ComDet: Complex Domain Adaptation for AI-Generated Text Detection

1st Lei Jiang
School of Computer Science and Technology
University of Chinese Academy of Sciences
Beijing, China
jianglei232@mails.ucas.ac.cn

2nd Desheng Wu*
School of Economics and Management
University of Chinese Academy of Sciences
Beijing, China
dwu@ucas.ac.cn

3rd Xiaolong Zheng
Institute of Automation
Chinese Academy of Sciences
Beijing, China
xiaolong.zheng@ia.ac.cn

4th Cuicui Luo
International College
University of Chinese Academy of Sciences
Beijing, China
luocuicui@ucas.ac.cn

***Abstract*—Labeled data scarcity, diverse feature distributions, and adversarial attacks represent three significant challenges in the AI-generated text detection (AGTD) task. These challenges demand detectors with strong generalization and robustness; however, prior works typically address only one of them in isolation. To bridge this gap and meet the demands of real-world applications, we introduce a novel AGTD scenario termed Complex Domain Adaptation (CDA), which integrates all three challenges into a unified task. Furthermore, we observe that most existing methods struggle to perform effectively in the CDA scenario. To address this limitation, we propose ComDet, a novel framework built on an unsupervised domain adaptation foundation. ComDet incorporates a reconstruction module to enhance robustness against adversarial attacks and a feature enhancement module to extract richer, domain-specific representations. Extensive experiments demonstrate that ComDet significantly outperforms all baseline methods in the CDA scenario while achieving comparable or superior performance in other real-world settings.**

***Index Terms*—AI-generated text detection, domain adaptation, generalization**

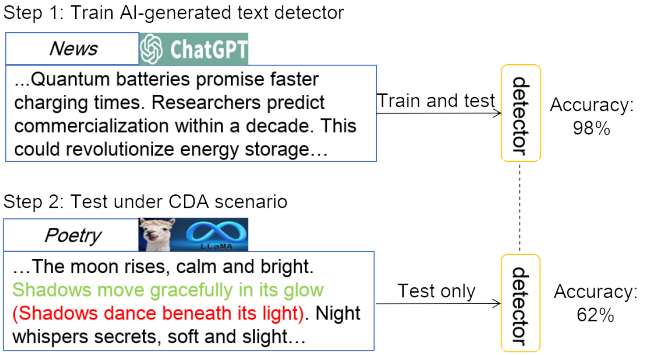

Fig. 1. An example of a CDA scenario. The text domain and generator of the target domain are different from those of the source domain, with adversarial perturbations present (red and green fonts represent the text before and after modification, respectively). Detectors that perform well in in-domain scenarios often exhibit poor detection performance in CDA scenarios.

I. Introduction

Large language models (LLMs) have recently driven major advances in natural language processing, enabling fluent text generation across applications like story writing, machine translation, and dialogue systems, thereby accelerating AI development and adoption [1]. However, their misuse poses risks to society and academia, such as by spreading fake news, rumors, or plagiarism [2]. This has made AGTD an increasingly urgent and critical research area.

Most current works in AGTD focus on in-domain scenarios, where the text distributions between the source and target domains are assumed to be similar [3]. In such cases, models trained on the source domain can transfer effectively to the target domain. However, real-world scenarios often involve diverse text domains (e.g., news, books, code) and texts generated by different LLMs [4], leading to significant distribution discrepancies that challenge the detector's generalization ability. Moreover, labeled data is often scarce and expensive in specific domains, restricting the applicability of supervised methods [5]. Consequently, detectors must acquire domain-invariant features and achieve efficient feature alignment across domains. Additionally, attackers may maliciously perturb generated text to evade detection [6], making robustness to adversarial perturbations another critical challenge for AGTD tasks. However, existing works lack a unified consideration of these scenarios. For example, ConDA [7] and Eagle [8] do not consider cross-generator detection, while RoBERTa-Ranker [9] does not address adversarial robustness.

Based on the above discussion, we summarize three significant challenges for AGTD in real-world scenarios: **labeled data scarcity**, **diverse feature distribution**, and **adversarial attacks**. Our intuition is that *integrating these three challenges will significantly increase the difficulty of the AGTD task.*

In practical applications, deploying an AGTD product must be cost-effective, allowing it to generalize across unlabeled data from various sources. It must also be designed to ensure robustness from a security perspective. Hence, we define a

more universal and challenging AGTD scenario, termed **CDA** (Complex Domain Adaptation). As present in Fig. 1, it refers to an unsupervised domain adaptation for binary classification AI-generated text detection task, simultaneously meeting the following requirements:

Labeled data scarcity: In the source domain, we have labeled (either "human" or "AI") text data; however, in the target domain, the text data is unlabeled. **Diverse feature distribution**: The text domain and generator of the target domain differ from those in the source domain. **Adversarial attack**: The texts to be detected in the target domain contain adversarial perturbations.

We identify the major difficulties of the CDA scenario as the **substantial inconsistency** in text feature distributions between the source and target domains, and the presence of **adversarial perturbations**. To address the issues discussed above, we propose a novel Complex Domain Adaptation for AI-generated text detection framework, named **ComDet**. We summarize our contributions as follows:

(1) We define a real-world AGTD scenario, CDA, which combines three significant challenges: labeled data scarcity, diverse feature distributions, and adversarial attacks into a unified task. Furthermore, we have observed that most existing methods struggle to perform effectively in the CDA scenario. The introduction of the CDA scenario imposes higher demands on the generalization and robustness of detection models.

(2) We propose a novel framework, ComDet, built on an unsupervised domain adaptation foundation. ComDet integrates multiple innovative modules, including a reconstruction module that enhances robustness by simulating adversarial attacks and denoising processes, and a feature enhancement module that enriches domain-specific feature representations.

(3) We conduct extensive experiments on publicly available datasets. Our results demonstrate that ComDet significantly outperforms existing baseline methods in the CDA scenario and achieves comparable or superior performance in other real-world settings. Source code is available at https://github.com/TristoneJiang/ComDet.

II. Related Work

Metric-based approaches for AGTD use proxy LLMs to extract features like entropy, token probability, perplexity, and cross-perplexity, capturing generation patterns [10], [11]. Similarity-based methods, such as higher cosine similarity in AI-generated text before and after revision, and the "invariance" under rewriting and antonym transformations, also prove effective [12]. Other approaches compare n-gram similarity with model-generated completions [13]. Perturbation-based strategies exploit the negative curvature of generative text [14]. While metric-based methods are training-free and generalizable, their prediction accuracy is often lower than supervised approaches, limiting practical effectiveness.

Our work falls under **model-based** detectors, which utilize supervised classifiers to identify AI-generated text. When direct access to generative models is unavailable, extracting deep features becomes a critical focus. Style representations, which capture writing style variations, have proven more effective than semantic representations [15]. Resampling proxy LLMs to estimate generation probability and generating pseudo-probability features [16] have also shown promise. Log probabilities from white-box models and transforming classification into sequence labeling tasks have been effective [17]. However, the manual extraction of features can result in the loss of important information. To improve robustness, some studies enhance model resilience to adversarial perturbations, using techniques like siamese calibrated reconstruction networks with Gaussian noise [18] and adversarial purification [19], [20]. To enhance the detector's generalization capabilities, some research combines self-supervised contrastive learning with domain-adversarial training to address text inconsistencies across domains [21]. However, current research rarely simultaneously considers both generalization and robustness. In comparison, our work aims to improve accuracy for AGTD tasks in the CDA scenario.

III. Proposed Method

A. Task Definition

Let $\mathcal{D}_S = \{(x_i^S, y_i^S)\}_{i=1}^{N_S}$ denote the labeled source domain dataset, where $x_i^S \in \mathcal{X}$ is a text sample and $y_i^S \in \{0, 1\}$ is its label indicating whether it is human-written (0) or AI-generated (1). Let $\mathcal{D}_T = \{x_j^T\}_{j=1}^{N_T}$ denote the unlabeled target domain dataset, where $x_j^T \in \mathcal{X}$, possibly perturbed by adversarial noise.

The goal of the **CDA** task is to learn a detection model $f : \mathcal{X} \to \{0, 1\}$, such that f achieves high classification accuracy on the unlabeled and distribution-shifted target domain $\mathcal{D}_T$, despite:

- **Labeled data scarcity**: The target domain $\mathcal{D}_T$ contains no labeled samples, i.e., $\nexists(x_j^T, y_j^T)$, making supervised learning infeasible on $\mathcal{D}_T$.
- **Diverse feature distribution**: The feature distributions between the source and target domains differ substantially, i.e., $\mathrm{Dist}(P_S(x), P_T(x)) \gg 0$, where P and Dist denote the distributions and distance function.
- **Adversarial attack**: Some target samples x_j^T are perturbed by an adversarial transformation δ, such that $x_j^T = x_j + \delta(x_j)$, where $\delta(x_j)$ is crafted to mislead the detector f, i.e., $f(x_j^T) \neq f(x_j)$.

B. Model Architecture

The framework of the proposed ComDet is shown in Fig. 2, which consists of the encoding module (EM), reconstruction module (RM), feature enhancement module (FEM), and feature alignment module (FAM).

1) Encoding Module: EM aims to capture the primary semantic features. For the input samples from the source domain x_i^S and target domain x_i^T, they are encoded through a pre-trained language model (PLM) to obtain initial representations:

$$p_{i[in]}^S = \mathrm{PLM}(x_i^S), \quad p_{i[in]}^T = \mathrm{PLM}(x_i^T), \tag{1}$$

where $p_{i[in]}^S$ and $p_{i[in]}^T$ have dimensions of d.

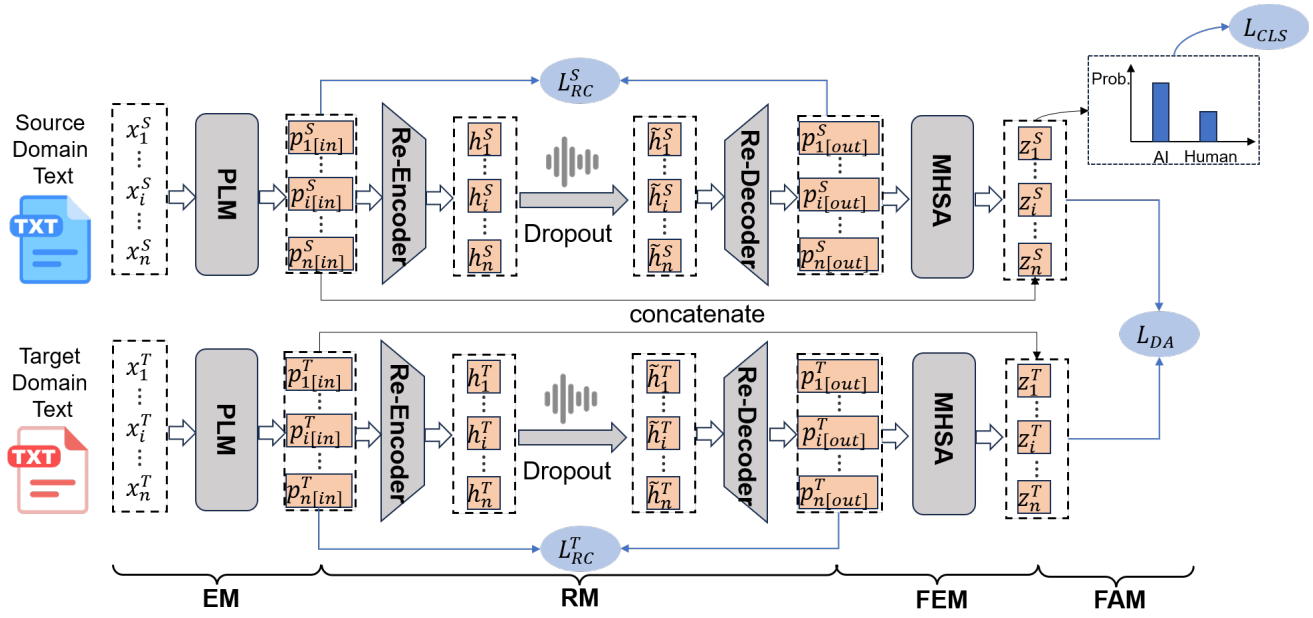

Fig. 2. Workflow of the ComDet framework. "EM": Encoding module, which encodes the input text into high-dimensional embeddings. "RM": Reconstruction module, which simulates the process of introducing and denoising real-world adversarial perturbations. "FEM": Feature enhancement module, which concatenates the features from the encoding and reconstruction modules. "FAM": Feature alignment module, which aligns the feature distributions between the source and target domains. During the inference phase, only the below branch of ComDet is required.

2) Reconstruction Module: The Reconstruction Module (RM) is designed to simulate the process of injecting and denoising real-world adversarial perturbations. It consists of three components: a re-encoder (RE), a dropout layer, and a re-decoder (RD). RM follows the denoising autoencoder paradigm [22], where the noise-removal principle has been successfully demonstrated in diffusion-based models [23].

RE maps the initial token representations into a higher-dimensional latent space using a multi-layer perceptron (MLP) for feature transformation:

$$\begin{aligned} h_i^S &= \mathrm{RE}(p_{i[\mathrm{in}]}^S) = \mathrm{ReLU}(W_{en}^S p_{i[\mathrm{in}]}^S + B_{en}^S), \\ h_i^T &= \mathrm{RE}(p_{i[\mathrm{in}]}^T) = \mathrm{ReLU}(W_{en}^T p_{i[\mathrm{in}]}^T + B_{en}^T), \end{aligned} \tag{2}$$

where $W_{en}^S, W_{en}^T \in \mathbb{R}^{r\times d}$, $B_{en}^S, B_{en}^T \in \mathbb{R}^r$, and r denotes the hidden dimension.

To simulate adversarial attacks, the latent representations h_i^S and h_i^T are processed by a dropout operation with a per-element retention probability γ. This operation randomly deactivates neurons and can be interpreted as injecting Gaussian noise into the representations. To illustrate this more clearly, we present a theoretical analysis below, following the derivation of [24].

Let a latent vector be $\mathbf{v} = [v_1, v_2, \ldots, v_r]$ and its dropout-modified version be $\tilde{\mathbf{v}} = \mathrm{dropout}(\mathbf{v}, \gamma)$, where the dropout operation is defined element-wise as:

$$\tilde{v}_i = \begin{cases} \frac{v_i}{\gamma_i}, & \text{w.p. } \gamma_i, \\ 0, & \text{w.p. } 1-\gamma_i, \end{cases} \tag{3}$$

with $\gamma_i \in (0, 1]$ being the retention probability for the i-th element. This ensures that $\mathbb{E}[\tilde{v}_i] = v_i$ and $\mathrm{Var}[\tilde{v}_i] = \frac{1-\gamma_i}{\gamma_i} v_i^2$.

Now consider a linear transformation applied to the dropout-perturbed input:

$$o_{ij} = \mathbf{W}_i^j \tilde{\mathbf{v}} + b_{ij} = \sum_{k=1}^{r} W_{i,k}^j \tilde{v}_k + b_{ij}, \tag{4}$$

where $\mathbf{W}_i^j \in \mathbb{R}^r$ and b_{ij} denote the weights and bias of the j-th neuron in the i-th layer. Since each $\tilde{v}_k$ is a random variable with variance $\frac{1-\gamma_k}{\gamma_k} v_k^2$, the variance of o_{ij} can be written as:

$$\mathrm{Var}[o_{ij}] = \sum_{k=1}^{r} \left(W_{i,k}^j v_k\right)^2 \cdot \frac{1-\gamma_k}{\gamma_k}. \tag{5}$$

As the dimensionality r increases, the Central Limit Theorem implies that o_{ij} approaches a normal distribution:

$$o_{ij} \sim \mathcal{N}(\mu_{o_{ij}}, \sigma_{o_{ij}}^2),$$

where $\mu_{o_{ij}} = \mathbf{W}_i^j \mathbf{v} + b_{ij}$ and $\sigma_{o_{ij}}^2 = \mathrm{Var}[o_{ij}]$.

Hence, the dropout-induced randomness can be equivalently represented as additive Gaussian noise:

$$o_{ij} = \mathbf{W}_i^j \mathbf{v} + b_{ij} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_{v_{ij}}^2), \tag{6}$$

Thus, the dropout operation effectively injects zero-mean Gaussian noise into the layer outputs, with the variance controlled by the retention probability γ.

This theoretical perspective shows that dropout introduces "adaptive noise", which can simulate a broader spectrum of adversarial perturbations compared to fixed-form word-level transformations [19]. While previous work such as [18] employs additional "semantic" and "perturbation" terms, dropout

provides a lightweight alternative by avoiding extra parameters and regularization terms, thus simplifying training. The perturbed representations are:

$$\tilde{h}_i^S = \text{dropout}(h_i^S, \gamma), \quad \tilde{h}_i^T = \text{dropout}(h_i^T, \gamma). \tag{7}$$

RD reconstructs the original token representations from the noisy latent space using another MLP layer:

$$\begin{aligned} p_{i[\text{out}]}^S &= \text{RD}(\tilde{h}_i^S) = \text{ReLU}(W_{de}^S \tilde{h}_i^S + B_{de}^S), \\ p_{i[\text{out}]}^T &= \text{RD}(\tilde{h}_i^T) = \text{ReLU}(W_{de}^T \tilde{h}_i^T + B_{de}^T), \end{aligned} \tag{8}$$

where $W_{de}^S, W_{de}^T \in \mathbb{R}^{d \times r}$ and $B_{de}^S, B_{de}^T \in \mathbb{R}^d$.

3) Feature Enhancement Module: We employ a multi-head self-attention (MHSA) [25] mechanism to enhance the domain-specific representations. Combining shallow and deep feature fusion with MHSA's global modeling capability improves domain-invariant feature extraction. For the source and target domain inputs $p_{i[out]}^S$ and $p_{i[out]}^T$, MHSA computes self-attention for each head. Let the number of attention heads denote m, and the dimension of each head $d_k = d/m$. The expression of MHSA is as follows:

$$\text{MHSA}(p_{i[out]}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_m) W_O, \tag{9}$$

where the computation for each head is given by:

$$\text{head}_k = \text{softmax}\left(\frac{Q_k K_k^T}{\sqrt{d_k}}\right) V_k. \tag{10}$$

Here $Q_k = p_{i[out]} W_Q^k$, $K_k = p_{i[out]} W_K^k$, $V_k = p_{i[out]} W_V^k$ are the linear transformations of the query, key, and value, respectively; $W_Q^k, W_K^k, W_V^k \in \mathbb{R}^{d \times d_k}$ and $W_O \in \mathbb{R}^{d \times d}$ are learnable parameters.

The enhanced features from MHSA are concatenated with the initial representations to form the final domain-specific feature representation:

$$\begin{aligned} z_i^S &= p_{i[in]}^S \oplus \text{MHSA}(p_{i[out]}^S), \\ z_i^T &= p_{i[in]}^T \oplus \text{MHSA}(p_{i[out]}^T). \end{aligned} \tag{11}$$

4) Feature Alignment Module: The domain-specific feature representations z_i^S and z_i^T from the source and target domains are passed through the domain alignment module. The alignment process is constrained by the Kullback-Leibler (KL) divergence [26] loss L_{DA}, which ensures distribution consistency, as described in (14).

C. Training Objective

The training objective of the proposed framework consists of three primary components: reconstruction loss L_{RC}, classification loss L_{CLS}, and domain alignment loss L_{DA}.

The reconstruction loss enforces consistency between the input representation $p_{i[in]}$ and the output representation $p_{i[out]}$ after passing through the reconstruction module. This loss is computed using the mean squared error:

$$\begin{aligned} L_{RC} = &\frac{1}{N} \sum_{i=1}^{N} \left\| p_{i[in]}^S - p_{i[out]}^S \right\|^2 \\ &+ \frac{1}{M} \sum_{i=1}^{M} \left\| p_{i[in]}^T - p_{i[out]}^T \right\|^2, \end{aligned} \tag{12}$$

where N and M represent the number of samples in the source and target domains, respectively.

The classification loss is employed for supervised learning on the source domain samples and is computed using binary cross-entropy loss:

$$L_{CLS} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right], \tag{13}$$

where y_i denotes the true label (0 or 1), and $\hat{y}_i$ represents the predicted probability from the model.

To mitigate the discrepancy between the feature distributions of the source and target domains and learn domain-invariant feature representations, we employ KL divergence to align the distributions $p(z_i^S)$ and $p(z_i^T)$ of the source and target domains, respectively. The KL divergence is defined as:

$$\text{KL}\left(p(z_i^S) \parallel p(z_i^T)\right) = \sum_{j=1}^{d} p_j(z_i^S) \log \frac{p_j(z_i^S)}{p_j(z_i^T)}, \tag{14}$$

where $p_j(z_i^S)$ and $p_j(z_i^T)$ denote the softmax-normalized feature probabilities for the source and target domains, computed as:

$$p_j(z_i) = \frac{\exp(z_{ij})}{\sum_{k=1}^{d} \exp(z_{ik})} \tag{15}$$

Finally, the domain alignment loss is defined as the average KL divergence across all N paired examples:

$$\mathcal{L}_{DA} = \frac{1}{N} \sum_{i=1}^{N} \text{KL}\left(p(z_i^S) \parallel p(z_i^T)\right). \tag{16}$$

The overall objective function is the weighted sum of the reconstruction loss, classification loss, and domain alignment loss:

$$L = \lambda_{RC} L_{RC} + \lambda_{CLS} L_{CLS} + \lambda_{DA} L_{DA}, \tag{17}$$

where λ_{RC}, λ_{CLS}, and λ_{DA} are the weight coefficients that control the relative importance of each loss term.

Aligned with the two major difficulties we identified for the CDA scenario, the first term of (17) is designed to address the issue of adversarial perturbations, while the second and third terms aim to mitigate the problem of substantial inconsistency in text feature distributions.

IV. Experiments and Main Results

A. Settings

We randomly select subsets from three distinct text domains of the publicly available dataset introduced by [27]: News (NW), Books Summary (BS), and Poetry (PT). In this dataset, the positive samples for NW, BS, and PT are generated by GPT-4 [28], ChatGPT [29], and LLaMA-Chat [30], respectively, based on specific prompts, while the negative samples consist of human-written texts. The training and testing datasets for each subset comprise 2000 and 1000 samples, respectively, with a balanced 1:1 ratio of positive and negative samples. Additionally, we employ paraphrased texts generated by the T5-11B model [27]. Our encoding module

TABLE I

PERFORMANCE COMPARISON OF DIFFERENT METHODS ACROSS VARIOUS SCENARIOS (NW→BS). CDA RESULTS ARE ANNOTATED WITH THEIR DEVIATION FROM THE MINIMUM VALUE AMONG OTHER SCENARIOS, AND THE SAME APPLIES TO TABLE II.

Category	Method	Base		Cross-G		Cross-D		Attack		CDA	
		F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
Metric-based	GLTR	87.2	94.4	88.4	96.5	85.1	94.8	85.1	95.6	83.9	97.1
	Log-Rank	89.0	98.6	87.5	97.5	86.3	94.6	87.6	95.8	88.8	95.9
	DetectGPT	90.8	98.6	90.0	99.2	90.5	99.3	90.2	97.1	90.0	97.6
	DNA-GPT	90.1	96.0	91.1	97.7	89.1	96.5	89.3	96.6	87.0	95.8
Model-based	RoBERTa	98.8	99.9	91.9	97.3	76.3	89.8	81.7	92.6	61.8 (−14.5)	88.5 (−1.3)
	RMLM	98.9	99.9	92.5	97.3	77.9	96.3	88.8	97.9	65.8 (−12.1)	89.3 (−7.0)
	RADAR	99.1	**100**	92.2	98.5	80.4	90.7	98.2	**100**	69.4 (−11.0)	82.2 (−8.5)
	SCRN	99.4	**100**	92.7	98.5	78.3	88.5	98.1	99.2	68.2 (−10.1)	88.3 (−0.2)
	ConDA	99.2	**100**	95.7	99.0	95.5	98.4	87.7	97.2	83.8 (−3.9)	92.4 (−4.8)
	OURS	**99.7**	99.9	**98.5**	**99.8**	**98.3**	**99.7**	**98.3**	99.8	**96.2** (−2.1)	**99.3** (−0.4)

TABLE II

PERFORMANCE COMPARISON OF DIFFERENT METHODS ACROSS VARIOUS SCENARIOS (NW→PT).

Category	Method	Base		Cross-G		Cross-D		Attack		CDA	
		F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
Metric-based	GLTR	85.3	93.5	84.7	95.3	87.0	95.2	82.9	93.2	81.4	98.5
	Log-Rank	87.7	96.7	89.7	97.9	89.0	95.8	85.3	93.6	84.5	95.8
	DetectGPT	90.7	97.5	88.1	96.1	91.3	97.2	89.9	96.7	89.8	97.1
	DNA-GPT	89.8	97.5	90.7	97.6	90.2	98.8	88.0	95.0	87.5	94.4
Model-based	RoBERTa	98.7	99.9	91.2	97.5	73.3	83.2	78.0	90.9	60.5 (−12.8)	75.1 (−8.1)
	RMLM	98.8	**100**	91.7	98.2	75.0	88.1	81.7	94.2	64.0 (−11.0)	80.9 (−7.2)
	RADAR	98.9	**100**	92.0	99.0	75.8	93.1	97.7	99.7	68.6 (−7.2)	87.0 (−6.1)
	SCRN	99.2	99.9	92.4	97.2	77.1	93.3	97.6	**99.8**	66.2 (−10.9)	89.9 (−3.4)
	ConDA	99.1	99.9	95.6	99.1	95.0	99.2	85.8	96.7	80.6 (−5.2)	95.1 (−1.6)
	OURS	**99.4**	**100**	**98.4**	**99.9**	**98.2**	**99.9**	**97.8**	99.6	**95.4** (−2.4)	**99.5** (−0.1)

utilizes RoBERTa [31] as the encoder, with an embedding dimension $d = 768$. Following the same setting in [18], the hyperparameters in the loss function, namely λ_{RC}, λ_{CLS}, and λ_{DA}, are set to 0.5, 0.01, and 0.5, respectively. The retention rate γ is set to 0.8, the number of multi-head attention heads m is 8, the random seed is 42, and the hidden layer dimensions of the re-encoder and re-decoder r are both set to 2048. The model is optimized using Adam [32] with a batch size of 16, a learning rate of 2×10^{-5}, and is trained for 5 epochs. The entire experiment is deployed on two NVIDIA A100-80G GPUs.

For evaluation, we employ the F1 score and AUC as metrics for classification performance. For comparative evaluation, we benchmark our model against four metric-based methods: GLTR [10], Log-Rank [14], DetectGPT [14], DNA-GPT [13], and five model-based methods: RoBERTa [31], RMLM [20], RADAR [33], SCRN [18], and ConDA [21]. The experimental configurations follow their respective default settings.

B. Detection Results across Different Scenarios

We assess a variety of AGTD methods across five different scenarios, including **Base**, which represents the ideal case where the text generators and text domains in the source and target domains are identical; **Cross-G**, where the AI text generators differ between the source and target domains, with the target generator aligned to that of the actual target domain dataset; **Cross-D**, where the text domain in the target domain differs from that in the source domain; **Attack**, where the text in the target domain has been adversarially perturbed via paraphrase attacks; and **CDA**, which refers to the complex domain adaptation introduced in our work.

As shown in Tables I and II, all detectors perform best in the **Base** scenario. ComDet achieves F1/AUC scores of 0.997/0.999 (NW→BS) and 0.994/1.000 (NW→PT). In the **Cross-G** scenario, model-based baselines see F1 drops of 3.5%–7.6% and AUC decreases of up to 2.7%, while ComDet consistently outperforms them. Under the **Cross-D** scenario, performance drops are larger (F1 down by up to 25.7%, AUC by up to 16.7%), but ComDet still improves over the best baselines by 2.9%–3.4% in F1 and 0.7%–1.3% in AUC. In the **Attack** scenario, RMLM and ConDA degrade significantly due to paraphrase attacks, while ComDet maintains strong robustness with F1/AUC around 0.98/0.99, comparable to the best baselines.

The **CDA** scenario considers both the differences in generators between the source and target domains, as well as variations in text domains while accounting for adversarial attack scenarios. Experimental results demonstrate that, under this scenario, the decline in classification accuracy for the detector is more pronounced compared to other scenarios. Specifically, in the NW→BS test set, model-based baseline methods exhibit a significant drop, with F1 decreasing by 18.4% to 59.9% and AUC decreasing by 8.2% to 21.7%. Similarly, in the NW→PT test set, F1 decreases by 18.7%

TABLE III
ABLATION EXPERIMENT OF THE COMDET FRAMEWORK UNDER CDA SCENARIO. "-FAM", "-RM", AND "-FEM" INDICATE THE REMOVAL OF THE FEATURE ALIGNMENT MODULE, RECONSTRUCTION MODULE, AND FEATURE ENHANCEMENT MODULE, RESPECTIVELY.

Datasets	Method	w/o. attack		w. attack	
		F1	AUC	F1	AUC
NW→BS	Ours	96.8	99.7	96.2	99.3
	-FAM	72.8	91.3	69.0	88.2
	-RM	96.5	99.3	83.4	94.8
	-FEM	93.5	98.1	93.3	98.2
NW→PT	Ours	96.0	99.1	95.4	99.5
	-FAM	70.6	92.6	66.7	86.2
	-RM	95.8	99.1	79.4	91.0
	-FEM	92.9	97.5	92.6	97.4

TABLE IV
THE STATISTICAL RESULTS OF THE FEATURE DISTRIBUTIONS OF ROBERTA AND COMDET ON THE TEST SETS. THE SYMBOLS ↑ AND ↓ INDICATE THE DIRECTION OF PREFERENCE (HIGHER AND LOWER).

Datasets	Metric	RoBERTa	Ours
NW→BS	Silhouette (↑)	0.066	0.257
	DBI (↓)	3.711	1.502
	CH-index (↑)	141.1	860.9
	MMD (↓)	0.228	0.002
NW→PT	Silhouette (↑)	0.046	0.294
	DBI (↓)	4.676	1.382
	CH-index (↑)	88.4	1019.8
	MMD (↓)	0.351	0.002

to 38.7%, and AUC decreases by 4.8% to 24.8%. These results suggest that the combined challenges of cross-domain, cross-generator, and adversarial attack significantly amplify the complexity of the AGTD task, rendering it more difficult than any individual scenario in isolation.

In contrast, our proposed method outperforms the optimal baseline, yielding F1 improvements of 14.8% and 18.4% and AUC improvements of 7.5% and 4.6%, respectively. Moreover, our approach closely approximates the performance of the optimal fully supervised algorithm. In the NW→BS test set, F1 reaches 0.962, and AUC reaches 0.993, with only a slight drop of 3.5% in F1 and 0.6% in AUC compared to the Base scenario. Similarly, in the NW→PT test set, F1 is 0.954, and AUC is 0.995, with a minimal drop of 4.0% in F1 and 0.5% in AUC compared to the Base scenario. These results validate the effectiveness of our proposed method in the CDA scenario.

In conclusion, ComDet significantly outperforms all baseline methods in the CDA scenario while achieving comparable or superior performance in other real-world settings.

C. Ablation Study

To validate the effectiveness of the individual modules in the ComDet framework, we conduct ablation experiments on both test sets, and experimental results are shown in Table III.

When unlabeled data from the target domain is excluded from the training process, the detection performance of ComDet significantly deteriorates. On the NW→BS test set, the F1 score decreases by 28.3%, and the AUC drops by 11.2%. On the NW→PT test set, F1 decreases by 30.0%, and AUC drops by 13.4%. The absence of the reconstruction module decreases the model's robustness against adversarial attacks. In the standard CDA scenario, compared to the original ComDet, F1 drops by 15.3% and AUC drops by 4.5% on the NW→BS test set. On the NW→PT test set, F1 drops by 16.8%, and AUC drops by 8.5%. Finally, removing the feature extraction module weakens the framework's feature extraction capability. On the NW→BS test set, F1 decreases by 3.0%, and AUC decreases by 1.1%. On the NW→PT test set, F1 decreases by 2.9%, and AUC decreases by 2.1%.

D. Feature Distribution Analysis

We extract features for the text to be detected from the final hidden layer of both RoBERTa and ComDet, and employ t-SNE [34] to project the feature distribution of the test set into a 2D space for visualization, as illustrated in Fig. 3. To evaluate the transfer effect, we introduce two categories of metrics. The first category consists of clustering analysis metrics, including the Silhouette coefficient [35], DBI [36], and CH-index [37], which assess the clustering quality of features with the same detection label (either "human" or "AI"). The second category includes divergence analysis metric, MMD [38], which measures the consistency of feature distributions across different domain labels (either "source" or "target").

The experimental results shown in Table IV indicate that our method significantly improves the Silhouette coefficient and the CH-index, while also considerably reducing the DBI. This demonstrates the strong clustering performance of the extracted text features. Furthermore, our method achieves a notably lower MMD compared to RoBERTa on both test sets, effectively minimizing the differences in feature distribution between domains.

These results demonstrate that our proposed framework not only enhances the discrimination between AI and human text features but also reduces distribution discrepancies between domains, resulting in high classification accuracy and effective transferability.

E. Hyperparameter Analysis

We further investigate the effect of the hyperparameter γ in the NW→BS setting, as shown in Fig. 4. The results indicate that a dropout rate of 0.2 ($\gamma = 0.8$) is a relatively reasonalble choice. Under this setting, ComDet achieves the highest F1 score, which we attribute to the balance it strikes between noise injection and the preservation of key features.

F. Number of Training Samples

To analyze ComDet's performance under different data sizes, we re-implement CDA experiments in NW→BS using subsets of 100, 200, and 400 labeled training samples (test set fixed at 1000 samples). As shown in Fig. 4, ComDet consistently outperforms the best baseline ConDA across all settings. With 100, 200, and 400 training samples, ComDet surpasses ConDA by 11.6%, 15.2%, and 14.2% in F1 score,

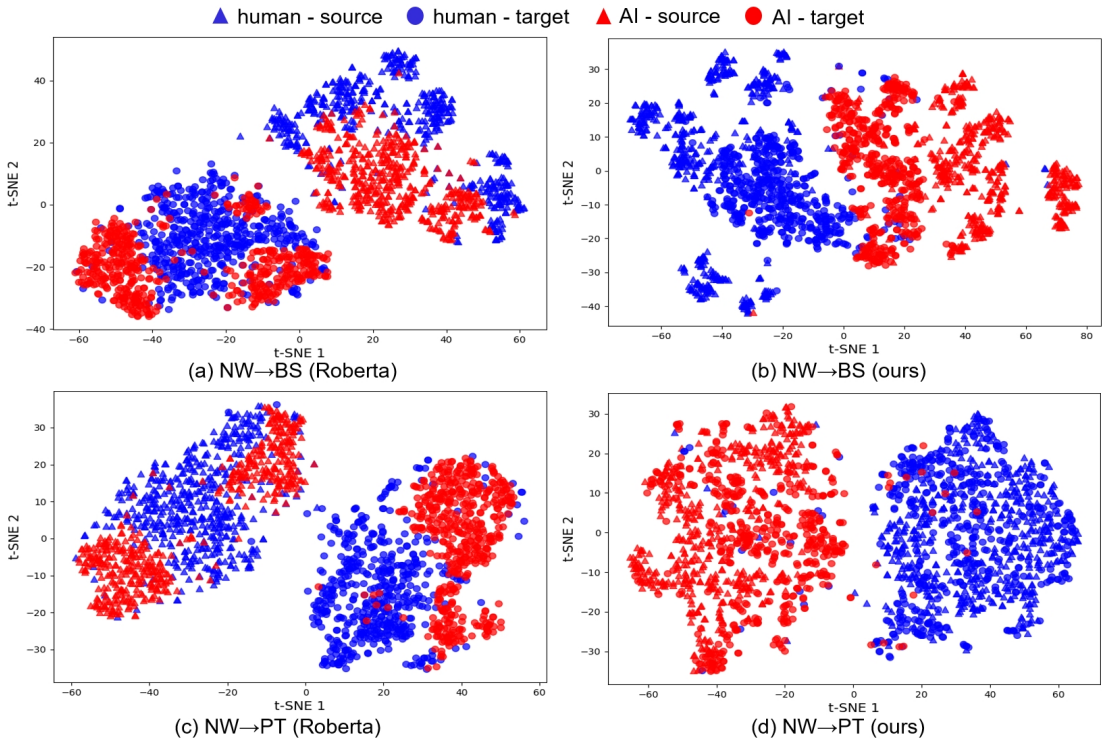

Fig. 3. t-SNE plots showing text representations from RoBERTa and our ComDet.

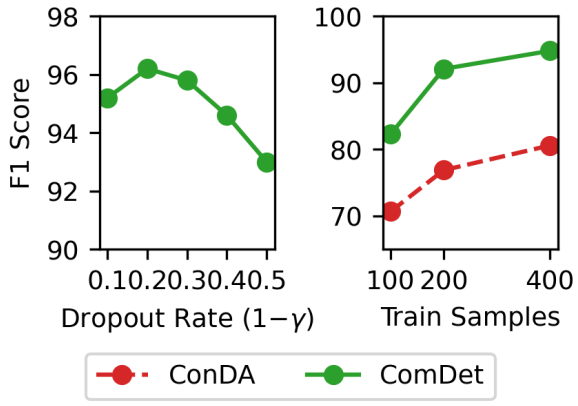

Fig. 4. The effect of the dropout rate $1-\gamma$ (0.1–0.5) and the number of training samples (100, 200, 400) on detection performance under the CDA scenario (NW→BS).

respectively. Moreover, the smaller performance gap between low- and high-resource settings suggests that our method exhibits lower dependence on labeled data.

V. Conclusion

Our work introduces a novel scenario for AI-generated text detection task called Complex Domain Adaptation, which combines three significant challenges: label scarcity, diverse distribution, and adversarial attacks into a unified task. It reflects a more complex situation in real-world AGTD applications. Through a series of experiments, we demonstrate that existing baseline methods struggle to perform effectively in the CDA scenario. In response, we propose an effective detection framework called ComDet, which significantly outperforms all baseline methods in the CDA scenario while achieving comparable or superior performance in other real-world settings. Furthermore, we validate the effectiveness of the ComDet framework through ablation experiments and feature distribution analysis.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 72210107001, 72225011, 72434005 and L242400108, the Chinese Academy of Science President's International Fellowship Initiative (CAS PIFI) International Outstanding Team Project under Grant 2024PG0013, the National Social Science Fund of China (Major Program, Grant No. 25&ZD161), and the CAS Pilot Project for Basic and Interdisciplinary Scientific Research (Grant No. XDB1380303).

References

[1] F. Wang, Z. Zhang, X. Zhang, Z. Wu, T. Mo, Q. Lu, W. Wang, R. Li, J. Xu, X. Tang *et al.*, "A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness," *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 6, pp. 1–87, 2025.

[2] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 5, pp. 1–72, 2025.

[3] J. Wu, S. Yang, R. Zhan, Y. Yuan, L. S. Chao, and D. F. Wong, "A survey on llm-generated text detection: Necessity, methods, and future directions," *Computational Linguistics*, vol. 51, no. 1, pp. 275–338, 2025.

[4] Y. Li, M. Milling, L. Specia, and B. W. Schuller, "Discourse features enhance detection of document-level machine-generated content," in *2025 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2025, pp. 1–8.

[5] J. Huang, C. Zhang, T. Liu, Q. Li, T. Su, C. Chen, and Y. Li, "Language models understand themselves better: A zero-shot ai-generated text detection method via reading and writing," in *2025 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2025, pp. 1–8.

[6] B. Luo, "Enhancing the undetectability of ai-generated text: A semantics-preserving evasion technique," in *2025 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2025, pp. 1–8.

[7] A. Bhattacharjee, T. Kumarage, R. Moraffah, and H. Liu, "Conda: Contrastive domain adaptation for ai-generated text detection," in *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 598–610.

[8] A. Bhattacharjee, R. Moraffah, J. Garland, and H. Liu, "Eagle: A domain generalization framework for ai-generated text detection," *arXiv preprint arXiv:2403.15690*, 2024.

[9] Y. Zhou and J. Wang, "Detecting ai-generated texts in cross-domains," in *Proceedings of the ACM Symposium on Document Engineering 2024*, 2024, pp. 1–4.

[10] S. Gehrmann, H. Strobelt, and A. Rush, "GLTR: Statistical detection and visualization of generated text," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, M. R. Costa-jussà and E. Alfonseca, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 111–116. [Online]. Available: https://aclanthology.org/P19-3019/

[11] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps *et al.*, "Release strategies and the social impacts of language models," *arXiv preprint arXiv:1908.09203*, 2019.

[12] B. Zhu, L. Yuan, G. Cui, Y. Chen, C. Fu, B. He, Y. Deng, Z. Liu, M. Sun, and M. Gu, "Beat llms at their own game: Zero-shot llm-generated text detection via querying chatgpt," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 7470–7483.

[13] X. Yang, W. Cheng, Y. Wu, L. R. Petzold, W. Y. Wang, and H. Chen, "Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text," in *ICLR*, 2024. [Online]. Available: https://openreview.net/forum?id=Xlayxj2fWp

[14] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, "Detectgpt: Zero-shot machine-generated text detection using probability curvature," in *International Conference on Machine Learning*. PMLR, 2023, pp. 24 950–24 962.

[15] R. A. R. Soto, K. Koch, A. Khan, B. Y. Chen, M. Bishop, and N. Andrews, "Few-shot detection of machine-generated text using style representations," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=cWiEN1plhJ

[16] Y. Shi, Q. Sheng, J. Cao, H. Mi, B. Hu, and D. Wang, "Ten words only still help: Improving black-box ai-generated text detection via proxy-guided efficient re-sampling," *arXiv preprint arXiv:2402.09199*, 2024.

[17] P. Wang, L. Li, K. Ren, B. Jiang, D. Zhang, and X. Qiu, "SeqXGPT: Sentence-level AI-generated text detection," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 1144–1156. [Online]. Available: https://aclanthology.org/2023.emnlp-main.73/

[18] G. Huang, Y. Zhang, Z. Li, Y. You, M. Wang, and Z. Yang, "Are AI-generated text detectors robust to adversarial perturbations?" in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 6005–6024. [Online]. Available: https://aclanthology.org/2024.acl-long.327/

[19] L. Li, D. Song, and X. Qiu, "Text adversarial purification as defense against adversarial attacks," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 338–350. [Online]. Available: https://aclanthology.org/2023.acl-long.20/

[20] Z. Wang, Z. Liu, X. Zheng, Q. Su, and J. Wang, "RMLM: A flexible defense framework for proactively mitigating word-level adversarial attacks," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 2757–2774. [Online]. Available: https://aclanthology.org/2023.acl-long.155/

[21] A. Bhattacharjee, T. Kumarage, R. Moraffah, and H. Liu, "ConDA: Contrastive domain adaptation for AI-generated text detection," in *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, J. C. Park, Y. Arase, B. Hu, W. Lu, D. Wijaya, A. Purwarianti, and A. A. Krisnadhi, Eds. Nusa Dua, Bali: Association for Computational Linguistics, Nov. 2023, pp. 598–610. [Online]. Available: https://aclanthology.org/2023.ijcnlp-main.40/

[22] G. Huang, Y. Zhang, Z. Li, Y. You, M. Wang, and Z. Yang, "Are ai-generated text detectors robust to adversarial perturbations?" in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 6005–6024.

[23] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[24] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[25] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[26] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[27] L. Dugan, A. Hwang, F. Trhlík, A. Zhu, J. M. Ludan, H. Xu, D. Ippolito, and C. Callison-Burch, "RAID: A shared benchmark for robust evaluation of machine-generated text detectors," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 12 463–12 492. [Online]. Available: https://aclanthology.org/2024.acl-long.674/

[28] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[29] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[30] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[31] Z. Liu, W. Lin, Y. Shi, and J. Zhao, "A robustly optimized bert pre-training approach with post-training," in *China National Conference on Chinese Computational Linguistics*. Springer, 2021, pp. 471–484.

[32] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] X. Hu, P.-Y. Chen, and T.-Y. Ho, "Radar: Robust ai-text detection via adversarial learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 15 077–15 095, 2023.

[34] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[35] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[36] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[37] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[38] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample-problem," *Advances in neural information processing systems*, vol. 19, 2006.