). First, we generalize the notion of logits to an SID's intermediate layers from a centroid matching perspective, extending the use of logit-based CSFs to any layer of an SID. Then, we introduce a preference optimization algorithm that aggregates confidence scores extracted from different layers to a final confidence estimate by minimizing an upper bound of the area under the risk-coverage curve (AURC). Extensive experimental results show that ReSIDe significantly boosts the SC performance of various logit-based CSFs under common covariate shifts, achieving up to 69.55% AURC reduction.

1 Introduction

The rapid development of image generation models [Brock et al., 2019, Nichol et al., 2022, Gu et al., 2022, Rombach et al., 2022, Dhariwal and Nichol, 2021, Midjourney, 2022, Wukong, 2022, Sora, 2025] has made synthetic images almost visually indistinguishable from real ones. To mitigate the risk caused by the misuse of realistic synthetic images, reliable synthetic image detectors (SIDs) are of urgent need. Although modern deep neural network-based (DNN-based) SIDs can be trained to attain near-perfect in-distribution (ID) accuracy, they are often unreliable in deployment upon real-world covariate shifts such as blurring, JPEG compression, low resolution, adversarial attacks, unseen semantic classes, images from unseen generators, etc.

To tackle this challenge and enhance the deployment reliability of SIDs, selective classification (SC) [Chow, 1970, Geifman and El-Yaniv, 2017] provides a principled solution. In an SC framework, a classifier is allowed to abstain from making predictions on samples with low confidence, aiming to increase the accuracy on selected samples. This creates a tradeoff between coverage (the proportion of selected samples) and selective risk (the risk on the selected samples), and an effective SC system should maximize the coverage while minimizing the selective risk. Existing SC methods can be categorized into two streams: methods that require retraining the classifier, and post-hoc SC methods

*This work was done during an internship at Reality Defender.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

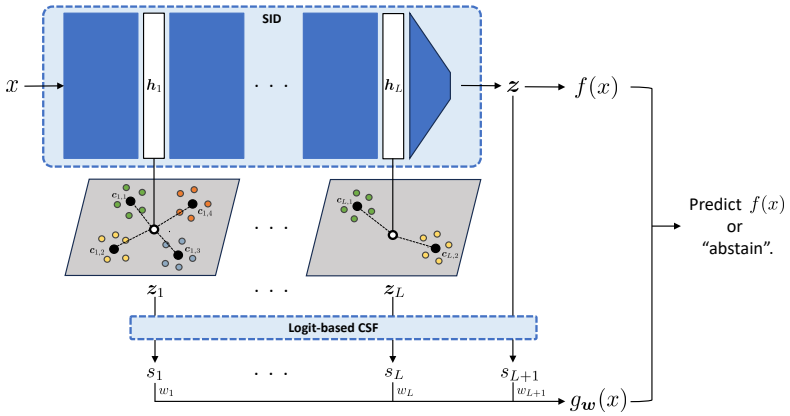


Figure 1: Illustration of the overall ReSIDe framework.

that build upon a fixed classifier without retraining. In this work, we focus on the latter, as it's more cost-effective in practical scenarios. For post-hoc SC, since the classifier is pretrained and fixed, effective SC reduces to designing a confidence score function (CSF) that assigns higher confidence scores to correctly classified samples compared to wrongly classified ones. However, in Sec. 4.2, we observe that for synthetic image detection, conventional logit-based CSFs such as the maximum softmax probability (MSP) [Geifman and El-Yaniv, 2017] often become unreliable under common covariate shifts and can give performance at or worse than using a completely uninformed CSF.

To fix this pathological behavior of logit-based CSFs, we propose ReSIDe, a compact SC framework for **Re**liable **S**ynthetic **I**mage **De**tection with two key components. First, we generalize the notion of logits to an SID's intermediate layers as similarity scores to feature centroids via unsupervised linear probing (ULP). Concretely, for each intermediate layer, we use spherical k-means to compute the centroids of feature vectors from training samples, and use them to form a linear prober to get layerwise logits. Similar to SID's final logits, conventional logit-based CSFs can also be applied to these layerwise logits for confidence estimation, yielding complementary signals to the final layer's logits. Second, we linearly aggregate the confidence scores extracted from all layers, including that from the final logits, with weights optimized by a preference optimization algorithm on a hold-out validation set. We prove that minimizing the preference optimization objective, dubbed the ReSIDe loss, lowers an upper bound to the AURC, thus improving SC performance.

We evaluate ReSIDe on GenImage [Zhu et al., 2023], a large-scale SID benchmark, considering various covariate shifts. On both convolution-based and transformer-based SIDs, ReSIDe consistently and substantially improves SC performance of all kinds of logit-based CSFs, reducing AURC by up to 69.55%. Importantly, ReSIDe is completely plug-and-play, successfully fixing the pathological behavior of final logit-based CSFs with simple linear probing and merely tens of trainable parameters.

The contributions of this work are summarized as follows:

- We adopt selective classification to tackle the deployment reliability issue of SIDs.
- We identify a pathological behavior of final logit-based confidence estimation in synthetic image detection under common covariate shifts.
- We extend the use of logit-based CSFs to SID's intermediate layers based on unsupervised linear probing (ULP).
- We propose a preference optimization algorithm to aggregate layerwise confidence scores, provably minimizing an upper bound of the AURC.
- We evaluate our unified ReSIDe framework with extensive experiments, demonstrating significant and consistent AURC reductions across DNN architectures, covariate shifts, and CSFs, confirming ReSIDe as a practical post-hoc solution for reliable SID deployment.

2 Background and related work

A comprehensive review of related work on SIDs, SC, and CSFs is provided in Appendix A, where we carefully position our work among the relevant literature. In the following, we briefly discuss some background information to facilitate understanding of our proposed method.

Logits as centroid matching scores. Although logits are typically seen as a linear projection of DNN's penultimate layer features, they have also been interpreted as similarity scores between a feature vector and feature centroids (also known as prototypes or templates). For example, Müller et al. [2019] view each (column) vector in DNN's last linear layer as a class template, and therefore each logit measures the dot product similarity between the penultimate layer feature vector and the corresponding class template. Neural collapse [Papyan et al., 2020] further strengthens this view by showing these weight vectors indeed converge to the class means of DNN's penultimate layer features, up to rescaling. In fact, this centroid matching principle is baked in the design of many DNN architectures. For example, in Prototypical Networks [Snell et al., 2017], the centroids are given by the class means of learnable embeddings, and logits are computed by the negative squared Euclidean distance between a query's embedding and each class mean. In CLIP [Radford et al., 2021], the centroids are given by the normalized text embeddings, and logits are computed by the cosine similarity between the normalized image embedding and each text embedding. In this work, inspired by the centroid matching interpretation of logits, we generalize the notion of logits to intermediate layers of SIDs using centroid-based linear probes, thus extending the use of logit-based CSFs from SID's final output to any layer of interest.

Pairwise contrast for preference optimization. In SC, a desirable CSF should give higher confidence scores to correctly classified samples than misclassified ones. To learn scalar functions with such explicit preference, the following probabilistic approach is commonly adopted. Consider a model $r_\phi : \mathbb{R}^d \to \mathbb{R}$ parameterized by ϕ that receives a d-dimensional feature vector as input and outputs a real number. An input a is preferred over another input b, denoted by $a \succ b$, if $r_\phi(a) > r_\phi(b)$. Modeling the probability of $a \succ b$ by the Bradley–Terry model [Bradley and Terry, 1952], i.e., $\Pr\{a \succ b\} = e^{r_\phi(a)}/(e^{r_\phi(a)} + e^{r_\phi(b)})$, one can train r_ϕ by minimizing the expected negative log-likelihood (NLL) loss

$$\min_\phi \mathbb{E}_{a\sim A, b\sim B}\left[-\ln \Pr\{a \succ b\}\right] = \min_\phi \mathbb{E}_{a\sim A, b\sim B}\left[\ln(1+\exp(r_\phi(b) - r_\phi(a)))\right], \tag{1}$$

where A and B are the distributions of the preferred and non-preferred patterns. Such trained model tends to return a higher score for a future input if it's more preferable.

This general framework has wide applications in various fields. In information retrieval, RankNet [Burges et al., 2005] follows the framework to rank query-document pairs for building search engines; in large language model (LLM) post-training, reinforcement learning from human feedback (RLHF) [Ouyang et al., 2022] and direct preference optimization (DPO) [Rafailov et al., 2023] follow the framework for reward modeling based on human preference; more recently, Mole-PAIR [He et al., 2025] also follows the framework to perform OOD detection for molecular foundation models. In this work, we adopt this framework to aggregate confidence scores extracted from different DNN layers for better SC performance.

3 Notations and preliminaries

Selective classification. Consider a classification problem with input space $\mathcal{X}$ and label space $\mathcal{Y} = [C] := \{1, \ldots, C\}$. Let P be an unknown data distribution over $\mathcal{X} \times \mathcal{Y}$. The risk of a classifier $f : \mathcal{X} \to \mathcal{Y}$ is defined as $R(f) = \mathbb{E}_{(x,y)\sim P}[l(f(x), y)]$, where l is a loss function. By convention, we set l to the 0/1 loss, $l(f(x), y) = \mathbb{1}[f(x) \neq y]$, where $\mathbb{1}[\cdot]$ denotes the indicator function. In this case, the risk is equal to the error rate of the classifier. A selective classifier is defined as a pair (f, g), where f is a classifier and $g : \mathcal{X} \to \mathbb{R}$ is a confidence score function (CSF) that measures the classifier's confidence on its prediction for a given input. Given a threshold t, the selective classifier accepts an input x and outputs the corresponding prediction $f(x)$ if $g(x) \geq t$; otherwise, it abstains from making a prediction, i.e.,

$$(f, g)(x) := \begin{cases} f(x) & \text{if } g(x) \geq t, \\ \text{“abstain”} & \text{otherwise.} \end{cases} \tag{2}$$

The performance of a selective classifier is characterized by its coverage and selective risk. Given t, the coverage of a selective classifier is defined as the probability of accepting an input, i.e.,

$\phi_t(f,g) = \Pr\{g(x) \geq t\}$, and the selective risk is defined as the conditional risk given acceptance, i.e., $R_t(f,g) = \mathbb{E}_P[l(f(x),y) \mid g(x) \geq t]$. For 0/1 loss, the selective risk is equal to the conditional error rate given acceptance, i.e., $R_t(f,g) = \Pr\{f(x) \neq y \mid g(x) \geq t\}$. Note that a classifier's risk is equal to its selective risk at full coverage, i.e., $R(f) = R_t(f,g)$ if t is chosen such that $\phi_t(f,g) = 1$. Empirically, given a test set $T = \{(x_i, y_i)\}_{i=1}^N$ drawn i.i.d from P, we evaluate the empirical coverage and the empirical selective risk as follows:

$$\hat{\phi}_t(f,g,T) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[g(x_i) \geq t],\ \hat{R}_t(f,g,T) = \frac{\sum_{i=1}^{N} l(f(x_i),y_i)\mathbb{1}[g(x_i) \geq t]}{\sum_{i=1}^{N} \mathbb{1}[g(x_i) \geq t]}. \tag{3}$$

Given a decent CSF, one can typically trade coverage for selective risk by increasing the threshold t and abstaining from more uncertain samples. This tradeoff can be visualized by the risk-coverage (RC) curve, which plots $\hat{R}_t(f,g,T)$ as a function of $\hat{\phi}_t(f,g,T)$. A lower RC curve indicates better SC performance, since for a fixed coverage a smaller selective risk is preferred and for a fixed selective risk a larger coverage is preferred. To summarize the RC curve using a single scalar, an established metric is the *area under the risk-coverage curve* (AURC), denoted by $\text{AURC}(f,g,T)$. Since this paper focuses exclusively on post-hoc SC, we suppress the notational dependence on the fixed classifier f. In this case, the quality of the CSF is pivotal to the SC performance, since the RC curve and the resulting $\text{AURC}(g,T)$ are directly determined by g, given the test set T.

Two special cases of CSF and the corresponding SC performance. First, if a CSF g^* enables perfect ranking of samples in T such that correctly classified samples always get higher confidence scores than those of wrongly classified ones, i.e., $g^*(x^+) > g^*(x^-)$ for any $x^+ \in \{x \in T \mid f(x) = y\}$ and $x^- \in \{x \in T \mid f(x) \neq y\}$, then we achieve the optimal SC performance on T with the lowest possible AURC. Second, if a CSF is independent to the classifier's error, e.g., a random CSF, its AURC is equal to the classifier's risk, which can be approximated on T by the empirical error rate $\hat{r} = \frac{1}{N}\sum_{i=1}^{N} l(f(x_i),y_i) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[f(x_i) \neq y_i]$. CSFs used in practice typically yield AURCs between $\text{AURC}(g^*,T)$ and $\hat{r}$. However, if a CSF generally gives higher confidence scores to wrongly classified samples than correctly classified ones, the AURC can be larger than $\hat{r}$.

Logit-based CSFs. DNN-based classifiers generally make predictions by $f(x) = \hat{y} := \arg\max_{c\in\mathcal{Y}} z_c$, where $\boldsymbol{z}(x) = [z_1, \ldots, z_C] \in \mathbb{R}^C$ is referred to as logits. In this case, the most popular CSFs are logit-based CSFs, which compute confidence scores directly from the logits $\boldsymbol{z}$. Since softmax is routinely applied to logits to obtain an estimate of the posterior distribution $[P(y|x)]_{y\in\mathcal{Y}}$, the maximum probability mass output by softmax can be naturally regarded as the confidence on the prediction $\hat{y}$. This yields a logit-based CSF called the maximum softmax probability (MSP) [Hendrycks and Gimpel, 2017, Geifman and El-Yaniv, 2017]. Beyond MSP, other typical logit-based CSFs include softmax margin (SM) [Belghazi and Lopez-Paz, 2021], the max logit (ML) [Hendrycks et al., 2022], the logits margin (LM) [Le Cun et al., 1990, Liang et al., 2024], the negative entropy (NE) [Belghazi and Lopez-Paz, 2021], and the negative Gini index (NGI) [Granese et al., 2021]. Formally, they are defined as:

$$\begin{array}{lll} \text{MSP}(\boldsymbol{z}) = \sigma_{\hat{y}}(\boldsymbol{z}), & \text{SM}(\boldsymbol{z}) = \sigma_{\hat{y}}(\boldsymbol{z}) - \max_{c\in\mathcal{Y}:\, c\neq\hat{y}} \sigma_c(\boldsymbol{z}), & \text{ML}(\boldsymbol{z}) = z_{\hat{y}}, \\ \text{LM}(\boldsymbol{z}) = z_{\hat{y}} - \max_{c\in\mathcal{Y}:\, c\neq\hat{y}} z_c, & \text{NE}(\boldsymbol{z}) = \sum_{c\in\mathcal{Y}} \sigma_c(\boldsymbol{z}) \ln \sigma_c(\boldsymbol{z}), & \text{NGI}(\boldsymbol{z}) = -1 + \sum_{c\in\mathcal{Y}} \sigma_c(\boldsymbol{z})^2, \end{array}$$

where $\sigma(\cdot)$ is the softmax operation, and $\sigma_c(\cdot)$ is the c-th element of the softmax-processed vector.

In this paper, the classifier f is a DNN-based SID, $C = 2$ for the binary SID task, and the data distribution P denotes the training distribution of f with potential covariate shifts.

4 Empirical evaluation of logit-based CSFs

In this section, we evaluate the SC performance of standard logit-based CSFs for synthetic image detection with six types of common covariate shifts. We first detail the data and model setup and then demonstrate pathological SC performance with these logit-based CSFs, thus motivating our method.

4.1 Data and model preparation

Dataset. GenImage [Zhu et al., 2023] is a large-scale dataset for synthetic image detection, consisting of over 2.6 million images. Roughly half of the images are real and the other half are synthetic from eight generators, namely BigGAN [Brock et al., 2019], GLIDE [Nichol et al., 2022], VQDM [Gu

et al., 2022], Stable Diffusion V1.4 [Rombach et al., 2022], Stable Diffusion V1.5 [Rombach et al., 2022], ADM [Dhariwal and Nichol, 2021], Midjourney [Midjourney, 2022], and Wukong [Wukong, 2022]. Note that GenImage is built upon ImageNet-1K [Deng et al., 2009] and covers images with the same 1000 semantic classes. We curate the datasets used in our experiments from GenImage to create representative covariate shift scenarios. The training set consists of real images and images generated by ADM and BigGAN, covering 50 semantic classes. To test SID's performance under different distributions, we curate seven test sets. The in-distribution (ID) test set follows the same distribution as the training set, i.e., containing images from the same sources and semantic classes. The remaining six test sets contain various covariate shifts: (1) **Blur:** images are processed by Gaussian blurring with $\sigma \sim \text{Uniform}[3, 5]$, where σ is the standard deviation (std) of the Gaussian kernel; (2) **Unseen semantic classes (Unseen Cls.):** images from 950 semantic classes not presented in training; (3) **Low resolution (Low Res.):** images are resized while preserving the aspect ratio, such that the length of the shorter edge is uniformly random in $[64, 96]$; (4) **FGSM:** images are perturbed using the fast gradient sign method (FGSM) [Goodfellow et al., 2015], with the distortion budget ϵ uniformly random in $\{1/255, 2/255, ..., 8/255\}$; (5) **JPEG:** images are compressed with JPEG with $q \sim \text{Uniform}[30, 70]$, where q is the quality factor of JPEG compression; (6) **Unseen generators (Unseen Gen.):** synthetic images are generated by Wukong, Midjourney, Stable Diffusion V1.4, and Stable Diffusion V1.5, which are not presented in training. Note that to isolate the effect of different covariate shifts and study them independently, each covariate-shifted testing distribution differs from the training distribution in only one aspect, while other aspects stay the same as the training distribution. For example, for the dataset with unseen semantic classes, the synthetic images are generated by ADM and BigGAN; for the dataset with unseen generators, the semantic classes are the same 50 classes as in the training set.

Models. We train two SIDs using representative DNN architectures, a CNN-based architecture ResNet50 [He et al., 2016] and a transformer-based architecture Swin-T [Liu et al., 2021]. Both models are trained on the training set using standard supervised learning and kept fixed during all SC evaluations. Note that the test set with FGSM perturbations is generated by attacking the ResNet50-based SID.

Tab. 1 reports the testing error rates of the two SIDs under different data distributions. As shown, both models achieve error rates close to zero on ID data but suffer from substantial performance degradation under most covariate shifts, often approaching random guessing level with error rates close to 0.5. The covariate shift based on unseen semantic classes is the only exception, where the accuracy degradation is below 5% for both models.

Table 1: Testing error rates of SIDs under different data distributions.

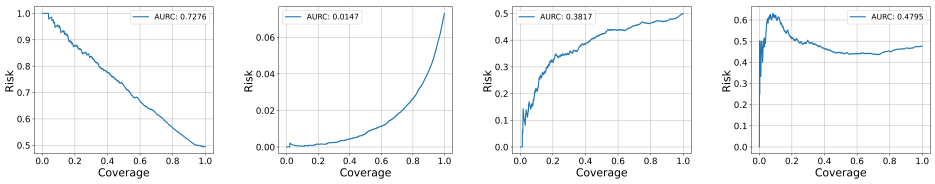
	ID	Blur	Unseen Cls.	Low Res.	FGSM	JPEG	Unseen Gen.
ResNet50	0.0242	0.4950	0.0732	0.5000	0.4767	0.4783	0.4850
Swin-T	0.0150	0.4950	0.0341	0.4875	0.3975	0.4792	0.4669

4.2 Pathological SC performance

We now examine the SC performance of logit-based CSFs under different data distributions. Figs. 2(a)-(f) show the RC curves for MSP evaluated using ResNet50 on six covariate-shifted test sets, together with the resulting AURC. While the SC performance on Unseen Cls. and JPEG are decent, those on the remaining cases are much poorer. Specifically, the AURCs on Blur and FGSM are greater than the empirical error rates on respective test sets, meaning that MSP is behaving worse than a random CSF. A pathological SC performance is especially clear in the case of Blur, where the selective risk is going up when the coverage goes down, indicating that correctly classified samples generally have lower MSP scores than wrongly classified ones and are being rejected first when the threshold t increases. Similar pathological performance is demonstrated for all six logit-based CSFs and both models, whose AURCs are presented in Tab. 2.

In practical deployment scenarios, an input could be in-distribution (ID) or from various covariate-shifted distributions. It's therefore of interest to analyze the SC performance on a mixture of multiple data distributions. We consider an even mixture of seven distributions[2]: ID, Blur, Unseen Cls., Low Res., FGSM, JPEG, Unseen Gen. are assumed to happen equal likely. In this case, the error rate of ResNet50 is 0.3618, and that of Swin-T is 0.3393 (the average of error rates over all seven cases).

[2]The number of samples varies among test sets. To handle the imbalance and create an even mixture, we reweight samples when computing SC metrics. Please refer to Appendix B for details.



(a) Blur (b) Unseen Cls. (c) Low Res. (d) FGSM

(e) JPEG (f) Unseen Gen. (g) ResNet50 (h) Swin-T

Figure 2: (a)-(f): RC curves for MSP evaluated using ResNet50 on datasets with different covariate shifts. (g)-(h): RC curves for MSP under the mixture distribution, evaluated using ResNet50 and Swin-T, respectively.

Figs. 2(g)-(h) plot the RC curves for MSP evaluated using both models under the mixture distribution, together with the resulting AURC. As shown, the selective risk hardly goes down as the coverage decreases, resulting in AURCs close to those of random CSFs on both models. Other logit-based CSFs yield similar SC performance, as shown in Tab. 3. These findings reveal that conventional final logit-based CSFs are not reliable for SID selective classification under covariate shifts. Therefore, we are motivated to exploit additional information from SID's intermediate layers.

5 Methodology

5.1 Generalizing logit-based CSFs

As mentioned in Sec. 2, logits can be regarded as similarity scores between a feature vector and feature centroids. Based on this principle, we generalize the notion of logits to SID's intermediate layers as follows. Given a DNN-based SID f with L hidden layers, for each layer $l \in [L]$, define a set of K_l feature centroids as $\{\boldsymbol{c}_{l,1}, ..., \boldsymbol{c}_{l,K_l}\}$. Then, the generalized logits at layer l can be computed by

$$\boldsymbol{z}_l = [z_{l,k}]_{k=1}^{K_l} = [s(\boldsymbol{h}_l, \boldsymbol{c}_{l,k})]_{k=1}^{K_l}, \tag{4}$$

where $\boldsymbol{h}_l$ is a feature vector at layer l corresponding to some input x, and $s(\cdot, \cdot)$ is a given similarity metric such as cosine similarity, negative Euclidean distance, negative Mahalanobis distance, etc. Given $\boldsymbol{z}_l$ and any logit-based CSF, we can compute a confidence score s_l at layer l. Defining $\hat{y}_l := \arg\max_{k \in [K_l]} z_{l,k}$, we have, for example, $s_l = \text{MSP}(\boldsymbol{z}_l) = \sigma_{\hat{y}_l}(\boldsymbol{z}_l)$, with MSP being the underlying logit-based CSF. Thus, given a logit-based CSF, we can extract in total $L+1$ confidence scores $\{s_1, ..., s_L, s_{L+1}\}$ from an SID, where the first L scores are extracted from L hidden layers, respectively, and the last score s_{L+1} is the standard score computed from SID's final logits $\boldsymbol{z}$.

Centroids and similarity metric. In this work, we construct feature centroids by clustering. For each layer l, we obtain feature vectors $\boldsymbol{h}_l$ of all samples in SID's training set. Then, we perform spherical k-means [Dhillon and Modha, 2001] to get K_l feature centroids, where K_l is determined by silhouette analysis [Shahapure and Nicholas, 2020]. As for the similarity metric $s(\cdot, \cdot)$, we adopt cosine similarity to be consistent with the clustering objective of spherical k-means. Given these specific design choices, we can rewrite Eq. (4) as

$$\boldsymbol{z}_l = C_l^\top \hat{\boldsymbol{h}}_l, \tag{5}$$

where $C_l = [\boldsymbol{c}_{l,1}, ..., \boldsymbol{c}_{l,K_l}]$ is a matrix composed of feature centroids obtained from spherical k-means, and $\hat{\boldsymbol{h}}_l = \boldsymbol{h}_l / ||\boldsymbol{h}_l||_2$ is a l_2-normalized feature vector. Note that the centroids given by spherical k-means are on the unit sphere, so no normalization is needed for them. Since the feature centroids are obtained by unsupervised clustering, and Eq. (5) is a simple linear projection of SID's intermediate features, we dub the logits computation in Eq. (5) unsupervised linear probing (ULP).

5.2 Preference optimization for layer aggregation

Based on ULP, we augment the single confidence score s_{L+1} computed from SID's final logits to a set of $L+1$ layerwise confidence scores $\{s_1, ..., s_L, s_{L+1}\}$. However, to perform SC, we need a single scalar as the output of the CSF g. Therefore, in this section, we discuss how to aggregate the $L+1$ confidence scores into a final confidence estimate.

For simplicity, we construct the final confidence estimate as a linear combination of all $L+1$ scores, i.e, $g_{\boldsymbol{w}}(x) = \sum_{l=1}^{L+1} w_l s_l$, where $\boldsymbol{w} = [w_1, ..., w_{L+1}] \in \mathbb{R}^{L+1}$. Assuming access to a hold-out validation set $V = \{(u_j, v_j)\}_{j=1}^{M}$, with inputs $u_j \in \mathcal{X}$ and labels $v_j \in \mathcal{Y}$, we optimize $\boldsymbol{w}$ on it by minimizing a preference optimization objective inspired by Eq. (1)

$$\min_{\boldsymbol{w}} \Big\{ L_{ReSIDe}(g_{\boldsymbol{w}}, V) := \frac{1}{(M-n)n} \sum_{u^+ \in U^+} \sum_{u^- \in U^-} \ln(1 + \exp(g_{\boldsymbol{w}}(u^-) - g_{\boldsymbol{w}}(u^+))) \Big\}, \tag{6}$$

where $U^+ = \{u \in V \mid f(u) = v\}$ and $U^- = \{u \in V \mid f(u) \neq v\}$ are the sets of correctly and wrongly classified samples, respectively, and $n = |U^-|$ is the number of wrongly classified samples.

Denote a solution to Eq. (6) as $\boldsymbol{w}^*$. As mentioned in Sec. 2, $g_{\boldsymbol{w}^*}$ tends to assign higher scores to correctly classified samples and lower scores to wrongly classified ones. In an extreme case, if $g_{\boldsymbol{w}^*}(u^+) > g_{\boldsymbol{w}^*}(u^-)$ for any $u^+ \in U^+$ and $u^- \in U^-$, then we essentially achieve the optimal SC performance on V. To further justify the use of L_{ReSIDe} in the context of SC, we show that L_{ReSIDe} is in fact an upper bound of AURC in Thm. 1, which is proved in Appendix C.

Theorem 1 (Upper Bound of AURC via the ReSIDe Loss)**.** *Let L_{ReSIDe} be the ReSIDe loss defined in Eq. (6). Then, we have*

$$\mathit{AURC}(g_{\boldsymbol{w}}, V) < \frac{2(M-n)n}{M^2 \ln 2} L_{ReSIDe}(g_{\boldsymbol{w}}, V) + \frac{2n^2}{M^2}. \tag{7}$$

Further assuming that each sample in U^- has a unique confidence score, a tighter bound follows

$$\mathit{AURC}(g_{\boldsymbol{w}}, V) < \frac{2(M-n)n}{M^2 \ln 2} L_{ReSIDe}(g_{\boldsymbol{w}}, V) + \frac{n(n+1)}{M^2}. \tag{8}$$

Therefore, minimizing the ReSIDe loss effectively minimizes an upper bound of the AURC, thus resulting in $g_{\boldsymbol{w}}$ with superior SC performance. To reduce complexity in the actual optimization process, we sample mini-batches of correctly and wrongly classified samples from V, denoted as $\mathcal{U}^+$ and $\mathcal{U}^-$, respectively, and minimize $L_{ReSIDe}(g_{\boldsymbol{w}}, \mathcal{U}^+ \cup \mathcal{U}^-)$ via gradient descent for each mini-batch until convergence or up to a fixed number of epochs. To conclude this section, we refer readers to Fig. 1 for an illustration of the overall ReSIDe framework.

6 Experiments

6.1 Experimental settings

To perform preference optimization, we curate seven hold-out validation sets based on GenImage, each corresponding to one of the testing distributions within ID, Blur, Unseen Cls., Low Res., FGSM, JPEG, and Unseen Gen. Note that in real-world scenarios, it's feasible for practitioners to construct hold-out validation sets corresponding to common types of covariate shift, but it's unrealistic to anticipate the exact covariate shifts in the wild. For example, at the time of preparing the validation sets, new generators, from which potential testing images are generated, may not have been invented yet. Therefore, to reflect this inevitable bias, there's a distributional difference between the covariate shifts in our hold-out validation sets and those in the test sets. For instance, the validation set with Unseen Gen. contains synthetic images generated by GLIDE and VQDM, instead of those generated by the four generators used in the Unseen Gen. test set. More information on validation sets, training recipes, and full implementation details are provided in Appendix D.

As an approach aiming to fix the pathological behavior of logit-based CSFs, we benchmark ReSIDe against each underlying logit-based CSF applied only at the final layer. Given a logit-based CSF, we optimize $\boldsymbol{w}$ on a hold-out validation set with some covariate shift, and compare the AURC of $g_{\boldsymbol{w}^*}$ to that of the underlying final logit-based CSF on the test set with the same covariate shift. For comprehensiveness, this comparison is done for both SID models, all six types of logit-based CSFs and all six types of covariate shifts. Moreover, we also optimize $\boldsymbol{w}$ on the combination of all validation sets, and evaluate the resulting $g_{\boldsymbol{w}^*}$ under the mixture distribution mentioned in Sec. 4.2.

Table 2: Comparison to final logit-based CSFs under different covariate shifts (AURC). Results for ReSIDe are averaged over 3 trials with stds reported. Δ shows the AURC reduction achieved by ReSIDe in percentage.

Classifier	CSF	Blur			Unseen Cls.			Low Res.		
		Baseline	ReSIDe	Δ	Baseline	ReSIDe	Δ	Baseline	ReSIDe	Δ
ResNet50	MSP	0.7276	$0.2406_{\pm 0.0006}$	66.93%	0.0147	$0.0095_{\pm 0.0000}$	35.37%	0.3817	$0.3610_{\pm 0.0016}$	5.42%
	SM	0.7276	$0.2467_{\pm 0.0011}$	66.09%	0.0147	$0.0095_{\pm 0.0001}$	35.37%	0.3817	$0.3811_{\pm 0.0001}$	0.16%
	ML	0.7271	$0.2214_{\pm 0.0007}$	69.55%	0.0146	$0.0128_{\pm 0.0001}$	12.33%	0.3782	$0.3368_{\pm 0.0005}$	10.95%
	LM	0.7276	$0.2414_{\pm 0.0003}$	66.82%	0.0147	$0.0092_{\pm 0.0001}$	37.41%	0.3817	$0.3691_{\pm 0.0064}$	3.30%
	NE	0.7276	$0.2472_{\pm 0.0001}$	66.03%	0.0147	$0.0096_{\pm 0.0003}$	34.69%	0.3817	$0.3552_{\pm 0.0009}$	6.94%
	NGI	0.7276	$0.2524_{\pm 0.0006}$	65.31%	0.0147	$0.0096_{\pm 0.0002}$	34.69%	0.3817	$0.3582_{\pm 0.0011}$	6.16%
Swin-T	MSP	0.6535	$0.2889_{\pm 0.0003}$	55.79%	0.0047	$0.0025_{\pm 0.0000}$	46.81%	0.4167	$0.3255_{\pm 0.0004}$	21.89%
	SM	0.6535	$0.2379_{\pm 0.0007}$	63.60%	0.0047	$0.0025_{\pm 0.0000}$	46.81%	0.4167	$0.3178_{\pm 0.0001}$	23.73%
	ML	0.6007	$0.2311_{\pm 0.0006}$	61.53%	0.0059	$0.0040_{\pm 0.0000}$	32.20%	0.3967	$0.3236_{\pm 0.0038}$	18.43%
	LM	0.6535	$0.2187_{\pm 0.0003}$	66.53%	0.0047	$0.0025_{\pm 0.0000}$	46.81%	0.4167	$0.3073_{\pm 0.0035}$	26.25%
	NE	0.6535	$0.3079_{\pm 0.0006}$	52.88%	0.0047	$0.0025_{\pm 0.0000}$	46.81%	0.4167	$0.3293_{\pm 0.0000}$	20.97%
	NGI	0.6535	$0.3154_{\pm 0.0001}$	51.74%	0.0047	$0.0025_{\pm 0.0000}$	46.81%	0.4167	$0.3349_{\pm 0.0004}$	19.63%

Classifier	CSF	FGSM			JPEG			Unseen Gen.		
		Baseline	ReSIDe	Δ	Baseline	ReSIDe	Δ	Baseline	ReSIDe	Δ
ResNet50	MSP	0.4795	$0.3449_{\pm 0.0009}$	28.07%	0.1922	$0.1922_{\pm 0.0000}$	0.00%	0.3684	$0.3630_{\pm 0.0000}$	1.47%
	SM	0.4795	$0.4030_{\pm 0.0003}$	15.95%	0.1922	$0.1922_{\pm 0.0000}$	0.00%	0.3684	$0.3630_{\pm 0.0000}$	1.47%
	ML	0.4847	$0.4501_{\pm 0.0009}$	7.14%	0.1925	$0.1864_{\pm 0.0001}$	3.17%	0.3650	$0.3650_{\pm 0.0000}$	0.00%
	LM	0.4795	$0.4035_{\pm 0.0011}$	15.85%	0.1922	$0.1855_{\pm 0.0001}$	3.49%	0.3684	$0.3630_{\pm 0.0000}$	1.47%
	NE	0.4795	$0.3461_{\pm 0.0001}$	27.82%	0.1922	$0.1922_{\pm 0.0000}$	0.00%	0.3684	$0.3630_{\pm 0.0000}$	1.47%
	NGI	0.4795	$0.3467_{\pm 0.0002}$	27.70%	0.1922	$0.1922_{\pm 0.0000}$	0.00%	0.3684	$0.3630_{\pm 0.0000}$	1.47%
Swin-T	MSP	0.3023	$0.2889_{\pm 0.0000}$	4.43%	0.1857	$0.1764_{\pm 0.0001}$	5.01%	0.3480	$0.3193_{\pm 0.0000}$	8.25%
	SM	0.3023	$0.3014_{\pm 0.0002}$	0.30%	0.1857	$0.1727_{\pm 0.0002}$	7.00%	0.3480	$0.3193_{\pm 0.0000}$	8.25%
	ML	0.3022	$0.2460_{\pm 0.0021}$	18.60%	0.1899	$0.1642_{\pm 0.0009}$	13.53%	0.3737	$0.3737_{\pm 0.0000}$	0.00%
	LM	0.3023	$0.2956_{\pm 0.0024}$	2.22%	0.1857	$0.1698_{\pm 0.0000}$	8.56%	0.3480	$0.3193_{\pm 0.0000}$	8.25%
	NE	0.3023	$0.2889_{\pm 0.0000}$	4.43%	0.1857	$0.1720_{\pm 0.0001}$	7.38%	0.3480	$0.3193_{\pm 0.0000}$	8.25%
	NGI	0.3023	$0.2889_{\pm 0.0000}$	4.43%	0.1857	$0.1729_{\pm 0.0002}$	6.89%	0.3480	$0.3193_{\pm 0.0000}$	8.25%

Table 3: Comparison to final logit CSFs under the mixture distribution (AURC).

Classifier	CSF	Baseline	ReSIDe	Δ
ResNet50	MSP	0.3563	$0.2004_{\pm 0.0008}$	43.76%
	SM	0.3563	$0.2072_{\pm 0.0006}$	41.85%
	ML	0.3635	$0.2266_{\pm 0.0034}$	37.66%
	LM	0.3563	$0.2094_{\pm 0.0014}$	41.23%
	NE	0.3563	$0.1972_{\pm 0.0003}$	44.65%
	NGI	0.3563	$0.2028_{\pm 0.0035}$	43.08%
Swin-T	MSP	0.3347	$0.1947_{\pm 0.0006}$	41.83%
	SM	0.3347	$0.1902_{\pm 0.0022}$	43.17%
	ML	0.2868	$0.2048_{\pm 0.0001}$	28.59%
	LM	0.3347	$0.1858_{\pm 0.0002}$	44.49%
	NE	0.3347	$0.1879_{\pm 0.0005}$	43.86%
	NGI	0.3347	$0.1845_{\pm 0.0003}$	44.88%

Table 4: Comparison to best single layer CSFs under the mixture distribution (AURC).

Classifier	CSF	Best Layer	ReSIDe	Δ
ResNet50	MSP	0.2554	$0.2004_{\pm 0.0008}$	21.53%
	SM	0.2554	$0.2072_{\pm 0.0006}$	18.87%
	ML	0.2580	$0.2266_{\pm 0.0034}$	12.17%
	LM	0.2554	$0.2094_{\pm 0.0014}$	18.01%
	NE	0.2554	$0.1972_{\pm 0.0003}$	22.79%
	NGI	0.2554	$0.2028_{\pm 0.0035}$	20.60%
Swin-T	MSP	0.2266	$0.1947_{\pm 0.0006}$	14.08%
	SM	0.2266	$0.1902_{\pm 0.0022}$	16.06%
	ML	0.2152	$0.2048_{\pm 0.0001}$	4.83%
	LM	0.2266	$0.1858_{\pm 0.0002}$	18.01%
	NE	0.2299	$0.1879_{\pm 0.0005}$	18.27%
	NGI	0.2266	$0.1845_{\pm 0.0003}$	18.58%

6.2 Main results

Tab. 2 shows superior performance of ReSIDe for each logit-based CSF, covariate shift, and SID model. Compared to the baseline of the underlying final logit-based CSF, ReSIDe achieves consistent and significant improvement in SC performance, reducing AURC by up to 69.55%. Tab 3 compares ReSIDe to the baseline on the mixture distribution for all logit-based CSFs and both models. ReSIDe again outperforms the baseline in all cases, achieving up to 44.88% AURC reduction.

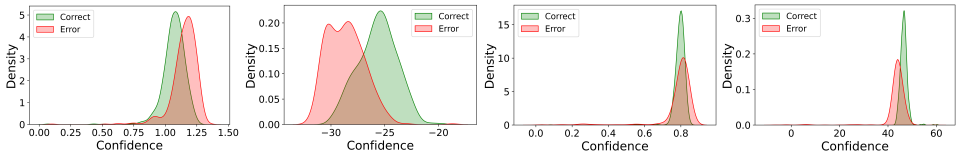
We expand these evaluations in Appendices G ~ J where we demonstrate generalization to unknown covariate shifts, selective classification performance on the recent EvalGen benchmark [Chen et al., 2025], behavior of ReSIDe on a CLIP-based SID, and compare against the work in Yumlembam et al. [2025] and the OOD detection methods presented in Liu et al. [2020] and Lee et al. [2018].

6.3 Extensions

Orthogonality to logit transformation. Tunable logit transformation are a collection of methods to improve post-hoc SC performance of logit-based CSFs. For example, pNorm [Cattelan and Silva, 2024, 2023] applies centralization followed by l_p normalization to logits to boost the SC performance of MaxLogit (ML), where p is optimized by grid search on a hold-out validation set. Here, we show

Table 5: Comparison of AURCs using ML as the underlying logit-based CSF.

		Blur	Unseen Cls.	Low Res.	FGSM	JPEG	Unseen Gen.
ResNet50	pNorm	0.5144	0.0247	0.4834	0.1747	0.1453	0.1540
	ReSIDe-pNorm	0.2834	0.0204	0.4498	0.1747	0.1453	0.1540
	Δ	44.91%	17.41%	6.95%	0.00%	0.00%	0.00%
Swin-T	pNorm	0.5370	0.0167	0.4928	0.2452	0.1503	0.1496
	ReSIDe-pNorm	0.3135	0.0144	0.3509	0.2182	0.1441	0.1471
	Δ	41.62%	13.77%	28.79%	11.01%	4.13%	1.67%



(a) Baseline, ResNet50 (b) ReSIDe, ResNet50 (c) Baseline, Swin-T (d) ReSIDe, Swin-T

Figure 3: Comparison of confidence score distributions between the baseline and ReSIDe.

that ReSIDe can be used on top of pNorm to get further gain. Particularly, we apply pNorm not only to SID's final logits but also to the layerwsie logits extracted by ULP, while keeping other procedures of ReSIDe unchanged; we dub this combined method as ReSIDe-pNorm. Using ML as the underlying logit-based CSF, Tab. 5 compares the performance of ReSIDe-pNorm to the standard pNorm, which only utilizes SID's final logits. It's shown that ReSIDe-pNorm outperforms the standard pNorm in general, achieving up to 44.91% AURC reduction. Another tunable logit transformation used in the SC literature is temperature scaling (TS) [Guo et al., 2017, Cattelan and Silva, 2024]. While TS can improve the SC performance of some logit-based CSFs in multi-class classification, it loses effectiveness for all of the six discussed logit-based CSFs in the case of binary classification, since the ranking of confidence scores is not influenced by TS. Therefore, TS is excluded from our discussion.

Ablation study: the single best layer vs. the optimized linear combination. The unified ReSIDe framework consists of two main components: (1) layerwsie score extraction based on ULP, and (2) score aggregation based on preference optimization. We hereby conduct an ablation study to analyze the contribution of each components. Given a logit-based CSF, we define the best layer as the layer with index $l^* = \arg\min_{l\in[L+1]} \text{AURC}(g_{\boldsymbol{e}_l}, V)$, where $\boldsymbol{e}_l \in \mathbb{R}^{L+1}$ denotes a one-hot vector whose l-th entry is one. Then, we evaluate $\text{AURC}(g_{\boldsymbol{e}_{l^*}}, T)$ on the corresponding test set. Tab. 4 shows the performance of the best layer for each logit-based CSF and SID model under the mixture distribution. Comparing the best layer's results to those of the baseline presented in Tab. 3, we see that the best layer consistently outperforms the baseline. Nonetheless, ReSIDe provides further improvement over the best layer, achieving up to 22.79% AURC reduction. This shows that both components within ReSIDe contribute meaningfully to its superior performance.

Distribution of confidence scores. Since the ranking of confidence scores determines the SC performance, it's informative to analyze the confidence score distributions of correctly classified samples and wrongly classified samples. Figs. 3(a) and (b) compare the score distributions of ReSIDe and the baseline using ML as the underlying logit-based CSF, evaluated for ResNet50 in the case of Blur. As shown, wrongly classified samples generally get higher scores than correctly classified ones in the baseline, causing the pathological SC behavior identified in Fig. 2; while ReSIDe addresses the issue by assigning generally higher scores to correctly classified samples. A similar analysis is done for Swin-T in Figs. 3(c) and (d), with SM as the underlying logit-based CSF, also evaluated in the case of Blur. Again, ReSIDe successfully flips the ordering between correctly classified samples and wrongly classified ones, on average giving the former higher scores than the latter. This analysis explains why ReSIDe substantially improves the SC performance of the baseline.

More analysis on the **empirical convergence behavior of our preference optimization algorithm** and the **computational complexity** are shown in Appendix E and F, respectively.

7 Conclusion

The paper identifies a pathological SC behavior of logit-based CSFs when applied to SIDs under common covariate shifts. To fix this issue, ReSIDe is proposed to generalize the use of logit-based CSFs to SID's intermediate layers. Using a linear combination of layerwise confidence scores as the final confidence estimate, ReSIDe can significantly improve the SC performance of logit-based CSFs.

Comprehensive experimental results show that ReSIDe is effective over all kinds of covariate shifts, logit-based CSFs and SID models.

Limitations: (1) Due to the use of intermediate layerwise logits, ReSIDe is not applicable to non-DNN-based SIDs; however, the aggregation component of our method is applicable whenever multiple confidence scores can be extracted from the SID. (2) The performance of the preference optimization algorithm depends on the hold-out validation set. Thus, practitioners need to make sure the validation distribution is sufficiently varied as to not overfit to any particular covariate shift. (3) Lastly, the creation of the feature centroids requires access to samples from the training distribution. When this is not available, one would need to sample instead from an approximation to this distribution, which could introduce additional approximation errors into the method.

Acknowledgments and Disclosure of Funding

This work was supported by Reality Defender and conducted during Kaixiang Zheng's internship at Reality Defender.

References

Taiga Abe, Estefany Kelly Buchanan, Geoff Pleiss, Richard Zemel, and John P Cunningham. Deep ensembles work, but are they necessary? *Advances in Neural Information Processing Systems*, 35: 33646–33660, 2022.

Mohamed Ishmael Belghazi and David Lopez-Paz. What classifiers know what they don't? *arXiv preprint arXiv:2107.06217*, 2021.

Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 50(1):3–44, 2021.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=B1xsqj09Fm`.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

Luís Felipe Prates Cattelan and Danilo Silva. On selective classification under distribution shift. In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*, 2023.

Luís Felipe Prates Cattelan and Danilo Silva. How to fix a broken confidence estimator: Evaluating post-hoc methods for selective classification with deep neural networks. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024. URL `https://openreview.net/forum?id=IJBWLRCvYX`.

Ruoxin Chen, Junwei Xi, Zhiyuan Yan, Ke-Yue Zhang, Shuang Wu, Jingyi Xie, Xu Chen, Lei Xu, Isabel Guan, Taiping Yao, et al. Dual data alignment makes ai-generated image detector easier generalizable. *arXiv preprint arXiv:2505.14359*, 2025.

C Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1):41–46, 1970.

Charles Corbiere, Nicolas Thome, Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Perez. Confidence estimation via auxiliary models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6043–6055, 2021.

Davide Cozzolino, Giovanni Poggi, Matthias Nießner, and Luisa Verdoliva. Zero-shot detection of ai-generated images. In *European conference on computer vision*, pages 54–72. Springer, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Inderjit S Dhillon and Dharmendra S Modha. Concept decompositions for large sparse text data using clustering. *Machine learning*, 42(1):143–175, 2001.

Vojtech Franc, Daniel Prusa, and Vaclav Voracek. Optimal strategies for reject option classifiers. *Journal of Machine Learning Research*, 24(11):1–49, 2023.

Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pages 3247–3258. PMLR, 2020.

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

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30, 2017.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International conference on machine learning*, pages 2151–2159. PMLR, 2019.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Federica Granese, Marco Romanelli, Daniele Gorla, Catuscia Palamidessi, and Pablo Piantanida. Doctor: A simple method for detecting misclassification errors. *Advances in Neural Information Processing Systems*, 34:5669–5681, 2021.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10696–10706, 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

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

Langzhou He, Junyou Zhu, Fangxin Wang, Junhua Liu, Haoyan Xu, Yue Zhao, Philip S Yu, and Qitian Wu. Can molecular foundation models know what they don't know? a simple remedy with preference optimization. *arXiv preprint arXiv:2509.25509*, 2025.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=Hkg4TI9xl`.

Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8759–8773. PMLR, 17–23 Jul 2022.

Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: beyond empirical risk minimization. *Advances in neural information processing systems*, 33:19365–19376, 2020.

Christos Koutlis and Symeon Papadopoulos. Leveraging representations from intermediate encoder-blocks for synthetic image detection. In *European Conference on computer vision*, pages 394–411. Springer, 2024.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Yann Le Cun, Ofer Matan, Bernhard Boser, John S Denker, Don Henderson, Richard E Howard, Wayne Hubbard, LD Jacket, and Henry S Baird. Handwritten zip code recognition with multilayer networks. In *[1990] Proceedings. 10th International Conference on Pattern Recognition*, volume 2, pages 35–40. IEEE, 1990.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

Hengyue Liang, Le Peng, and Ju Sun. Selective classification under distribution shifts. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=dmxMGW6J7N`.

Tao Liu, Jun Cheng, and Shan Tan. Spectral bayesian uncertainty for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18166–18175, 2023.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

Ziyin Liu, Zhikang Wang, Paul Pu Liang, Russ R Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. Deep gamblers: Learning to abstain with portfolio theory. *Advances in Neural Information Processing Systems*, 32, 2019.

Anatol Maier and Christian Riess. Reliable out-of-distribution recognition of synthetic images. *Journal of Imaging*, 10(5):110, 2024.

Midjourney. Midjourney. `https://www.midjourney.com/`, 2022.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Relaxed softmax: Efficient confidence auto-calibration for safe pedestrian detection. *NIPS Workshop MLITS*, 2018.

Tai D Nguyen, Aref Azizpour, and Matthew C Stamm. Forensic self-descriptions are all you need for zero-shot detection, open-set source attribution, and clustering of ai-generated images. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3040–3050, 2025.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Jonas Ricker, Denis Lukovnikov, and Asja Fischer. Aeroblade: Training-free detection of latent diffusion images using autoencoder reconstruction error. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9130–9140, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019.

Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image generation models. In *Proceedings of the 2023 ACM SIGSAC conference on computer and communications security*, pages 3418–3432, 2023.

Ketan Rajshekhar Shahapure and Charles Nicholas. Cluster quality analysis using silhouette score. In *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*, pages 747–748. IEEE, 2020.

Maohao Shen, Yuheng Bu, Prasanna Sattigeri, Soumya Ghosh, Subhro Das, and Gregory Wornell. Post-hoc uncertainty learning using a dirichlet meta-model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9772–9781, 2023.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Sora. Sora 2. `https://openai.com/index/sora-2/`, 2025.

Mattias Teye, Hossein Azizpour, and Kevin Smith. Bayesian uncertainty estimation for batch normalized deep networks. In *International conference on machine learning*, pages 4907–4916. PMLR, 2018.

Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8695–8704, 2020.

Wukong. Wukong. `https://xihe.mindspore.cn/modelzoo/wukong`, 2022.

Guoxuan Xia and Christos-Savvas Bouganis. On the usefulness of deep ensemble diversity for out-of-distribution detection. *arXiv preprint arXiv:2207.07517*, 2022.

Zhiyuan Yan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. Ucf: Uncovering common features for generalizable deepfake detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22412–22423, 2023.

Rahul Yumlembam, Biju Issac, Nauman Aslam, Eaby Kollonoor Babu, Josh Collyer, and Fraser Kennedy. Detection of ai generated images using combined uncertainty measures and particle swarm optimised rejection mechanism: R. yumlembam et al. *Scientific Reports*, 2025.

Han Zhou, Jordy Van Landeghem, Teodora Popordanoska, and Matthew B Blaschko. A novel characterization of the population area under the risk coverage curve (aurc) and rates of finite sample estimators. In *International Conference on Machine Learning*, pages 79226–79253. PMLR, 2025.

Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. Genimage: A million-scale benchmark for detecting ai-generated image. *Advances in neural information processing systems*, 36:77771–77782, 2023.

Ke Zou, Zhihao Chen, Xuedong Yuan, Xiaojing Shen, Meng Wang, and Huazhu Fu. A review of uncertainty estimation and its application in medical imaging. *Meta-Radiology*, 1(1):100003, 2023.

A Related work

Synthetic image detection. With the rapid progress of modern generative models such as GANs and diffusion models, synthetic image detection has become increasingly important to prevent the misuse of realistic synthetic images. The goal of this task is to build a synthetic image detector (SID) that distinguishes synthetic images from real ones. I.e., given an input image, an SID is a binary classifier that predicts either "real" or "fake". Although some works show promising results for SIDs based on hand-crafted features [Frank et al., 2020, Cozzolino et al., 2024, Ricker et al., 2024], modern SIDs are predominately based on DNNs trained by supervised learning [Rossler et al., 2019, Wang et al., 2020, Yan et al., 2023, Sha et al., 2023, Koutlis and Papadopoulos, 2024, Maier and Riess, 2024], with many demonstrating near-perfect in-distribution (ID) detection performance. However, recent studies show that these DNN-based SIDs often fail to generalize beyond their training distribution [Yumlembam et al., 2025, Maier and Riess, 2024, Nguyen et al., 2025], making their reliability questionable in actual deployment. In real-world scenarios, inputs may undergo common covariate shifts such as blurring, JPEG compression, low resolution, adversarial perturbations, images generated by unseen generative models, or images containing unseen semantics. Under these conditions, the performance of SIDs can degrade significantly. To tackle this, recent work proposes to train SIDs with a reject option [Maier and Riess, 2024]; however, this approach requires retraining the SID, arguably less practical and efficient than post-hoc methods that don't modify the weights of existing DNN-based SIDs. Yumlembam et al. [2025] address the issue by out-of-distribution (OOD) detection, which is a weak form of selective classification (SC) under distribution shifts[3] [Liang et al., 2024]. Moreover, most existing works only focus on limited types of covariate shifts such as unseen generators and JPEG compression [Maier and Riess, 2024, Yumlembam et al., 2025, Nguyen et al., 2025]. In contrast, this paper proposes a post-hoc SC framework to address SID's reliability issue, considering six types of covariate shifts, thus being practical, efficient, general, and comprehensive.

Selective classification. Selective classification augments a standard classifier with a rejection option, allowing the model to abstain from making predictions on inputs it is uncertain about. Instead of forcing a prediction for every input, the model produces both a prediction and a confidence score given an input, and outputs the prediction only if the confidence exceeds a predefined threshold. This cautious and conservative prediction framework is often desirable in safety-critical scenarios such as medical diagnostics, autonomous driving, and the justice system [Zou et al., 2023, Neumann et al., 2018, Berk et al., 2021], where any misclassification could lead to severe consequences. Existing SC approaches can be broadly categorized into *training-based* methods and *post-hoc* methods. Training-based approaches [Geifman and El-Yaniv, 2019, Huang et al., 2020, Liu et al., 2019, Zhou et al., 2025] consider the classifier as part of the learning problem to optimize for the SC performance, while post-hoc methods [Franc et al., 2023, Liang et al., 2024] assume that the classifier is pretrained and fixed, thus focusing exclusively on confidence estimation based on a fixed model. Since retraining the classifier could be costly or infeasible in practice, post-hoc SC methods are particularly attractive in real-world deployment scenarios, thus being the primary focus of this paper.

Confidence estimation. Towards better confidence estimation in SC and other applications, various CSFs have been proposed in the literature. The first line of work focuses on ensemble methods [Lakshminarayanan et al., 2017, Teye et al., 2018, Liu et al., 2023], with Monte-Carlo dropout [Gal and Ghahramani, 2016] being a prominent example. However, ensemble methods necessitate multiple forward passes to provide a confidence estimate, which is considered inefficient in practice. Also, recent work suggests that ensemble methods may not contribute much to confidence estimation, but to building a better performing predictor [Abe et al., 2022, Xia and Bouganis, 2022]. Some other works propose to train auxiliary DNN models to serve as CSFs [Shen et al., 2023, Corbiere et al., 2021]. However, such DNNs can be as large as the base classifier in the SC system, making their training inefficient. Therefore, we shift our focus to another line of work on logit-based CSFs, which are arguably the most efficient class of CSFs. Logit-based CSFs compute confidence scores using classifier's output logits. For example, since DNNs routinely adopt softmax over logits to provide a probabilistic output, the maximum softmax probability (MSP) [Hendrycks and Gimpel, 2017], also known as the softmax response [Geifman and El-Yaniv, 2017], is naturally used as a logit-based CSF. Other logit-based CSFs include softmax margin (SM) [Belghazi and Lopez-Paz, 2021], the

[3]OOD detection aims to enhance the reliability of a classifier by rejecting OOD samples. However, OOD samples may be correctly classified, while ID samples may be wrongly classified. Thus, if our goal is to achieve reliable predictions by rejecting samples which are likely to be wrongly classified, we should directly adopt the SC framework, instead of relying on OOD detection with a misaligned objective.

max logit (ML) [Hendrycks et al., 2022], the logits margin (LM) [Le Cun et al., 1990, Liang et al., 2024], the negative entropy (NE) [Belghazi and Lopez-Paz, 2021], and the negative Gini index (NGI) [Granese et al., 2021]. In this paper, we evaluate the SC performance of these logit-based CSFs in the context of synthetic image detection under distribution shifts, and propose an approach to improve their performance.

B Reweighting for dataset mixture

Let the test set T be the union of H disjoint subsets, i.e., $T = T_1 \cup T_2 \cup \cdots \cup T_H$, where $|T| = N$ and $|T_m| = N_m$ for $m \in [H]$. We reweight samples such that each subset takes up a total mass of $1/H$, ensuring an even mixture. Concretely, the empirical coverage, the empirical selective risk, and AURC are computed by

$$\hat{\phi}_t(f, g, T) = \frac{1}{H} \sum_{m=1}^{H} \sum_{i=1}^{N_m} \mathbb{1}[g(x_{m,i}) \geq t]/N_m,$$

$$\hat{R}_t(f, g, T) = \frac{\frac{1}{H} \sum_{m=1}^{H} \sum_{i=1}^{N_m} l(f(x_{m,i}), y_{m,i}) \mathbb{1}[g(x_{m,i}) \geq t]/N_m}{\hat{\phi}_t(f, g, T)},$$

and

$$\text{AURC}(f, g, T) = \frac{1}{H} \sum_{m=1}^{H} \sum_{i=1}^{N_m} \hat{R}_{g(x_{m,i})}(f, g, T)/N_m,$$

respectively, where $(x_{m,i}, y_{m,i})$ denotes the *i*-th sample from the *m*-th subset. It can be seen that if a sample belongs to the *m*-th subset, its mass is $1/HN_m$.

In practice, our implementation of RC curve plotting and AURC computation follows some packages, e.g., *torch-uncertainty*[4], where we sort samples based on their confidence scores and perform (1) a cumulative sum over sample masses to get all coverage levels, (2) a cumulative sum of weighted errors, i.e., 0/1 losses weighted by corresponding sample masses, divided by corresponding coverages, to get error rates at all coverage levels, and (3) a weighted sum or error rates to get AURC. A PyTorch-style pseudo-code is provided in Algorithm 1.

Algorithm 1 PyTorch-style pseudo-code for reweighted AURC computation.

```
# confidence: a list of N confidence scores for all samples
# error_flag: a list of N 0/1 losses for all samples
# mass: a list of N masses for all samples

# Sort samples by confidence (descending)
sorted_conf, indices = torch.sort(confidence, descending=True)
sorted_error = error_flag[indices]
sorted_mass = mass[indices]

# Cumulative statistics
cum_error = torch.cumsum(sorted_error * sorted_mass, dim=0)
coverage = torch.cumsum(sorted_mass, dim=0)

# Selective risks at all coverages
risk = cum_error / coverage

# Reweighted AURC
AURC = (risk * sorted_mass).sum()
```

[4]https://github.com/ENSTA-U2IS-AI/ torch-uncertainty

C Proof of Theorem 1

Proof. Define the ranking error as $L_{rank}(g_{\boldsymbol{w}}, V) = \frac{1}{(M-n)n}\sum_{u^+\in U^+}\sum_{u^-\in U^-}\mathbb{1}[g_{\boldsymbol{w}}(u^-) \geq g_{\boldsymbol{w}}(u^+)]$. Since $\mathbb{1}[q \geq 0] \leq \frac{\ln(1+e^q)}{\ln 2}$ for any $q \in \mathbb{R}$, we have

$$L_{rank}(g_{\boldsymbol{w}}, V) \leq \frac{1}{\ln 2} L_{ReSIDe}(g_{\boldsymbol{w}}, V). \tag{9}$$

Then, we invoke the SELE loss $\Delta_{sele}(g_{\boldsymbol{w}}, V) = \frac{1}{M^2}\sum_{u^-\in U^-}\sum_{u\in V}\mathbb{1}[g_{\boldsymbol{w}}(u^-) \geq g_{\boldsymbol{w}}(u)]$ from Franc et al. [2023], and decompose it as follows

$$\begin{aligned}
\Delta_{sele}(g_{\boldsymbol{w}}, V) &= \frac{1}{M^2}\sum_{u^-\in U^-}\left[\sum_{u^+\in U^+}\mathbb{1}(g_{\boldsymbol{w}}(u^-) \geq g_{\boldsymbol{w}}(u^+)) + \sum_{\hat{u}^-\in U^-}\mathbb{1}(g_{\boldsymbol{w}}(u^-) \geq g_{\boldsymbol{w}}(\hat{u}^-))\right] \\
&= \frac{(M-n)n}{M^2} L_{rank}(g_{\boldsymbol{w}}, V) + \frac{1}{M^2}\sum_{u^-\in U^-}\sum_{\hat{u}^-\in U^-}\mathbb{1}(g_{\boldsymbol{w}}(u^-) \geq g_{\boldsymbol{w}}(\hat{u}^-)) \\
&\leq \frac{(M-n)n}{M^2} L_{rank}(g_{\boldsymbol{w}}, V) + \frac{n^2}{M^2},
\end{aligned} \tag{10}$$

where the equality in Eq. (10) holds if and only if all wrongly classified samples have the same confidence score, i.e., $g_{\boldsymbol{w}}(u^-) = g_{\boldsymbol{w}}(\hat{u}^-)$ for any $u^- \in U^-$ and $\hat{u}^- \in U^-$.

Based on Eq. (9) and Eq. (10), we have

$$\Delta_{sele}(g_{\boldsymbol{w}}, V) \leq \frac{(M-n)n}{M^2 \ln 2} L_{ReSIDe}(g_{\boldsymbol{w}}, V) + \frac{n^2}{M^2}. \tag{11}$$

Further applying Theorem 8 from Franc et al. [2023], which states $\text{AURC}(g_{\boldsymbol{w}}, V) < 2\Delta_{sele}(g_{\boldsymbol{w}}, V)$, we get

$$\text{AURC}(g_{\boldsymbol{w}}, V) < \frac{2(M-n)n}{M^2 \ln 2} L_{ReSIDe}(g_{\boldsymbol{w}}, V) + \frac{2n^2}{M^2}. \tag{12}$$

If we assume that each sample in U^- has a unique score, then we can tighten Eq. (10) to

$$\Delta_{sele}(g_{\boldsymbol{w}}, V) = \frac{(M-n)n}{M^2} L_{rank}(g_{\boldsymbol{w}}, V) + \frac{n(n+1)}{2M^2}, \tag{13}$$

and therefore we have

$$\text{AURC}(g_{\boldsymbol{w}}, V) < \frac{2(M-n)n}{M^2 \ln 2} L_{ReSIDe}(g_{\boldsymbol{w}}, V) + \frac{n(n+1)}{M^2}. \tag{14}$$

This completes the proof of Theorem 1. □

D Implementation details

Hold-out validation sets. The ID validation set follows the same distribution as the training set, i.e., containing images from the same sources and semantic classes, with in total 1000 images. The remaining six validation sets have various covariate shifts, each containing 1000 images as well:

- **Blur:** images are processed by Gaussian blurring with $\sigma = 5$;
- **Unseen semantic classes (Unseen Cls.):** images from 50 semantic classes not presented in training;
- **Low resolution (Low Res.):** images are resized while preserving the aspect ratio, such that the length of the shorter edge is 64;
- **FGSM:** images are perturbed using FGSM, with $\epsilon = 8/255$;
- **JPEG:** images are compressed with JPEG with $q = 30$;

- **Unseen generators (Unseen Gen.):** synthetic images are generated by GLIDE and VQDM, which are not presented in training.

Note that there's a specific reason why we train SIDs using synthetic images from ADM and BigGAN, validate using GLIDE and VQDM, and test using Wukong, Midjourney, Stable Diffusion V1.4, and Stable Diffusion V1.5. In fact, to reflect the realistic scenario, we order image generators chronologically, and use the oldest generators for training SIDs, relatively newer generators for validation, and latest generators for testing. Also, it's worth noting that images in the hold-out validation sets are neither used in the training of SIDs nor included in the test sets.

Details of ULP. For ResNet50, we conduct ULP after each residual block, and therefore $L = 16$. Note that we apply global average pooling (GAP) over feature maps to get the feature vector $\boldsymbol{h}_l$. Base on silhouette analysis, we get $[K_l]_{l=1}^{L} = [4, 3, 3, 4, 4, 4, 5, 4, 4, 4, 4, 3, 3, 2, 2, 2]$. It can be seen that the number of centroids generally decreases as the layer gets deeper. In fact, for all three blocks in the last stage of ResNet50, the number of centroids stays at 2, the same as the number of classes in synthetic image detection. This behavior is consistent with our intuition, where image representations are more dispersed at early DNN layers, while become concentrated to 2 clusters at deeper layers for binary classification.

For Swin-T, we conduct ULP after each swin transformer block, and therefore $L = 12$. Similarly, GAP is adopted over feature maps to get the feature vector. Base on silhouette analysis, we get $[K_l]_{l=1}^{L} = [3, 2, 2, 2, 3, 3, 3, 2, 2, 2, 2, 2]$. A similar trend is exhibited as ResNet50, where earlier layers has generally more centroids than deeper ones. Again, for the 5 deepest blocks in Swin-T, the number of centroids stays at 2, the same as the number of classes in synthetic image detection.

The training recipe of the preference optimization algorithm. To optimize $\boldsymbol{w}$ following Eq. 6, we adopt an Adam optimizer with a learning rate of 0.1. The batch size is 16, i.e., $|\mathcal{U}^+| = |\mathcal{U}^-| = 16$, and we train for 100 epochs. The training wall-clock time is in the order of units of minutes using an NVIDIA V100 GPU.

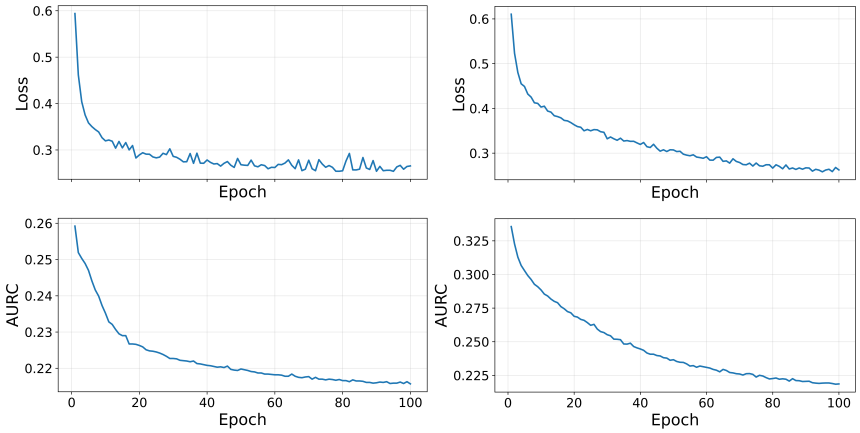
Note that we initialize $\boldsymbol{w}$ with $\boldsymbol{e}_{l^*}$, where $l^* = \arg\min_{l \in [L+1]} \text{AURC}(g_{\boldsymbol{e}_l}, V)$ is the index of the best layer, and $\boldsymbol{e}_l \in \mathbb{R}^{L+1}$ denotes a one-hot vector whose l-th entry is one. Since the preference optimization algorithm returns the checkpoint of $\boldsymbol{w}$ with the lowest validation AURC, this initialization enables an implicit fallback to the single best layer. If combining confidence scores from other layers do not provide AURC reduction compared to the single best layer, the final returned checkpoint will just be the initialization $\boldsymbol{e}_{l^*}$, and thus the performance of ReSIDe will be the same as the best layer. In a special case where $l^* = L + 1$ and $\boldsymbol{e}_{l^*}$ is returned by the preference optimization algorithm, the performance of ReSIDe will be the same as the baseline. This baseline fallback behavior explains why some results of ReSIDe in Tabs. 2 and 5 get 0.00% AURC reduction.

E Convergence curves

Fig. 4(a) plots the convergence curves of the ReSIDe loss and AURC on the validation set using ML as the underlying logit-based CSF, evaluated for ResNet50 in the case of Blur. It's shown that as the preference optimization progresses, both the ReSIDe loss and AURC decrease. This indicates that: (1) our proposed preference optimization algorithm based on Eq. (6) effectively minimizes the ReSIDe loss, and (2) as an upper bound of AURC, the minimization of the ReSIDe loss effectively lead to the minimization of AURC. A similar analysis is done for Swin-T in Fig. 4(b), with SM as the underlying logit-based CSF, also evaluated in the case of Blur. Again, both the ReSIDe loss and AURC converge successfully. Therefore, this analysis verifies the effectiveness of our preference optimization algorithm.

F Computational complexity

Tab.6 shows the inference latency of selective classifiers based on ReSIDe, measured on an NVIDIA L4 GPU. Compared to the baseline, i.e., selective classifiers based on the underlying final logit-based CSF, ReSIDe just slightly increases the latency by approximately 5% on both models, largely maintaining the efficiency of the SC system.



(a) ResNet50 (b) Swin-T

Figure 4: Convergence curves of the ReSIDe loss and AURC.

Table 6: Comparison of inference latency, measured in milliseconds per image (ms/img).

Classifier	CSF	Baseline	ReSIDe
ResNet50	MSP	1.996	2.091
	SM	2.004	2.103
	ML	1.994	2.066
	LM	2.008	2.085
	NE	2.002	2.096
	NGI	2.009	2.113
Swin-T	MSP	2.705	2.856
	SM	2.730	2.907
	ML	2.717	2.873
	LM	2.719	2.894
	NE	2.704	2.886
	NGI	2.735	2.908

G Generalization to unknown deployment shifts

In Section 6, we focus on evaluating ReSIDe under known types of shifts with unknown test-time specifications (e.g., JPEG quality factor, resizing ratio, unseen generator type, etc.). Here, we further study how ReSIDe generalizes to shift types not specified in the validation set. To this end, we transfer the layer aggregation weights $\boldsymbol{w}$ optimized for each of the 6 individual validation distribution to the mixture distribution, and present results in Table 7. Overall, 54 out of 72 individual-to-mixture transfers match or outperform the baseline. The transfer from Unseen Cls. improves all 12 classifier-CSF cases, and the transfer from Unseen Gen. is non-degrading in all 12. This shows that ReSIDe is decently robust to unknown threat modes not presented in the validation set.

H Results on EvalGen

We conduct evaluation on EvalGen [Chen et al., 2025], an additional benchmark containing images from more recent generators. Because EvalGen contains only synthetic images, we pair it with natural images from the ImageNet test set to form a balanced test set. For evaluation, we directly use the layer aggregation weights optimized on the GenImage-based Unseen Gen. validation set, instead of optimizing again on EvalGen. Results are shown in Table 8. Similar to the Unseen Gen. results in Table 2, ReSIDe matches or outperforms the baseline in all 12 cases on EvalGen.

Table 7: Comparison of AURCs. "Mix" denotes ReSIDe tuned directly on the mixture and is shown as a reference. Each column to the right of "Mix" corresponds to using a ReSIDe validation set consisting of only that type of shift, but still evaluating the performance over all shifts. **Bold** entries highlight individual-to-mixture transfers that match or outperform the baseline.

Classifier	CSF	Baseline	Mix	Blur	Unseen Cls.	Low Res.	FGSM	JPEG	Unseen Gen.
ResNet50	MSP	0.3563	0.2004	0.3645	**0.2554**	**0.3063**	**0.3354**	**0.3563**	**0.2554**
	SM	0.3563	0.2072	0.3620	**0.2418**	**0.2732**	**0.3202**	**0.3563**	**0.2554**
	ML	0.3635	0.2266	**0.3548**	**0.2755**	0.4263	**0.3082**	0.3850	**0.3635**
	LM	0.3563	0.2094	**0.3391**	**0.2516**	**0.2740**	**0.3375**	0.3705	**0.2554**
	NE	0.3563	0.1972	**0.3478**	**0.2373**	**0.3126**	**0.3416**	**0.3563**	**0.2554**
	NGI	0.3563	0.2028	**0.3545**	**0.2343**	**0.3080**	**0.3367**	**0.3563**	**0.2554**
Swin-T	MSP	0.3347	0.1947	0.3349	**0.2381**	**0.2759**	0.3822	0.3630	**0.2266**
	SM	0.3347	0.1902	**0.3290**	**0.2381**	**0.2749**	0.3931	0.3683	**0.2266**
	ML	0.2868	0.2048	0.3435	**0.2220**	0.2874	0.4305	0.3129	**0.2868**
	LM	0.3347	0.1858	**0.3013**	**0.2381**	**0.2633**	0.3948	0.3568	**0.2266**
	NE	0.3347	0.1879	**0.3114**	**0.2381**	**0.2642**	0.3822	**0.3309**	**0.2266**
	NGI	0.3347	0.1845	**0.3102**	**0.2381**	**0.2754**	0.3822	**0.3289**	**0.2266**

Table 8: Comparison to final logit-based CSFs on EvalGen (AURC).

Classifier	CSF	Baseline	ReSIDe	Δ
ResNet50	MSP	0.2544	0.2468	2.99%
	SM	0.2544	0.2468	2.99%
	ML	0.2500	0.2500	0.00%
	LM	0.2544	0.2468	2.99%
	NE	0.2544	0.2468	2.99%
	NGI	0.2544	0.2468	2.99%
Swin-T	MSP	0.2352	0.2031	13.63%
	SM	0.2352	0.2031	13.63%
	ML	0.2475	0.2475	0.00%
	LM	0.2352	0.2031	13.63%
	NE	0.2352	0.2031	13.63%
	NGI	0.2352	0.2031	13.63%

Table 9: Comparison to final logit-based CSFs on a CLIP ViT-B/16 SID (AURC).

Classifier	CSF	Baseline	ReSIDe	Δ
CLIP ViT-B/16	MSP	0.1948	0.1625	16.60%
	SM	0.1949	0.1547	20.64%
	ML	0.2149	0.1623	24.49%
	LM	0.1948	0.1559	19.99%
	NE	0.1948	0.1473	24.42%
	NGI	0.1948	0.1467	24.68%

I Results on an additional SID backbone

In additional to the two classifiers used in the main paper, we further evaluate a CLIP ViT-B/16 image encoder [Radford et al., 2021], equipped with a binary classification head and fully fine-tuned on the GenImage training set as the SID, whose risk on the mixture distribution is 0.3239. For this model, we extract the CLS-token representation after each of its 12 transformer blocks and construct one intermediate-layer score from each representation. Table 9 shows the results for this model on the mixture distribution. It's shown that ReSIDe improves all six CSFs on CLIP ViT-B/16, with AURC reductions ranging from 16.60% to 24.68%. This, together with the results in the main paper, supports the effectiveness of ReSIDe across conventional CNN, hierarchical-transformer, and ViT-based foundation-model backbones.

J Comparison to additional baselines

Here, we compare ReSIDe to Yumlembam et al. [2025] and two other OOD-style baselines, namely Energy [Liu et al., 2020] and Mahalanobis [Lee et al., 2018], with the same experimental setup as in

Table 10: Comparison of AURCs.

Classifier	ReSIDe	Energy	Mahalanobis	Yumlembam et al. [2025]
ResNet50	0.1972 ~ 0.2266	0.3646	0.2401	0.2293
Swin-T	0.1845 ~ 0.2048	0.2800	0.2087	0.3201

Section 6. Table 10 shows AURCs on the mixture distribution. Note that we report ReSIDe's result as the range of AURC over six underlying CSFs. It's shown that even the worst ReSIDe result is better than the best baseline: 0.2266 versus 0.2293 on ResNet50, and 0.2048 versus 0.2087 on Swin-T. This further shows ReSIDe's superiority when compared against stronger, feature-based baselines.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are supported by the theoretical and experimental results presented in Sections 4, 5 and 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work are discussed in Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The proof of Theorem 1 is presented in Appendix C, with assumptions clearly stated.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental settings and full details are presented in Sections 4, 6 and Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code for ReSIDe will not be released, as it is proprietary; however, complete instructions to reproduce the method are clearly outlined in the paper and appendices. The GenImage dataset is open-source and publicly available, and we have detailed in the paper and appendices how to prepare covariate-shifted test sets and hold-out validation sets based on the GenImage dataset.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental settings/details are presented in Sections 4, 6 and Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The results for ReSIDe are averaged over 3 trials, with standard deviations reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resource information is available in Appendix D and F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The work has no potential negative societal impacts. Potential positive societal impacts have been discussed in Section 1.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release data or models, and thus the paper poses no risks for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credit and cite all used existing assets and adhere to their terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: There are no released new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.