Preliminary Written Exam Report (WPE)
AID: Adaptive Integration of Detectors for Safe AI with Language Models

Xinran Wang
Department of Computer Science and Engineering
University of Minnesota
wang8740@umn.edu

Abstract

As Large Language Models (LLMs) increasingly influence content generation across diverse platforms, there is a heightened urgency to regulate their outputs to ensure safe usage. However, defining "safety" is complex, given that entities across domains may interpret it through varied lenses and develop detectors from specific safety criteria. To address this complexity, we introduce the approach of Adaptive Integration of Detectors (AID) to orchestrate the strengths of multiple pretrained detectors to ensure comprehensive effectiveness in diverse scenarios. AID employs a Mixture-of-Experts (MoE) framework, wherein it dynamically assigns and learns data-adaptive weights for each detector using domain-specific annotated data and LLM-extracted features. We provide theoretical insights into why MoE can be effective by showing its optimality in a classical Neyman-Pearson setting. Our experimental studies using various detection tasks curated from benchmark datasets demonstrate AID's ability to synergistically combine the unique capabilities of individual detectors. For example, it is observed that AID can improve the area under the curve (AUC) by an absolute value of 0.07 to 0.21, with a median of 0.12, compared with the best individual detectors. The improvement is particularly significant for complex detection tasks that mix different unsafe data sources.

1 Introduction

Large language models (LLMs) have seen widespread use across diverse domains, including healthcare, education, finance (Wu et al., 2023), and more, due to their remarkable ability to process and generate human-like text. However, with the increasing deployment of LLMs, safety concerns have emerged, including issues of bias, misinformation, and potential ethical implications. To address these challenges, there is a pressing need for customized, scalable safety detection mechanisms specific to LLMs, enabling effective and responsible application in various domains.

A primary concern is ensuring the safety of model-generated content. This has motivated recent advancements in content safety detection that target specific types of unsafe content. Notable among these are: Perspective API (Lees et al., 2022), a private model developed by Jigsaw and Google to identify toxic comments for moderating online discussions platforms, HateBERT (Caselli et al., 2020), which is a BERT-based model specifically fine-tuned to detect hate speech. RoBERTa-based models for hate speech detection (Baruah et al., 2020; Ali et al., 2022; Xu and Liu, 2023), which are fine-tuned from the RoBERTa model using abusive language on social platforms, and Detoxify developed by Unitary (Hanu and Unitary team, 2020), an open-source tool to pretrain models to predict toxic comments.

However, the concept of "safety" in digital content can be multifaceted and subjective, varying across different domains and societal norms. The existing developed detection models are either closed-source or pretrained from a specific data source that focuses on a specific type of unsafe content. As a result, despite their individual strengths, these state-of-the-art detectors typically operate in isolation. This siloed approach leads to limited effectiveness when encountering complex content that spans multiple categories of unsafety. Furthermore, these models, once trained, do not adapt to the evolving nature of online discourse, leading to reduced efficacy over time. These challenges motivate our work to adaptively integrate these diverse detection mechanisms.

This paper proposes a perspective to address the safety detection through an Adaptive Integration of Detectors (AID), which seeks to leverage the strengths of various pretrained detectors for comprehensive and effective safety regulation in content generation. Specifically, we formulate the

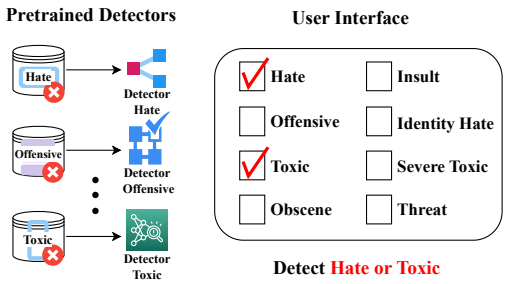

Figure 1: Illustration of the detection system that aims to detect user-specified set of unsafe tags.

problem as detecting whether a sentence should be tagged as "unsafe" based on user-selected unsafe aspects, such as insult, hate, offense, sex, and violence, each corresponding to a pretrained safety detector. Figure 1 shows the use scenario of our detection problem. Then, we train an AID model, which is inspired by the Mixture-of-Experts (MoE) framework. The experts represent many pretrained detectors, and we aim to learn a weighting scheme that aggregates detectors' results. In this way, we can accommodate various user-selected tasks by only adjusting the weights, which are data-adaptive, without intensive retraining.

The main contribution of this work is as follows:

- We develop an AID approach based on the MoE framework to integrate multiple content safety detectors. This approach parameterizes data-adaptive weights assigned to each detector to integrate and then learns those integration weights in a data-driven way.
- We give an optimality analysis of the proposed approach from a Bayesian perspective, which shows it can be significantly better than standard ensemble approaches.
- We conduct extensive experimental studies to demonstrate of AID's excellent performance using benchmark datasets "Toxic Comment Classification Challenge" and "Hate Speech and Offensive Language", for which we curated a variety of user-specified detection tasks that mix different unsafe tags. Our results show that AID can synergistically combine the unique capabilities of individual detectors, which corroborate our developed theory. It is observed that AID can improve the area under the curve (AUC) by an absolute value of 0.07 to 0.21, with a median of 0.12, compared with the best individual detectors. The improvement is particularly significant for complex detection tasks that mixes different unsafe data sources.
- We also propose data-free integration methods for the scenario where annotated data are unavailable. Our experiments show that the performance of data-free integration is not as effective as AID-based integration.

2 Related Work

Mixture-of-Experts Models. Existing approaches to AI safety often focus on single-task detection tasks, utilizing annotated data to train specialized detectors for identifying predefined safety tags. Our proposed approach seeks to extend beyond this by focusing on compound-task detection. Our developed approach aims to flag an input as positive if it is categorized as positive under any of the user-selected tags within a supported tag set. It achieves this by mixing the detection results offered by single-task detectors. This detection system can be regarded as a type of Mixture-of-Experts (MoE) architecture (Masoudnia and Ebrahimpour, 2014; Shazeer et al., 2017; Riquelme et al., 2021; Fedus et al., 2022; Chen et al., 2022). In deep learning, MoE refers to a process that learns a task through the division of labor among specialized neural networks. The existing literature focuses on the joint training of these specialized networks, known as "experts", along with a routing mechanism to determine the activation of experts for given inputs. MoE has also gained much research interest in training large foundation models with reduced memory costs. However, our approach deviates from this line of action by learning the way of integrating existing, pretrained detectors into the MoE architecture.

Safety Detection. LLMs have revolutionized multiple sectors by offering capabilities that closely mimic human-like text generation. However, the deployment of LLMs raises significant safety concerns, including the perpetuation of biases, dissemination of misinformation, and various ethical dilemmas. Recent efforts in content safety detection have yielded notable methods targeting specific unsafe content types. For instance, the Perspective API developed by Jigsaw and Google (Jigsaw and Conversation-AI, 2018) aims to identify toxic comments to aid moderation on online discussion platforms (Wang and Chang, 2022). HateBERT (Caselli et al., 2020) is a BERT-based model fine-tuned to specifically target hate speech. This model underscores the potential of transformer-

based architectures in identifying nuanced forms of online hate. Similarly, various RoBERTa-based models have been proposed for specific safety applications (Vidgen et al., 2020; Ali et al., 2022). Despite these advances, the concept of "safety" in digital content remains complex and subjective, varying across different contexts and societal norms. The siloed nature of existing detection models, often developed for specific content types and based on particular data sources, need to keep pace with the evolving dynamics of online discourse.

3 Problem Formulation

Setup and Notation. We formulate the problem as follows. Suppose the input is represented by a variable $x \in \tilde{X}$. A detection task is to decide whether the input is associated with a tag, denoted by t, that represents the safety aspect of interest. A detector associated with this task is defined as a function that maps from $\tilde{X}$ to $\tilde{Y} \triangleq \{0, 1\}$, written as $d_t : x \mapsto d_t(x)$, where $d_t(x) = 1$ means x should be tagged as positive t. Suppose $d(x)$ can be written as $d(x) = \mathbb{1}_{s(x)-\tau}$ for a score function s (Ding et al., 2018), in which varying choices of τ determine the tradeoffs between the false positive rate and detection power. A classical score function is $s(x) = \log p_t(x)$, where p_t is the density of x associated with positive t. A larger $s(x)$ is interpreted as more likelihood of x following p_t. While this logarithmic score has been widely used in decision theory and anomaly detection (Pang et al., 2021) due to its deep root in the uniformly most powerful test (Casella and Berger, 2021, Ch. 8) and Kullback-Leibler divergence (Shao et al., 2019; Wu et al., 2022). In practice, the distribution p_t is often approximated through generative models such as the autoencoders (Elkhalil et al., 2021; Bank et al., 2023). Our evaluation metric for detection performance is the area under the curve (AUC), which measures the two-dimensional area underneath the receiver operating characteristic (ROC) curve, which only depends on the score function s.

Formulation. Suppose a user can access a set of K pretrained detectors from several entities, e.g., companies and research labs that release models or their APIs, through private or public clouds platform such as Huggingface (Jain, 2022). Each detector d_k has been trained to detect a tag t_k, for $k \in [K]$. The union of all tags is denoted by $\mathcal{T} = \{t_k, k \in [K]\}$. Suppose a user is interested in detecting whether any given input is associated with a particular subset of unsafe tags $\mathcal{C} \subseteq \mathcal{T}$. Our problem is to develop a detection strategy to meet any user-specific need, namely to accurately flag an input as positive if it is associated with any tags within $\mathcal{C}$ (as shown in Figure 1).

To put this problem into perspective, let us consider two special cases as examples. In the first case, $\mathcal{C} = \{t\}$ is a singleton set corresponding to a classical detection problem that uses only the detector associated with t. Our insight is that detectors associated with tags in $\mathcal{T} - \mathcal{C}$ can possibly provide side information to enhance detection accuracy if they contain mutual information (from an information-theoretic perspective) with the tag of interest conditional on the input. As a result, theoretically, there should exist a way to utilize detectors associated with seemingly unrelated tags to contribute to the decision about t. For example, detecting whether content has 'toxicity' could be closely related to detecting 'violence'. The second case is $\mathcal{C} = \mathcal{T}$, which concerns a union of the existing tags available, e.g., detecting whether content falls into either 'toxic', 'violent', or 'offensive' as they may constitute all those that should be moderated. The two cases show the interest for integrating detectors for any user-specific task $\mathcal{C}$.

4 Adaptive Integration of Detectors (AID)

This section proposes 1) an approach to learn the integration mechanism, 2) a theoretical justification of using the MoE structure for detection, 3) an example to show why data-adaptive integration can perform significantly better than nonadaptive methods, and 4) natural baselines of data-free integrating detectors to be revisited in the experiments.

4.1 Description of AID

We construct a detector in the form of $d_{\mathcal{T},\mathcal{C}}(x) \triangleq \mathbb{1}_{s(x)>\tau}$, the subscript $\mathcal{C}$ highlights the user's interest of detection and s is in the form of

$$s(x; \Theta) \triangleq \sum_{k\in[K]} w_k(x; \Theta) \cdot s_k(x), \qquad (1)$$

a weighted sum of detectors associated with $\mathcal{T}$. Here, the weights $w_k(x; \Theta)$ are adaptive to x and satisfy $\sum_{k\in[K]} w_k(x; \Theta) = 1$. The τ is a threshold that, once sweeping from $-\infty$ to ∞, produces an ROC of the decision rule. We parameterize

$$w_k(x; \Theta) = \frac{\exp\{-\theta_k^{\mathrm{T}} h_k(x)\}}{\sum_{k\in[K]} \exp\{-\theta_k^{\mathrm{T}} h_k(x)\}}, \qquad (2)$$

where $h_k(x)$ is an embedding of x, e.g., from pretrained sentence-transformer (SBERT) mod-

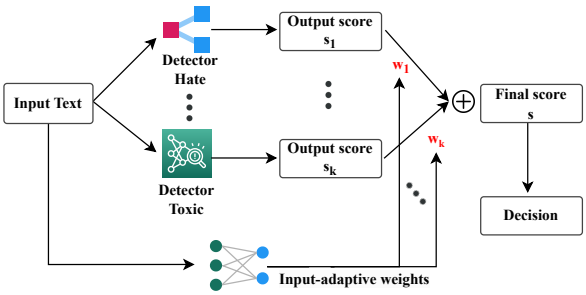

Figure 2: Illustration of the AID approach, which aims to construct an data-adaptive weighting of detectors to integrate for detecting a user-specified set of unsafe tags.

els (Reimers and Gurevych, 2019), and $\Theta = \{\theta_k, k \in [K]\}$ are learnable parameters.

To learn Θ, consider a set of data inputs annotated with tags in $\mathcal{C}$, denoted by $(x, y^t, t \in \mathcal{T})$. From this, we can create a set of n training data in the form of $(x, y^{\mathcal{C}})$ where $y^{\mathcal{C}} = 1$ if and only if there exists a $t \in \mathcal{C}$ such that $y^t = 1$.

Loss function. Optimizing directly for AUC, which is the area under the ROC curve, is not straightforward because the AUC itself is not differentiable with respect to the model parameters. Thus, we choose a loss function as a surrogate for the AUC. A common method is to use a ranking loss, like a pairwise ranking loss. This loss encourages the model to correctly rank positive samples higher than negative samples. Given that AUC is the probability a randomly chosen positive example is ranked more highly than a randomly chosen negative example (Fawcett, 2006), it is natural to minimize the following ranking loss to indirectly optimize for AUC:

$$\min_{\Theta} L(\Theta) \triangleq \sum_{(i,j):\ y_i^{\mathcal{C}}=1, y_j^{\mathcal{C}}=0} \log\left(1 + e^{-\lambda \cdot \delta_{i,j}(\Theta)}\right)$$

$$\text{with } \delta_{i,j}(\Theta) \triangleq s(x_i;\Theta) - s(x_j;\Theta)$$

where s was introduced in Equation (1) and λ is a tuning parameter. However, we found the above problem is computationally costly to run. To see that, consider optimizing the above loss with the stochastic gradient descent approach and a batch size of b. Each batch requires $O(b^2)$ computation and memory costs to evaluate the gradient of the loss function. Moreover, the optimization results could be sensitive to hyperparameters such as λ.

Alternatively, we propose to minimize the following loss:

$$\min_{\Theta} L(\Theta) \triangleq \big(\mathbb{E}\{D_{\text{neg}}\} + \mathbb{S}\{D_{\text{neg}}\}\big) - \big(\mathbb{E}\{D_{\text{pos}}\} - \mathbb{S}\{D_{\text{pos}}\}\big) \quad (3)$$

where $\mathbb{E}$ and $\mathbb{S}$ denote the empirical expectation and standard deviation of a dataset,

$$D_{\text{neg}}(\Theta) \triangleq \{s(x_j;\Theta) : j : y_j^{\mathcal{C}} = 0\}$$
$$D_{\text{pos}}(\Theta) \triangleq \{s(x_i;\Theta) : i : y_i^{\mathcal{C}} = 1\}.$$

The intuition of the above loss function is to maximize the gap between the average score of the positive example and that of the negative example, accounting for their uncertainty given a finite sample (as reflected by one standard deviation). From our experimental studies, the detection performance is not sensitive to the use of other multiplies of the standard deviation, such as two and three. With the above loss function, each batch would require only $O(b)$ computation and memory costs.

Figure 2 shows the AID approach. The pseudocode is summarized in Algorithm 1.

Algorithm 1 AID algorithm

Input: Detectors represented by functions $s_t, t \in \mathcal{T}$, input representations $h_t : x \mapsto h_t(x), t \in \mathcal{T}$, training data $(x_i, y_i^{\mathcal{C}})$, $i = 1, \ldots, n$, pertaining to the user-specified task

Parameter: Θ as introduced in (2)

Output: $x \mapsto s(x; \hat{\Theta})$

1: Initialize parameter $\theta_t = 0, t \in \mathcal{T}$
2: Run SGD to optimize the Objective (3)
3: **return** $\hat{\Theta}$

4.2 Why Using Linear Aggregations

It is natural to consider a broader form of aggregating the individual detectors' scores to target any user-specific detection task. We theoretically show that linear aggregation can be AUC-optimal from a Bayesian perspective when the data distributions associated with each tag are known.

First, we introduce some background notations needed for the theory. Recall the simple-versus-simple hypothesis testing problems. Let p_0 and p_1 be two distribution densities that represent "safe" and "unsafe" data. The classical Neyman-Pearson Lemma (Neyman and Pearson, 1933) states that for detecting the presence of an alternative hypothesis, $x \sim p_1$, against a null hypothesis, $x \sim p_0$, the likelihood ratio test is uniformly the most powerful,

giving the largest AUC. Specifically, one decides to reject the null, or claim "unsafe", if the statistic

$$\log \frac{p_1(x)}{p_0(x)} = \log p_1(x) - \log p_0(x) \quad (4)$$

is above a threshold τ. In our detection problem setup, the optimal score knowing the unsafe data source is from p_1 would be

$$s_1(x) \triangleq \log p_1(x) - \log p_0(x). \quad (5)$$

However, in a general setting where unsafe content could be generated from diverse sources, the notion of optimality relies on the formulation of the user-specified detection task. Consider a mixture of $p_1, \ldots, p_K$ that represent K unsafe data distributions, in the form of $\sum_{k\in[K]} w_k p_k$. Let $W \in [K]$ denote a Multinomial random variable with probabilities $w_k = \mathbb{P}(W = k)$, $k \in [K]$. Then, we can represent the user-specified unsafe data X and distribution as

$$X \sim p_W,\ W \sim \text{Multinomial}(w_1, \ldots, w_K). \quad (6)$$

Based on Equation (5), the optimal score involving a random data source W would be $S \triangleq \log p_W(x) - \log p_0(x)$.

Consider S as an unknown quantity, and let $\hat{S}(x)$ be an estimator of it given measurements x. Recall that the mean square risk is defined by $\mathbb{E}(\hat{S}(x) - S)^2$, where the expectation is taken over the joint distribution of S, x.

Theorem 1 *The Bayes estimate of the optimal score, namely the one that minimizes the Bayes risk among all estimators, is given by*

$$s^{Bayes}(x) = \sum_{k\in[K]} w_k(x) \log p_k(x) - \log p_0(x),$$

$$\textit{where } w_k(x) \triangleq \frac{w_k p_k(x)}{\sum_{j\in[K]} w_j p_j(x)} \quad (7)$$

for any given x.

Theorem 1 gives a theoretical justification for using a linear MoE architecture–it contains the best estimator (under square loss) of the score. Moreover, the optimal mixing weights $w(x)$ corresponding to Equation (1) can be interpreted as the posterior probability of x belonging to the unsafe distribution. This is intuitive as the user-specified task is to detect a mixture of individual unsafe distributions. Without knowing where the input x comes from, we use its posterior probability of belonging to the training data of each expert as a soft indicator, in which the prior probability is specified by $w_i, i \in [K]$.

4.3 Why Data-Adaptive Weights

Next, we provide an example showing that using data-adaptive weights can be significantly better than any non-adaptive weighted average in integrating detectors.

Consider the safe data distribution p_0 and two unsafe data distributions p_1, p_2 are Gaussian with means $0, \mu_1, \mu_2$, respectively, and unit variance. The user-specified task is to detect unsafe data drawn from an equal mixture of p_1 and p_2. In other words, we have $x \sim p_0$ under the null hypothesis, and $x \sim (p_1 + p_2)/2$ under the alternative hypothesis. Consider two integration schemes based on the MoE model: 1) data-adaptive weights following Theorem 1, and 2) non-adaptive weights $s^{\text{Avg}}(x) = \sum_{k=1}^{2} \frac{1}{2} \log p_k(x) - \log p_0(x)$.

Theorem 2 *Suppose the safe data input is generated from $x \sim p_0$ and the unsafe data input is from $x \in p_1$. Suppose $\mu_1 < 0 < |\mu_1| < \mu_2$ and $|\mu_2/\mu_1|$ is bounded by a constant. Using the data-adaptive integration introduced in Equation (7) with equal prior weights $w_1 = w_2 = 1/2$, the AUC converges to one as $\mu_2 + \mu_1$ and $-\mu_1$ converge to infinity. Meanwhile, using the non-adaptive, equal-weight integration, the AUC would be no larger than 0.5 offered by the random guess.*

The above theorem shows the necessity of integrating detectors adaptively to the inputs.

4.4 Data-Free Alternatives to AID

In case there is no annotated data for the user-specified detection task for AID training, we propose some data-free integration methods. They also provide natural baselines that we will revisit in the experimental evaluation. In these methods, we suppose the input x is represented by the SBERT embedding vector.

Equal-weight integration ("Avg"). We use $s(x) \triangleq \sum_{k\in[K]} s_k(x)/K$ for detection.

Max-score integration ("Max"). We use $s(x) \triangleq \max_{k\in[K]} s_k(x)$ for detection. The intuition is that the larger the score, the more tendency the input is generated from one of the unsafe distributions.

Similarity-based integration using input embeddings ("Similarity"). Suppose a user can access the mean of the training inputs' SBERT embeddings for each detector k, denoted by $\bar{x}_k$. The cosine similarity ("cos") between an input x and $\bar{x}_k$ is used to quantify the relevance of the kth detector. Then, the user uses $s(x) = \sum_{k\in[K]} w_k s_k(x)$, where w the Softmax applied to $\cos(\bar{x}_k, x), k \in [K]$.

Bayesian integration using input embeddings ("Bayes-Input"). Inspired by Theorem 1, we use data-adaptive weights as defined in (7), but approximating $w_k(x)$ by assuming $w_k = 1/K$, $p_k(x)$ is Gaussian whose mean and covariance matrix are estimated from training data for detector k.

Bayesian integration with variational autoencoder (VAE) embeddings ("Bayes-VAE"). We use a similar approach as the above, but approximating p_k using the Gaussian distribution of the VAE embeddings (Xu et al., 2017).

5 Experimental Study

5.1 Experimental Setups

Data sources. We use a public dataset called "Toxic Comment Classification Challenge" (Jigsaw and Conversation-AI, 2018) (referred to as *ToxicComment*), which contains a large number of Wikipedia comments. Each comment has been labeled by human raters for safety behavior, where the data curator names the safety as "toxicity". The subtypes of annotated toxicity includes: "toxic", "severe_toxic", "obscene", "threat", "insult", "identity_hate" (id-hate). In other words, each sentence, if unsafe, is annotated with one or more of the above six tags; otherwise, it is regarded as "safe". We also use a dataset called "Hate Speech and Offensive Language" (referred to as *HateOffensive*) made by the authors of (Davidson et al., 2017), which contains a large number of tweets originally collected from Twitter API and annotated by CrowdFlower (Van Pelt and Sorokin, 2012) workers. Each tweet in *HateOffensive* was annotated with one of three categories: hate speech ("hate"), offensive but not hate speech ("offensive"), or neither offensive nor hate speech ("safe").

User-specified safety detection task. In practice, a user may be interested in detecting unsafe sentences with specific contexts, so the definition of safety tags in the data source originally used for training safety detectors may not be relevant to the user. To simulate a complex real-world setting where a user's unsafe data distribution may deviate from pre-defined categories, we construct user-specified safety detection tasks by mixing data from the existing unsafe tags. For example, the unsafe data distribution of a user's interest consists of 50% "toxic" from the *ToxicComment* data source and 50% "offensive" from the *HateOffensive* data source. Accordingly, our performance evaluation of detectors is based on test data constructed for various user-specified safety detection tasks.

Data preparation. For each data source, we split it into three sets: pretraining, training, and testing. The pretraining set is used to pretrain safety detectors that are integrated for a variety of user-specified safety detection tasks. The training set is used to construct data for integrating pretrained detectors. Depending on the user-specific safety detection tasks that we simulate, the training data is constructed accordingly by mixing the annotated data from the original data sources. The testing set is used to evaluate the performance of integrated detectors and baselines. Its construction is in line with the training set to follow the same distribution as designated by the user's need. To avoid double use of data in both training and testing, we evenly split each data source into three sets at the beginning of all the experiments. For our ablation study of the influence of training size, we resample with replacement from the pre-split training set.

Pretrained detectors to integrate. To demonstrate the proposed approach of integrating detectors, we need to curate several pretrained detectors. The detector employs a hybrid model architecture that integrates a Variational Autoencoder (VAE) with embeddings derived from a transformer-based model (bert-base-uncased from Huggingface). Specifically, the VAE utilizes the averaged embeddings from the transformer model's last hidden state as its input. This setup enables the VAE to be trained on user-specific data deemed unsafe, without necessitating the inclusion of safe data during the training process. For detection purposes, a future input sentence is processed, and the VAE generates a score based on the negative log-likelihood of the output—the higher the score, the greater the likelihood of the input being safe. For the above data sources, we would have a total of 8 pretrained detectors. For AID integration, we use the embedding extracted from a public SBERT model (Reimers and Gurevych, 2019; Huggingface, 2023) as the representation of each input sentence.

Metrics. Each detector takes an input sentence and outputs a score. In practice, one needs to set a threshold to determine whether the score is sufficiently large to claim it as safe. As different thresholds lead to varying tradeoffs in Type I/II error rates, we use AUC as a quantitative measure to evaluate the performance of detectors.

5.2 User-Specified Safety Detection Tasks

We first summarize the performance of AID and two data-free integration approaches: Avg and Max

Method	toxic	severe	obscene	threat	insult	id-hate	hate	offensive
AID	**0.92**	**0.99**	**0.95**	**0.97**	**0.96**	**0.97**	**0.90**	**0.93**
Avg	0.76	0.85	0.80	0.80	0.79	0.78	0.67	0.70
Max	0.79	0.85	0.82	0.83	0.82	0.80	0.66	0.70
Similarity	0.75	0.84	0.79	0.79	0.79	0.76	0.65	0.71
Bayes-Input	0.51	0.63	0.51	0.70	0.52	0.76	0.57	0.76
Bayes-VAE	0.68	0.78	0.72	0.73	0.71	0.71	0.58	0.62
D_{toxic}	0.72	0.81	0.76	0.72	0.75	0.72	0.66	0.70
D_{severe}	0.82	0.92	0.86	0.88	0.85	0.85	0.69	0.74
D_{obscene}	0.73	0.83	0.77	0.75	0.76	0.74	0.68	0.73
D_{threat}	0.79	0.88	0.83	0.88	0.83	0.80	0.66	0.71
D_{insult}	0.70	0.80	0.74	0.72	0.73	0.72	0.65	0.68
$D_{\text{id-hate}}$	0.76	0.84	0.80	0.80	0.80	0.81	0.67	0.71
D_{hate}	0.73	0.80	0.77	0.74	0.77	0.76	0.63	0.65
$D_{\text{offensive}}$	0.69	0.75	0.73	0.74	0.72	0.71	0.58	0.61

Table 1: Performance comparison of AID that integrates pretrained detectors, baseline methods, and individual pretrained detectors, evaluated by AUC. The safety detection task is defined by declaring the sentences drawn from a tag shown as the column name.

Method	toxic/offensive	toxic/hate	toxic/severe	obscene/threat
AID	**0.92**	**0.91**	**0.95**	**0.96**
Avg	0.71	0.70	0.80	0.80
Max	0.74	0.72	0.82	0.83
Similarity	0.72	0.70	0.80	0.79
Bayes-Input	0.64	0.62	0.72	0.72
Bayes-VAE	0.55	0.60	0.56	0.61
D_{toxic}	0.70	0.69	0.76	0.74
D_{severe}	0.77	0.76	0.68	0.87
D_{obscene}	0.72	0.71	0.69	0.76
D_{threat}	0.74	0.73	0.69	0.86
D_{insult}	0.68	0.68	0.69	0.73
$D_{\text{id-hate}}$	0.73	0.72	0.68	0.80
D_{hate}	0.66	0.65	0.68	0.76
$D_{\text{offensive}}$	0.62	0.61	0.68	0.74

Table 2: Like Table 1, but with safety detection tasks defined by sentences drawn from a mixture of 50% tag A and 50% tag B indicated by the column name.

introduced in Subsection 4.4, and those pretrained detectors on the same detection tasks (denoted by D_{tag}). Other data-free methods will be studied in the ablation studies. In training AID, we use 2000 sentences randomly sampled from the data distribution as defined by the detection task and 10000 sentences from the same distribution for testing. The results are summarized in Table 1 and Table 2.

In Table 1, the detection tasks are defined by whether a sentence is associated with a particular unsafety tag as in the original data source. In Table 2, we curate several user-specified tasks by mixing data of different tags from the original annotated data sources. Specifically, the detection tasks are defined by whether a sentence is associated with either one of two unsafety tags, e.g., "toxic or offensive". We implement this by a sampling of 50% "toxic" and 50% "offensive" from the original data sources as the task-specific unsafe data.

The results show that the proposed integration method can achieve much better performance when compared with baseline methods. For example, Table 1 indicates that by integrating detectors whose associated tags are not directly relevant to the task could significantly boost the detection power. Table 2 show the inadequacy of a single detector to complicated tasks and how combining detectors could help gain. For example, on the task that specifies "toxic" and "offensive" as unsafe tags, the AID achieves an AUC of 0.92, while an individual detector could only achieve 0.7 or 0.62.

5.3 Sparse AID and Computational Analysis

This section analyzes the computational efficiency of the AID approach. The computation for each input can be divided into three main steps: 1) Representation, where a sentence is converted into a numerical vector, 2) AID Processing, where the vector undergoes a forward pass through the AID model to compute K weights, and 3) Score Aggregation, where scores from K detectors are calculated and aggregated to derive a final decision score. The AID Processing step as defined in Equation (2) incurs a constant time cost per input and is substantially less time-consuming than the Score Aggregation step. Notably, the latter's overall computational demand increases with the number of detectors K, which is also the computation required for the Avg and Max baseline methods.

To enhance efficiency, we introduce a variant of the AID method that sparsify the smaller weights to zero on a per-input basis, thereby only keeping $L < K$ non-zero weights $w_k(x; \Theta)$, which are then normalized to sum to one. This adaptation ensures that only L detectors are evaluated, significantly decreasing the computational load in the Score Aggregation step to a ratio of L/K.

Employing the same experimental setup as presented in Table 1, we conducted tests to assess both the accuracy and the computation time of the original AID utilizing all 8 detectors ("AID-8") and its adaptive variant that selectively activates only L detectors, for $L = 7, \ldots, 1$ ("AID-L"). In Figure 3, we report both the detection performance and computation time. The AUC was averaged across all the tasks. The computation time was measured in seconds, run on an A100 GPU, averaged over batches of 1024 inputs. Values are reported at a scale of 10^{-4}, and standard errors are within 10^{-5}. As shown in Figure 3, this adaptive pruning approach maintains performance integrity until the number of activated detectors is reduced to one. Moreover, implementing $L < K$ substantially low-

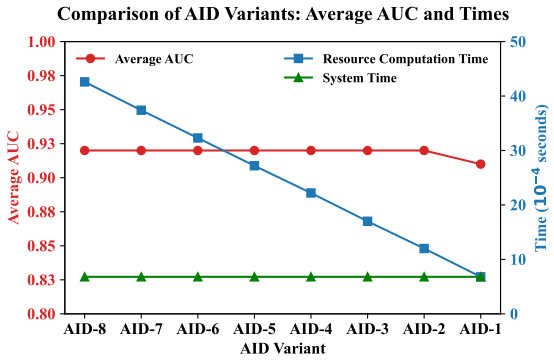

Figure 3: Performance, accumulated resource computation time, and system time comparison of AID and its variants that retain only L active detectors ("AID-L"), with the same setup as Table 1. System time stays almost constant as detectors can operate in parallel.

Method	toxic/offensive	toxic/hate	toxic/severe_toxic	obscene/threat
AID ($n = 200$)	0.89	0.89	0.93	0.94
AID ($n = 600$)	0.91	0.90	0.94	0.94
AID ($n = 2000$)	**0.92**	**0.91**	**0.95**	**0.96**
AID ($n = 6000$)	**0.92**	0.90	**0.95**	**0.96**
Avg	0.72	0.70	0.80	0.80
Max	0.74	0.72	0.82	0.83

Table 3: Ablation on sample sizes used for AID training.

ers the overall computational burden. Additionally, we recorded the time costs for executing the AID forward pass and generating per-detector scores, which were 1.7 and 5.1 seconds per 1024 inputs, respectively. As illustrated in Figure 3, the AID component introduces a negligible increment in computation time to the overall detection process.

5.4 Ablation Studies

Different training sizes. We vary the sample size used for training AID to $n = 200, 600, 2000, 6000$, and consider the same experimental setting as in Table 3. The results show that AID is not much sensitive to training sample size.

Other baseline methods. We perform an ablation study of all the data-free baseline methods in Section 4.4, using the same setting as Table 1. We report the results in Table 4. The results show that the data-free integration approaches do not perform better than the simpler approaches based on average or max score. We believe the "Bayes-Input" and "Bayes-VAE" approaches suffer from a poor estimation of the posterior probabilities $w_k(x) = \mathbb{P}(W = k \mid x)$, and the "Similarity" approach requires a temperature parameter, which is infeasible to train due to a lack of annotated data. Overall, this ablation study indicates the importance of learning the integration weights.

Sequential increase of detectors. In this ablation study, we explore the integration of detectors

Method	toxic	severe	obscene	threat	insult	id-hate	hate	offensive
AID	**0.92**	**0.99**	**0.95**	**0.97**	**0.96**	**0.97**	**0.90**	**0.93**
Avg	0.76	0.85	0.80	0.80	0.79	0.78	0.67	0.70
Max	0.79	0.85	0.82	0.83	0.82	0.80	0.66	0.70
Similarity	0.75	0.84	0.79	0.79	0.79	0.76	0.65	0.71
Bayes-Input	0.51	0.63	0.51	0.70	0.52	0.76	0.57	0.76
Bayes-VAE	0.68	0.78	0.72	0.73	0.71	0.71	0.58	0.62

Table 4: Extended performance comparison incorporating data-free integration methods. The safety detection task is defined by sentences drawn from tags indicated by the column names.

	toxic	+ severe	+ obscene	+ threat	+ insult	+ id-hate	+ hate	+ offensive
AID	0.70	**0.90**	**0.95**	**0.91**	**0.91**	**0.95**	**0.95**	**0.92**
Avg	0.70	0.73	0.80	0.74	0.73	0.81	0.81	0.71
Max	0.70	0.77	0.87	0.75	0.75	0.84	0.81	0.74
	offensive	+ hate	+ id-hate	+ insult	+ threat	+ obscene	+ severe	+ toxic
AID	0.62	**0.88**	**0.90**	**0.91**	**0.91**	**0.91**	**0.92**	**0.92**
Avg	0.62	0.64	0.67	0.68	0.70	0.70	0.71	0.71
Max	0.62	0.66	0.71	0.70	0.73	0.73	0.74	0.74

Table 5: Ablation studies on various user-selected detectors to integrate. Each row corresponds to a sequence of expanding set of pretrained detectors, as indicated by the column names.

in a sequential manner, as specified by users. The safety detection task is defined by declaring the sentences drawn from a mixture of 50% "toxic" and 50% "offensive". Our investigation involves two sets of pretrained detectors, detailed in Table 5, which are introduced incrementally. For example, the first row examines the integration starting with a single "toxic" detector, then progressively adding "severe" and other detectors in sequence. The findings indicate that augmenting the number of detectors typically enhances overall performance. However, it is observed that the addition of certain detectors may marginally impact the integration outcomes.

6 Conclusion

We introduced an AID approach to harness the collective strengths of pretrained detectors to enhance content safety detection. We theoretically elucidated the adaptability of its Mixture-of-Experts structure, and empirically demonstrated its effectiveness through several detection tasks and input distributions. However, we acknowledge the limitations of our work. The efficacy of AID is contingent upon the quality and diversity of the detectors it incorporates. When domain-specific data is scarce or lacks comprehensive representation, AID's ability to generalize across distinct domains, such as identifying moral hazards in healthcare or detecting money laundering in finance, may be compromised. Addressing these challenges will be a focus of our future research endeavors.

Impact Statements

This paper presents new approaches and insights aimed at enhancing content moderation practices within the realm of artificial intelligence. We expect our work will foster safer online environments, reduce the spread of harmful content, and support the development of more responsible AI. By advancing these content moderation practices, we anticipate substantial benefits for community well-being, user safety, and trust in digital platforms. While we have carefully considered the potential implications of our research, we believe that our work does not present any specific negative impacts that must be specifically highlighted here.

References

Raza Ali, Umar Farooq, Umair Arshad, Waseem Shahzad, and Mirza Omer Beg. 2022. Hate speech detection on twitter using transfer learning. *Computer Speech & Language*, 74:101365.

Dor Bank, Noam Koenigstein, and Raja Giryes. 2023. Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook*, pages 353–374.

Arup Baruah, Kaushik Das, Ferdous Barbhuiya, and Kuntal Dey. 2020. Aggression identification in english, hindi and bangla text using bert, roberta and svm. In *Proceedings of the second workshop on trolling, aggression and cyberbullying*, pages 76–82.

George Casella and Roger L Berger. 2021. *Statistical inference*. Cengage Learning.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2020. Hatebert: Retraining bert for abusive language detection in english. *arXiv preprint arXiv:2010.12472*.

Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. 2022. Towards understanding the mixture-of-experts layer in deep learning. *Advances in neural information processing systems*, 35:23049–23062.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Jie Ding, Vahid Tarokh, and Yuhong Yang. 2018. Model selection techniques: An overview. *IEEE Signal Processing Magazine*, 35(6):16–34.

Khalil Elkhalil, Ali Hasan, Jie Ding, Sina Farsiu, and Vahid Tarokh. 2021. Fisher auto-encoders. In *International Conference on Artificial Intelligence and Statistics*, pages 352–360. PMLR.

Tom Fawcett. 2006. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

William Fedus, Jeff Dean, and Barret Zoph. 2022. A review of sparse expert models in deep learning. *arXiv preprint arXiv:2209.01667*.

Laura Hanu and Unitary team. 2020. Detoxify. Github. https://github.com/unitaryai/detoxify.

Huggingface. 2023. Sentence transformer all-minilm-l6-v2. In *https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2*.

Shashank Mohan Jain. 2022. Hugging face. In *Introduction to Transformers for NLP: With the Hugging Face Library and Models to Solve Problems*, pages 51–67. Springer.

Edwin T Jaynes. 2003. *Probability theory: The logic of science*. Cambridge university press.

Jigsaw and Conversation-AI. 2018. Toxic comment classification challenge. In *https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data*.

Alyssa Lees, Vinh Q Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. 2022. A new generation of perspective api: Efficient multilingual character-level transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3197–3207.

Saeed Masoudnia and Reza Ebrahimpour. 2014. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42:275–293.

Jerzy Neyman and Egon Sharpe Pearson. 1933. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337.

Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 54(2):1–38.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. 2021. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595.

Stephane Shao, Pierre E Jacob, Jie Ding, and Vahid Tarokh. 2019. Bayesian model comparison with the hyvärinen score: Computation and consistency. *Journal of the American Statistical Association*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Chris Van Pelt and Alex Sorokin. 2012. Designing a scalable crowdsourcing platform. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 765–766.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2020. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv preprint arXiv:2012.15761*.

Yau-Shian Wang and Yingshan Chang. 2022. Toxicity detection with generative prompt-based inference. *arXiv preprint arXiv:2205.12390*.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Suya Wu, Enmao Diao, Khalil Elkhalil, Jie Ding, and Vahid Tarokh. 2022. Score-based hypothesis testing for unnormalized models. *IEEE Access*, 10:71936–71950.

Meijia Xu and Shuxian Liu. 2023. Rb_bg_mha: A roberta-based model with bi-gru and multi-head attention for chinese offensive language detection in social media. *Applied Sciences*, 13(19):11000.

Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. 2017. Variational autoencoder for semi-supervised text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

A Appendix: Experiments Related to Classifier-Based Detectors

Recall that our pretrained detectors were based on the BERT-VAE architecture. Specifically, for each unsafe tag from each data source, e.g., "offensive" from HateOffensive, we trained a VAE that takes the BERT embeddings of user-specific unsafe data as inputs. To use it for detection, we forward-pass a future input sentence and obtain the negative log likelihood from the VAE output as the score–the larger, the more likely to be safe. It is worth noting that the VAE-based pretrained detector does not require any safe data for training.

To demonstrate the generalizability of the proposed approach, we curate another type of pretrained detectors based on classification models. Specifically, we introduce a binary classifier based on a feed-forward neural network with a hidden layer, which takes the average of the embeddings from the last hidden state of BERT as input, and safe/unsafe label as output. Data are tokenized and padded using BERT's tokenizer to a maximum length of 512 tokens and the labels are balanced by randomly resampling. To use it for detection, we forward-pass a future input sentence and obtain the prediction score associated with the safety output, a value between 0 and 1–the larger, the more likely to be safe.

Classification-based pretrained detectors. We have run experiments on classification-based pretrained detectors, summarized in Table 6 and Table 7. In these experiments, the pretrained detectors perform well enough on most of the tasks that the gain brought by AID is incremental. Nevertheless, the performance of AID is consistent as before.

Method	toxic	severe_toxic	obscene	threat	insult	identity_hate	hate	offensive
AID	**0.96**	**1.00**	**0.98**	**0.99**	**0.98**	**0.99**	0.88	0.91
Avg	**0.96**	**1.00**	**0.98**	0.98	**0.98**	**0.99**	0.87	0.9
Max	0.94	0.99	0.97	0.97	0.96	0.96	0.87	0.91
Similarity	**0.96**	**1.00**	**0.98**	0.98	**0.98**	0.98	0.88	0.91
Bayes-Input	0.94	0.99	0.96	0.97	0.97	0.98	0.88	0.89
Bayes-VAE	0.96	0.99	**0.98**	**0.99**	**0.98**	0.98	0.82	0.83
toxic-clf	**0.96**	**1.00**	**0.98**	0.98	**0.98**	**0.99**	0.84	0.83
severe_toxic-clf	0.94	0.99	0.97	0.97	0.97	0.97	0.83	0.86
obscene-clf	**0.96**	**1.00**	**0.98**	0.97	**0.98**	**0.99**	0.84	0.86
threat-clf	0.93	0.99	0.95	**0.99**	0.96	0.96	0.81	0.83
insult-clf	**0.96**	**1.00**	**0.98**	0.97	**0.98**	**0.99**	0.84	0.84
identity_hate-clf	0.94	0.99	0.96	0.97	0.96	**0.99**	0.84	0.83
hate-clf	0.91	0.98	0.94	0.94	0.95	0.97	**0.9**	0.89
offensive-clf	0.88	0.99	0.94	0.92	0.93	0.92	0.87	**0.95**

Table 6: Performance comparison of the AID approach that integrates classifier-based pretrained detectors, baseline methods Avg and Max, and pretrained detectors, evaluated by ROC. The safety detection task is defined by declaring the sentences drawn from a tag indicated by the column name.

Integration of heterogeneous detectors. We show the performance of combining a mixture of classifier-based and VAE-based detectors. The results, as summarized in Table 8, show the AID performance is stable even though the detectors' output scores are of different scales.

B Appendix: Proofs of Technical Results

Proof 1 (Proof of Theorem 1) *The Bayes estimate (Jaynes, 2003, Ch. 6), namely the one that minimizes the mean square risk among all estimators, is given by the posterior mean*

$$s^{Bayes}(x) \stackrel{\Delta}{=} \mathbb{E}\{S \mid x\}. \tag{8}$$

Let $\mathbb{1}_{W=k}$ denote the indicator random variable for each $k \in [K]$. Then, we can rewrite S as

$$S = \sum_{k\in[K]} \mathbb{1}_{W=k} \cdot \log p_k(x) - \log p_0(x), \tag{9}$$

Method	toxic/offensive	toxic/hate	toxic/severe_toxic	obscene/threat
AID	0.9	**0.91**	**0.98**	**0.98**
Avg	**0.91**	0.9	**0.98**	**0.98**
Max	**0.91**	0.89	0.96	0.97
Similarity	**0.91**	**0.91**	**0.98**	**0.98**
Bayes-Input	0.90	0.90	0.96	0.96
Bayes-VAE	0.89	0.90	**0.98**	**0.98**
toxic-clf	0.89	0.9	**0.98**	**0.98**
severe_toxic-clf	0.87	0.86	0.97	0.97
obscene-clf	0.9	0.9	0.97	**0.98**
threat-clf	0.87	0.87	0.97	0.97
insult-clf	0.9	0.9	0.97	**0.98**
identity_hate-clf	0.85	0.86	0.97	0.96
hate-clf	0.89	0.9	0.96	0.94
offensive-clf	0.9	0.87	0.96	0.93

Table 7: Performance comparison of the AID approach that integrates classifer-based pretrained detectors, baseline methods Avg and Max, and pretrained detectors, evaluated by ROC. The safety detection task is defined by declaring the sentences drawn from a mixture of 50% tag A and 50% tag B indicated by the column name.

	toxic-clf + offensive-clf	toxic-vae + offensive-clf	toxic-clf + offensive-vae	toxic-clf + hate-vae	toxic-vae + hate-clf
AID	**0.94**	**0.92**	**0.9**	**0.91**	**0.92**
Avg	0.92	0.7	0.62	0.66	0.71
Max	0.91	0.91	**0.9**	0.89	0.89

offensive-vae	offensive-clf	toxic-vae	toxic-clf	hate-vae	hate-clf
0.62	0.91	0.7	0.89	0.64	0.65

Table 8: Performance comparison of the AID approach that integrates both classifer-based and VAE-based pretrained detectors, and baseline methods Avg and Max, evaluated by ROC. The first table summarizes the performance of integrating five user-selected pairs of detectors, and the second table summarizes the performance of pretrained detectors. The underlying safety task is defined based on a half-half mixture of "toxic" and "offensive" tags as from their respective data source.

and its posterior mean conditional on x is

$$\begin{aligned} s^{Bayes}(x) &= \mathbb{E}\{S \mid x\} \\ &= \mathbb{E}\Big\{ \sum_{k\in[K]} \mathbb{1}_{W=k} \cdot \log p_k(x) - \log p_0(x) \mid x \Big\} \\ &= \sum_{k\in[K]} \mathbb{E}\{\mathbb{1}_{W=k} \mid x\} \cdot \log p_k(x) - \log p_0(x) \\ &= \sum_{k\in[K]} \mathbb{P}\{W=k \mid x\} \cdot \log p_k(x) - \log p_0(x). \end{aligned} \tag{10}$$

By Bayes' Theorem, we have

$$\begin{aligned} \mathbb{P}\{W=k \mid x\} &= \frac{\mathbb{P}\{W=k\}p_i(x)}{\sum_{j\in[K]} \mathbb{P}\{W=j\}p_j(x)} \\ &= \frac{w_k p_k(x)}{\sum_{j\in[K]} w_j p_j(x)}. \end{aligned} \tag{11}$$

Taking Equation (11) to (10), we conclude the proof of Theorem 1.

Proof 2 (Proof of Theorem 2) *Based on the Gaussian assumptions of p_0, p_1, p_2, we have*

$$s^{Avg}(x) = -\sum_{k=1}^{2} \frac{1}{4}(x-\mu_i)^2 + \frac{1}{2}x^2 = (\mu_1+\mu_2)x/2 + c \tag{12}$$

where $c \triangleq -(\mu_1^2+\mu_2^2)/4$ is a constant that does not depend on x. Thus, the detection rule associated with the score function s^{Avg} is to identify the input as unsafe if $(\mu_1+\mu_2)x/2 > \eta$. Recall that the false alarm rate and detection power are respectively defined as

$$\alpha^{Avg}(\eta) = \mathbb{P}\{(\mu_1+\mu_2)x/2 > \eta \mid x \sim p_0\}, \tag{13}$$
$$\beta^{Avg}(\eta) = \mathbb{P}\{(\mu_1+\mu_2)x/2 > \eta \mid x \sim p_1\}. \tag{14}$$

Let $\phi : \mathbb{R} \to \mathbb{R}$ denote the cumulative distribution function of a standard Gaussian distribution. Then, under our setup of p_0 and p_1, and the assumption $\mu_1+\mu_2 > 0$, we can rewrite

$$\alpha^{Avg}(\eta) = \phi\left(-\frac{2\eta}{\mu_1+\mu_2}\right), \tag{15}$$
$$\beta^{Avg}(\eta) = \phi\left(-\frac{2\eta}{\mu_1+\mu_2} + \mu_1\right). \tag{16}$$

Thus, for $\mu_1 < 0$, we have $\alpha^{Avg} > \beta^{Avg}$ regardless of the values of η. Therefore, the AUC is no larger than 0.5.

On the other hand, the data-adaptive score can be written as

$$s^{Bayes}(x) = (w_1(x)\mu_1 + w_2(x)\mu_2)x - \frac{1}{2}(w_1(x)\mu_1^2 + w_2(x)\mu_2^2) \tag{17}$$

where the data-adaptive weights can be calculated as

$$\text{where } w_1(x) \triangleq \frac{p_1(x)/2}{\sum_{j\in[2]} p_j(x)/2} = \frac{p_1}{p_1(x)+p_2(x)} = \frac{1}{1+\exp\{(\mu_2-\mu_1)(x-(\mu_1+\mu_2)/2)\}},$$
$$w_2(x) = 1 - w_1(x).$$

In this case, the detection rule is to identify the input as unsafe if $(w_1(x)\mu_1 + w_2(x)\mu_2)x - \frac{1}{2}(w_1(x)\mu_1^2 + w_2(x)\mu_2^2) > \eta$ for any threshold η. Let A_1 denote the event that x is no larger than 0, which indicates that

$$x - (\mu_1+\mu_2)/2 \le -(\mu_1+\mu_2)/2 < 0, \tag{18}$$
$$w_1(x) \ge \frac{1}{1+\exp\{-(\mu_2^2-\mu_1^2)/2\}}. \tag{19}$$

For an arbitrarily small $\delta > 0$ such that

$$(1-\delta)\mu_1 + \delta\mu_2 < 0, \tag{20}$$

Inequality (19) implies that we have $w_1(x) \ge 1-\delta$ for all sufficiently large $\mu_2^2 - \mu_1^2$. Note that if $\mu_2+\mu_1$

goes to infinity, so does $\mu_2^2 - \mu_1^2$. Then, the detection power can be lower bounded by

$$\begin{aligned}
\beta^{Bayes}(\eta) &\triangleq \mathbb{P}\bigg\{(w_1(x)\mu_1 + w_2(x)\mu_2)x > \frac{1}{2}(w_1(x)\mu_1^2 + w_2(x)\mu_2^2) + \eta \mid x \sim p_1\bigg\} && (21)\\
&\geq \mathbb{P}\bigg\{(w_1(x)\mu_1 + w_2(x)\mu_2)x > \frac{1}{2}(w_1(x)\mu_1^2 + w_2(x)\mu_2^2) + \eta,\, A_1 \mid x \sim p_1\bigg\}\\
&\quad + \mathbb{P}\big\{A_1 \mid x \sim p_1\big\} - 1 && (22)\\
&\geq \mathbb{P}\bigg\{((1-\delta)\mu_1 + \delta\mu_2)x > \frac{1}{2}(1-\delta)\mu_1^2 + \delta\mu_2^2 + \eta,\, A_1 \mid x \sim p_1\bigg\} + \phi(-\mu_1) - 1 && (23)\\
&= \mathbb{P}\bigg\{x - \mu_1 < \frac{\frac{1}{2}((1-\delta)\mu_1^2 + \delta\mu_2^2) + \eta}{(1-\delta)\mu_1 + \delta\mu_2} - \mu_1, A_1 \mid x \sim p_1\bigg\} + \phi(-\mu_1) - 1 && (24)\\
&= \phi\bigg(\frac{\frac{1}{2}((1-\delta)\mu_1^2 + \delta\mu_2^2) + \eta}{(1-\delta)\mu_1 + \delta\mu_2} - \mu_1\bigg) + \phi(-\mu_1) - 1 && (25)\\
&\leq \phi\bigg(-\frac{1}{2}\mu_1 + \frac{\eta}{\mu_1}\bigg) + \phi(-\mu_1) - 1. && (26)
\end{aligned}$$

Likewise, let A_2 denote the event that x is no larger than $(\mu_1 + \mu_2)/4$, which indicates that

$$\begin{aligned}
x - (\mu_1 + \mu_2)/2 &\leq -(\mu_1 + \mu_2)/4 < 0, && (27)\\
w_1(x) &\geq \frac{1}{1 + \exp\{-(\mu_2^2 - \mu_1^2)/4\}}. && (28)
\end{aligned}$$

For any arbitrarily small $\delta > 0$, Inequality (28) implies that we have $w_1(x) \geq 1 - \delta$ for $\mu_2^2 - \mu_1^2$ larger than $c\log(\delta)$ for some constant c. The false alarm rate can be upper bounded by

$$\begin{aligned}
\alpha^{Bayes}(\eta) &\triangleq \mathbb{P}\bigg\{(w_1(x)\mu_1 + w_2(x)\mu_2)x > \frac{1}{2}(w_1(x)\mu_1^2 + w_2(x)\mu_2^2) + \eta \mid x \sim p_0\bigg\} && (29)\\
&\leq \mathbb{P}\bigg\{(w_1(x)\mu_1 + w_2(x)\mu_2)x > \frac{1}{2}(w_1(x)\mu_1^2 + w_2(x)\mu_2^2) + \eta,\, A_2 \mid x \sim p_0\bigg\}\\
&\quad + 1 - \mathbb{P}\big\{A_2 \mid x \sim p_0\big\} && (30)\\
&\leq \mathbb{P}\bigg\{\mu_2 x + \delta\frac{\mu_2^2 - \mu_1^2}{4} > \frac{1}{2}\mu_1^2 + \eta,\, A_2 \mid x \sim p_0\bigg\} + 1 - \phi\bigg(\frac{\mu_1 + \mu_2}{4}\bigg) && (31)\\
&\leq \mathbb{P}\bigg\{\mu_2 x + \delta\frac{\mu_2^2 - \mu_1^2}{4} > \frac{1}{2}\mu_1^2 + \eta \mid x \sim p_0\bigg\} + 1 - \phi\bigg(\frac{\mu_1 + \mu_2}{4}\bigg) && (32)\\
&\leq \mathbb{P}\bigg\{x > \frac{\mu_1^2}{2\mu_2} - \delta\frac{\mu_2^2 - \mu_1^2}{4\mu_2} + \frac{\eta}{\mu_2} \mid x \sim p_0\bigg\} + 1 - \phi\bigg(\frac{\mu_1 + \mu_2}{4}\bigg) && (33)\\
&= \phi\bigg(-\frac{\mu_1^2}{2\mu_2} - \frac{\eta}{\mu_2} + \delta\frac{\mu_1^2 - \mu_2^2}{4\mu_2}\bigg) + 1 - \phi\bigg(\frac{\mu_1 + \mu_2}{4}\bigg). && (34)
\end{aligned}$$

Recall that the AUC is defined as the area under the curve formed by $(\alpha^{Bayes}(\eta), \beta^{Bayes}(\eta))$ sweeping $\eta \in (-\infty, \infty)$. We let U be a uniform random variable that equals the first term on the right-hand side of Inequality (34). This induces a random variable η defined by

$$\eta = -\mu_2 \cdot \bigg(\phi^{-1}(U) + \frac{\mu_1^2}{2\mu_2} - \delta\frac{\mu_1^2 - \mu_2^2}{4\mu_2}\bigg), \tag{35}$$

where $Z \triangleq \phi^{-1}(U)$ follows the standard Gaussian by the definition of ϕ. Recall that the AUC can be equivalently written as $\mathbb{E}\{\beta^{Bayes}(\eta)\}$ where the expectation is over a uniformly distributed $\alpha^{Bayes}(\eta)$. Also, $1 - \phi((\mu_1 + \mu_2)/4)$ converges to one as $\mu_1 + \mu_2$ goes to infinity. Thus, using Inequalities (26) and (34),

we obtain

$$AUC = \mathbb{E}\{\beta^{Bayes}(\eta)\} \geq \mathbb{E}\phi\left(-\frac{1}{2}\mu_1 + \frac{\eta}{\mu_1}\right) + \phi(-\mu_1) - 1 + o(1) \tag{36}$$

$$= \mathbb{E}\phi\left(-\mu_1 + \delta\frac{\mu_1^2 - \mu_2^2}{4\mu_1} - \frac{\mu_2}{\mu_1}Z\right) + \phi(-\mu_1) - 1 + o(1), \tag{37}$$

where $o(1)$ is a small term in the asymptotic regime. From Equation (37), the derivation that $\delta(\mu_2^2 - \mu_1^2)$ can converge to zero, and the assumption that $|\mu_2/\mu_1|$ is bounded, we conclude that the AUC converges to one as $-\mu_1$ goes to infinity.