Breaking the Generator Barrier: Disentangled Representation for Generalizable AI-Text Detection

Anonymous ACL submission

Abstract

As large language models (LLMs) generate text that increasingly resembles human writing, the subtle cues that distinguish AI-generated content from human-written content become increasingly challenging to capture. Reliance on generator-specific artifacts is inherently unstable, since new models emerge rapidly and reduce the robustness of such shortcuts. This generalizes unseen generators as a central and challenging problem for AI-text detection. To tackle this challenge, we propose a progressively structured framework that disentangles AI-detection semantics from generator-aware artifacts. This is achieved through a compact latent encoding that encourages semantic minimality, followed by perturbation-based regularization to reduce residual entanglement, and finally a discriminative adaptation stage that aligns representations with task objectives. Experiments on MAGE benchmark, covering 20 representative LLMs across 7 categories, demonstrate consistent improvements over state-of-the-art methods, achieving up to 24.2% accuracy gain and 26.2% F_1 improvement. Notably, performance continues to improve as the diversity of training generators increases, confirming strong scalability and generalization in open-set scenarios. Our source code will be publicly available at http://github.com/***.

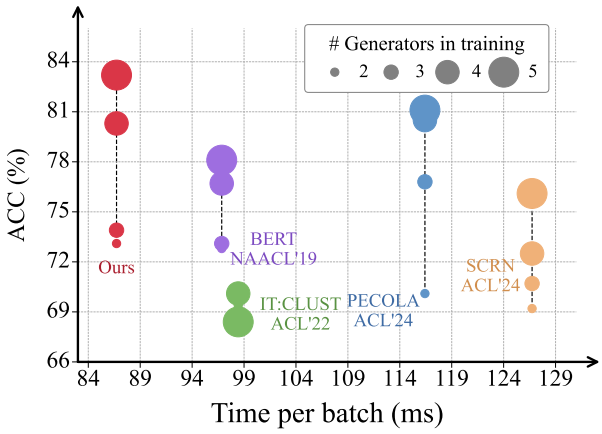


Figure 1: Comparison with competitive approaches. On the unseen OPT generators, our method achieves the best balance between accuracy and efficiency, and its advantage grows with training generator diversity.

1 Introduction

The rapid proliferation of large language models (LLMs) has resulted in a surge of AI-generated text (AIGT) across news media, social platforms, and academic domains. While this development offers new opportunities, it has also raised increasing concerns regarding misinformation (Wei et al., 2025; Hu et al., 2024), manipulation (Guan et al., 2024; Shao et al., 2023), and authorship integrity (Silva et al., 2024; Vasilatos et al., 2023). These concerns have made detecting AI-generated text a pressing task in trustworthy AI and content regulation, especially amid rapidly emerging unseen LLMs.

Although recent advances in AIGT detection have shown promising results under closed-world settings where training and test samples share the same generator (Wang et al., 2025; Liu et al., 2024b), these methods fail to generalize to previously unseen generators, resulting in substantial performance degradation. This issue is especially pronounced in real-world settings, where novel LLMs are continuously emerging. While prior efforts have explored cross-domain generalization through adversarial (Huang et al., 2024) or contrastive learning (Liu et al., 2024b), they tend to treat generator shift as a subset of domain shift and often overlook its distinct challenges. Recent studies (Wu et al., 2025; Guo et al., 2024; Li et al., 2024; Wang et al., 2024) have shown that shifts across generators introduce unique semantic, syntactic, and stylistic variations, resulting in generalization gaps beyond the scope of domain-centric solutions.

To address the core challenge of generalization

to unseen LLM-based generators, we propose a progressively structured framework that disentangles AI-detection semantics from generator-aware artifacts. The goal is to isolate task-relevant signals by suppressing entangled generator bias. The process begins with a dual-bottleneck design that enforces compactness and minimality in the latent space, encouraging essential task-focused representations. Building on this, cross-view regularization disrupts residual correlations and promotes independence among factors. Finally, a discriminator-guided adaptation stage further consolidates the separation by refining each representation stream against task-specific objectives and suppressing residual leakage across branches. This progressive pipeline incrementally purifies AI-detection semantics while filtering out generator-aware noise, yielding representations that generalize more effectively across diverse LLMs.

In summary, our contributions are as follows:

- We address the challenge of AIGT detection on unseen LLMs by learning task-focused representations that isolate AI-detection cues from generator-aware artifacts, enabling stronger generalization in open-set scenarios.
- Our solution follows a progressive design: it begins with dual-bottleneck encoding to encourage semantic compactness, applies cross-view regularization to disrupt entangled factors, and finalizes with discriminative-guided adaptation that sharpens task alignment and reinforces representational separation.
- Experiments on 20 representative LLMs demonstrate consistent improvements over state-of-the-art methods, with increasingly larger gains under diverse generator settings, validating the effectiveness and generalizability of our approach (see Figure 1).

2 Related Work

2.1 AI-generated Text Detection

With the widespread adoption of large language models (LLMs) such as GPT-3 (Brown et al., 2020), GPT-4 (Achiam et al., 2023), and Claude, detecting AIGT has become a central task in trustworthy NLP. Existing detection methods can be categorized into four main groups.

Statistic-based approaches (Mitchell et al., 2023; Su et al., 2023; Gehrmann et al., 2019) rely on entropy, perplexity, or curvature-based signals to distinguish human-written text from machine-generated outputs. Watermark-based methods (Liu et al., 2024a; Kirchenbauer et al., 2023) inject identifiable patterns into the generation process to facilitate detection. Classifier-based techniques (Liu et al., 2024b; Hu et al., 2023; Shnarch et al., 2022; Uchendu et al., 2020) typically fine-tune pre-trained models such as RoBERTa or BERT, often incorporating adversarial or contrastive objectives. Retrieval-based approaches (Krishna et al., 2023) match text against known generator outputs via semantic similarity.

Although many existing approaches achieve strong performance in closed-world settings, where training and test data originate from the same generator, they often fail to generalize to previously unseen generators. Recent benchmarks such as MAGE (Li et al., 2024) and M4 (Wang et al., 2024) have systematically exposed such weaknesses, revealing substantial performance drops under generator shifts. These findings underscore the need for open-set detectors that explicitly model generator-aware variations.

2.2 Beyond Domain Generalization: Generator-Aware Challenges

To improve robustness under distribution shifts, several studies have explored domain generalization (DG) techniques, which were initially developed for cross-topic or cross-genre transfer in NLP and vision. These methods aim to learn domain-invariant features through adversarial training (Ganin et al., 2016), feature alignment (Li et al., 2018), or data augmentation (Li et al., 2022; Peng et al., 2018). Motivated by their effectiveness, DG strategies have been adapted for AI text detection (Liu et al., 2024b; Huang et al., 2024), showing modest gains on certain unseen generators.

However, most DG-based approaches implicitly assume that generator shift resembles domain shift. Studies such as MAGE (Li et al., 2024) reveal that generator-aware variation introduces unique stylistic and semantic artifacts that differ qualitatively from conventional domain differences. Additional findings (Chen et al., 2025; Wang et al., 2024; Jiang et al., 2023) show that generators vary not only in syntax or topical preference, but also in prompt interpretation and semantics, factors that standard domain-invariant methods fail to disentangle.

These limitations highlight the need for approaches that transcend conventional domain gen-

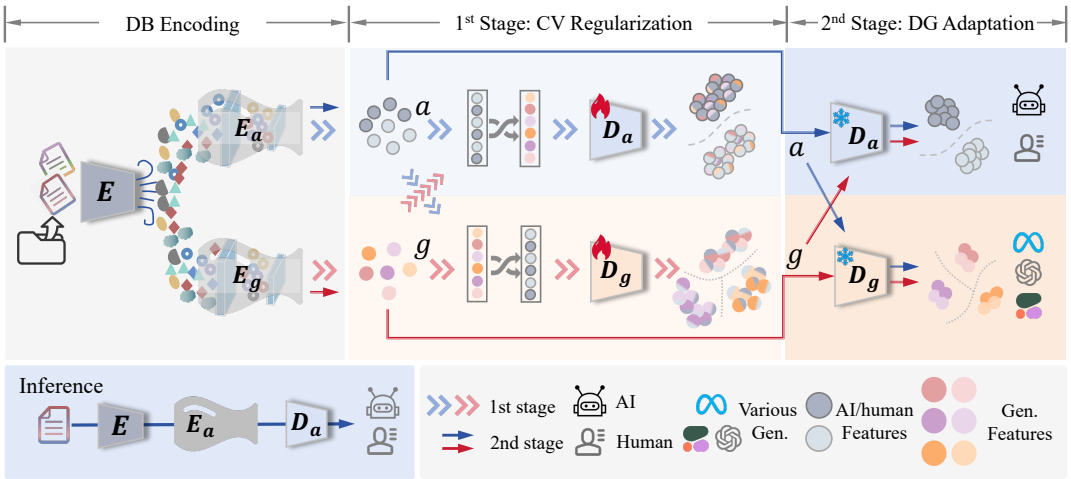


Figure 2: Overview of the proposed framework. The model enhances generalization to unseen generators by enforcing semantic disentanglement between AI-detection and generator-aware features. It integrates compact dual-bottleneck (DB) encoding, cross-view (CV) regularization, and discriminator-guided (DG) adaptation into a cohesive disentanglement-enhancing pipeline.

eralization by explicitly modeling generator-aware semantics. Unlike prior one-shot learning methods that implicitly subsume generator variation under general domain shifts, our approach progressively disentangles detection-relevant semantics from generator-aware artifacts, enabling robust and interpretable generalization to unseen LLMs.

3 Methodology

We propose a progressively structured framework for detecting AIGT from unseen generators by learning task-specific representations that disentangle AI-detection semantics from generator-aware patterns. The framework begins with a dual-bottleneck encoding that encourages compact and factorized representations by minimizing shared redundancy. A cross-view regularization then disrupts residual entanglement across branches, enhancing representational independence. Finally, discriminative adaptation refines each branch using dedicated pre-trained discriminators, aligning each with its target objective while suppressing interference from the opposite one. Together, these components form a cohesive pipeline that enhances semantic purity and significantly improves generalization across diverse generative sources. The overall architecture is shown in Figure 2.

3.1 Problem Setup

We address the task of detecting whether a given text sample is AI-generated, even when the generator that produced it has not been observed during training. Each training instance is denoted as $\mathcal{X} = \{(x_i, y_i, s_i)\}_{i=1}^{N}$, where x_i is a text sample, $y_i \in \{0, 1\}$ is the source label (human or AI) and s_i denotes the generator identity (e.g., GPT, LLaMA). This formulation enables the model to learn both AI-detection cues and generator-aware variations that are useful for open-set generalization.

3.2 Dual-Bottleneck Encoding

To improve generalization under distribution shifts from unseen LLM-based generators, we propose a dual-bottleneck encoding module that disentangles AI-detection semantics from generator-aware artifacts. This is achieved by enforcing compact, task-aligned latent representations guided by the information bottleneck (IB) principle.

The IB objective encourages latent features z to retain information about the target label y while discarding irrelevant input variations: $\min\big[-I(z;y) + \beta I(x;z)\big]$, where β balances prediction fidelity and compression. In practice, this translates into a prediction loss (approximating $-I(z;y)$) combined with a KL regularization term (approximating $I(x;z)$).

Concretely, we implement two parallel encoder branches: E_a for AI-detection semantics and E_g for generator-aware cues. Given an input x_i, we first obtain a contextual embedding $\mathbf{h}_i$ from a pre-trained BERT [CLS] token. This is projected into intermediate vectors $\mathbf{e}_i^{(a)}$ and $\mathbf{e}_i^{(g)}$ via MLPs, which parameterize diagonal-Gaussian posteriors:

$$q(\mathbf{a}_i \mid \mathbf{e}_i^{(a)}) = \mathcal{N}\Big(\boldsymbol{\mu}_i^{(a)}, \mathrm{diag}\big((\boldsymbol{\sigma}_i^{(a)})^2\big)\Big), \quad (1)$$

where $\boldsymbol{\mu}_i^{(a)}$ and $\boldsymbol{\sigma}_i^{(a)}$ are computed via separate linear layers, with softplus activation applied to the variance to ensure positivity. An analogous operation is applied to obtain the generator-aware posterior $q(\mathbf{g}_i \mid \mathbf{e}_i^{(g)})$.

To enable differentiable sampling during training, we draw K samples using the reparameterization trick (Kingma and Welling, 2013):

$$\mathbf{a}_i^{(K)} = \boldsymbol{\mu}_i^{(a)} + \boldsymbol{\sigma}_i^{(a)} \odot \boldsymbol{\epsilon}^{(K)}, \quad \boldsymbol{\epsilon}^{(K)} \sim \mathcal{N}(0, \mathbf{I}), \tag{2}$$

where $\odot$ denotes element-wise multiplication. Then we average the prediction losses across K Monte Carlo samples for stability. At inference time, the latent feature is taken as the posterior mean: $\mathbf{a}_i = \boldsymbol{\mu}_i^{(a)}$. The same sampling and mean-inference strategy is applied to derive the latent representation $\mathbf{g}_i$.

To enforce compactness and disentanglement, we introduce learnable priors $p(\mathbf{a})$ and $p(\mathbf{g})$ for each branch, which is initialized as a standard Gaussian $\mathcal{N}(0, \mathbf{I})$ and optionally updated during training. We then apply KL regularization as follows:

$$\mathcal{L}_{\text{DB}} = D_{\text{KL}}\left(q(\mathbf{a}_i \mid \mathbf{e}_i^{(a)}) \,\|\, p(\mathbf{a})\right) + D_{\text{KL}}\left(q(\mathbf{g}_i \mid \mathbf{e}_i^{(g)}) \,\|\, p(\mathbf{g})\right). \tag{3}$$

This regularization encourages latent representations $\mathbf{a}_i$ and $\mathbf{g}_i$ to align with task-agnostic priors, minimizing redundancy while promoting task-specific compression.

This dual-bottleneck design enforces semantic compression and separation at the feature level, providing a solid foundation for subsequent regularization and adaptation modules to operate on disentangled, robust representations.

3.3 Cross-View Regularization

While the dual-bottleneck encoding encourages semantic decoupling, residual entanglement may still persist due to implicit correlations and shared linguistic patterns across generators. To further separate AI-detection and generator-aware semantics, we introduce a cross-view regularization mechanism that explicitly perturbs each representation with signals from its complementary branch, thereby promoting representational independence.

For each input x_i, we sample another instance $x_j \neq x_i$ (x_j is a human or AI sample) within the batch and perturb the AI-detection feature $\mathbf{a}_i$ using the generator-aware representation $\mathbf{g}_j$. The perturbed output $\tilde{\mathbf{a}}_i$ is computed as:

$$\tilde{\mathbf{a}}_i = \gamma \cdot \mathbf{a}_i + (1 - \gamma) \cdot \phi(\mathbf{g}_j, \mathbf{a}_i), \tag{4}$$

where $\gamma \sim \mathcal{U}(0.5, 1)$ controls the interpolation, and $\phi(\cdot, \cdot)$ transfers the statistics style of $\mathbf{g}_i$ to $\mathbf{a}_j$:

$$\phi(\mathbf{g}_i, \mathbf{a}_j) = \frac{\mathbf{a}_j - \mu(\mathbf{a}_j)}{\sigma(\mathbf{a}_j)} \cdot \sigma(\mathbf{g}_i) + \mu(\mathbf{g}_i), \tag{5}$$

with $\mu(\cdot)$ and $\sigma(\cdot)$ denoting mean and standard deviation over features. A symmetric operation is applied to perturb $\tilde{\mathbf{g}}_i$ using $\mathbf{a}_j$

To further mitigate asymmetry, where AI-generated samples are more susceptible to misclassifications, we apply additional cross-branch perturbations exclusively to AI-generated samples, producing augmented variant $\tilde{\mathbf{a}}_i^{(\text{aug})}$. We then apply a dual-branch prediction loss with additional regularization terms on the augmented samples:

$$\mathcal{L}_{\text{reg}} = -\mathbb{E}\left[\log D_a^{(y_i)}(\tilde{\mathbf{a}}_i) + \log D_g^{(s_i)}(\tilde{\mathbf{g}}_i)\right] - \frac{1}{|\mathcal{B}_{\text{AI}}^{(\text{aug})}|} \sum_{i \in \mathcal{B}_{\text{AI}}^{(\text{aug})}} \log D_a^{(y_i)}(\tilde{\mathbf{a}}_i^{(\text{aug})}), \tag{6}$$

where $D^{(k)}(\cdot)$ denotes the predicted probability of class k and $\mathcal{B}_{\text{AI}}^{(\text{aug})}$ is the set of augmented samples.

By disrupting residual cross-branch dependencies and injecting controlled semantic noise, the perturbation-enhanced regularization encourages structurally independent representations, ultimately enhancing robustness to unseen generator shifts.

3.4 Discriminative-Guided Adaptation

Despite the earlier regularization, residual overlaps between branches may persist due to shared cues. To further enforce disentanglement, we introduce a discriminative-guided adaptation stage.

Specifically, we freeze the parameters of the AI-detection discriminator D_a and the generator-aware discriminator D_g, both pre-trained on perturbed representations to encode robust task semantics. For clean representations, each branch encoder is updated under two complementary constraints: (i) when passed into its own discriminator, the representation should be correctly classified, ensuring task alignment; (ii) when passed through a gradient reversal layer (GRL) into the opposite discriminator, the representation is encouraged to be misclassified, forcing the encoder to remove any information exploitable by the other branch. Formally, the

Categories	MAGE Corpus (Li et al., 2024)						
	FLAN-T5	GPT	LLaMA	OPT	GLM	BigScience	EleutherAI
LLMs included	small/base/large xl/xxl	davinci-002/003 gpt-turbo-3.5	6B/13B 30B/65B	2.7B/6.7B iml-1.3B/30B	130B	Bloom-7B	GPT-J-6B GPT-NeoX-20B
AI	9,000	13,800	9,000	9,000	9,230	8,925	9,000
Human	9,000	13,800	9,000	9,000	9,230	8,925	9,000

Table 1: Number of AI-generated and human-written samples across datasets. Balanced sampling is applied to categories with multiple LLMs so that each category contains a comparable number of samples, resulting in 20 representative models overall.

Algorithm 1: Structured Disentanglement Framework

Input: encoder E, bottleneck encoders E_a, E_g, discriminators D_a, D_g, weight factor β, training set $\mathcal{X}$
Output: optimized encoders and discriminators

for each training epoch **do**
  for each mini-batch $\mathcal{B} = \{x_i, y_i, s_i\}_{i=1}^{N}$ **do**
    Stage I: Encoding and Regularization
      $\mathbf{h}_i = E(x_i)$
      $\mathbf{e}_i^{(a)} \leftarrow \text{MLP}^{(a)}(\mathbf{h}_i), \quad \mathbf{e}_i^{(g)} \leftarrow \text{MLP}^{(g)}(\mathbf{h}_i)$
      $q(\mathbf{a}_i|\mathbf{e}_i^{(a)}), q(\mathbf{g}_i|\mathbf{e}_i^{(g)}) \leftarrow$ Compute posteriors
      $p(\mathbf{a}), p(\mathbf{g}) \leftarrow$ Learnable priors
      Compute DB loss $\mathcal{L}_{\text{DB}}$ via Eq. 3
      $\tilde{\mathbf{a}}_i, \tilde{\mathbf{g}}_i, \tilde{\mathbf{a}}_i^{(\text{aug})} \leftarrow$ Apply regularization
      Compute discriminative loss $\mathcal{L}_{\text{reg}}$ via Eq. 6
      Compute 1st-stage loss: $\mathcal{L}_{\text{stage1}} = \beta\mathcal{L}_{\text{DB}} + \mathcal{L}_{\text{reg}}$
      Update E, E_a, E_g, D_a, D_g w.r.t. $\mathcal{L}_{\text{stage1}}$
    Stage II: Discriminative-Guided Adaptation
      Freeze D_a, D_g
      $\mathbf{a}_i, \mathbf{g}_i \leftarrow$ pure features
      Compute 2nd-stage loss $\mathcal{L}_{\text{adapt}}$ via Eq. 7
      Update E_a, E_g w.r.t. $\mathcal{L}_{\text{adapt}}$

adaptation loss integrates both objectives:

$$\mathcal{L}_{\text{adapt}} = -\mathbb{E}\Big[\log D_a^{(y_i)}(\mathbf{a}_i) + \log D_a^{(y_i)}\big(\mathcal{G}(\mathbf{g}_i)\big) + \log D_g^{(s_i)}(\mathbf{g}_i) + \log D_g^{(s_i)}\big(\mathcal{G}(\mathbf{a}_i)\big)\Big], \quad (7)$$

where $\mathcal{G}(\cdot)$ denotes the GRL operation. This setup ensures that each encoder not only fits its own semantics but also actively eliminates cross-branch cues. By refining encoders under frozen discriminators with clean inputs, this stage produces more invariant, disentangled, and generalizable representations. The complete procedure, including loss definitions and optimization steps, is summarized in Algorithm 1.

4 Experiments

4.1 Experimental Details

Datasets. We adopt MAGE benchmark (Li et al., 2024) to comprehensively evaluate cross-generator generalization in AIGT detection. The original dataset consists of 27 LLMs grouped into 7 categories. To prioritize generalization over redundancy, we remove highly similar variants of the same categories, thereby obtaining a subset of 20 representative models. For category with a single model, the entire set is retained, whereas for those with multiple sub-models, we apply balanced random sampling to ensure comparable sizes. As summarized in Table 1, the final dataset spans diverse generator families (e.g., GPT, LLaMA, FLAN-T5), thereby providing a broad and challenging testbed for cross-generator evaluation.

Competitive Methods. To evaluate generalization on unseen generators, we compare with representative SOTA methods in two categories: training-based and zero-shot methods. Training-based methods include perturbation-based robustness models PECOLA (Liu et al., 2024b), SCRN (Huang et al., 2024), which add input noise or regularization during training to improve stability, and adaptation-based model IT:CLUST (Shnarch et al., 2022), which exploit clustering for intermediate supervision and domain adaptation. Zero-shot methods include Fast-DetectGPT (Bao et al., 2024), GLTR (Gehrmann et al., 2019), relying solely on pretrained LM statistics without fine-tuning. We also report BERT (Devlin et al., 2019) as an encoder-only baseline. These methods offer diverse strategies for evaluating the robustness and generalization of our method.

Implement Details. We use the Adam optimizer with a learning rate of 2×10^{-5} and a batch size of 16 across all experiments. During the DB-based encoding stage, each latent variable is sampled 5 times per instance to estimate expectations. Both

LLM Categories	Accuracy (%) / F_1-Measure						
	FLAN-T5	GPT	LLaMA	OPT	GLM	BigScience	EleutherAI
Zero-shot methods							
GLTR$_{\text{ACL'19}}$	56.9 / 52.4	71.8 / 71.5	75.1 / 75.8	75.1 / 77.3	77.8 / 78.2	86.6 / 86.4	80.2 / 80.0
Fast-DetectGPT$_{\text{ICLR'24}}$	57.3 / 74.9	70.0 / 82.5	73.4 / 79.8	74.6 / 76.1	69.9 / 72.8	79.3 / 84.2	77.0 / 82.9
Training-based methods							
BERT$_{\text{NAACL'19}}$	60.9 / 52.6	78.6 / 77.5	83.1 / 85.4	82.1 / 84.2	92.4 / 92.8	96.4 / 95.7	96.7 / 95.9
IT:CLUST$_{\text{ACL'22}}$	58.7 / 57.5	63.1 / 61.0	71.3 / 70.4	72.1 / 74.7	75.7 / 77.1	85.4 / 87.9	82.5 / 81.8
PECOLA$_{\text{ACL'24}}$	65.1 / 65.5	81.1 / 80.3	84.6 / 84.5	83.5 / 83.4	78.3 / 77.8	87.6 / 86.7	83.9 / 84.5
SCRN$_{\text{ACL'24}}$	59.6 / 58.7	69.0 / 67.8	75.0 / 73.3	81.3 / 79.2	71.8 / 67.7	87.8 / 87.1	90.6 / 90.0
Ours	**68.9 / 64.1**	**82.7 / 82.6**	**88.0 / 88.5**	**89.1 / 88.3**	**94.1 / 93.9**	**96.7 / 96.4**	**97.2 / 97.8**

Table 2: Cross-generator generalization results on seven LLM categories from the MAGE benchmark under the leave-one-generator-out protocol. Zero-shot and training-based methods are compared against our approach, which yields consistent gains across all held-out generators.

the AIGT and generator discrimination heads employ a dropout rate of 0.5 to mitigate overfitting. The loss balancing coefficient β, which controls the trade-off between the DB regularization and the discriminative objective in both stages, is set to 5×10^{-6}. All experiments are conducted on two NVIDIA GeForce RTX 3090 GPUs.

4.2 Overall Performance

4.2.1 Main Results on Cross-Generator Generalization

We evaluate our method under the leave-one-generator-out setting, training on 6 generator categories and testing on the held-out one. As shown in Table 2, accuracies vary widely from 60% to 98%, reflecting substantial variation in cross-generator difficulty. Across this range, our model consistently outperforms all competitive methods, achieving up to 24.2% accuracy improvement over Fast-DetectGPT on unseen GLM set. The gains are particularly notable under large distributional shifts. For instance, on the challenging unseen FLAN-T5 our approach improves accuracy by 8.0% over BERT, while on the high-performing EleutherAI, it yields 0.5%. These results highlight that our disentanglement design enables robust and scalable cross-generator generalization, particularly under large distributional shifts between train/test generators.

4.2.2 Effect of Training Generator Diversity

To examine the effect of training diversity on cross-generator generalization, we hold out OPT and FLAN-T5 as test categories while training on subsets of the remaining 5 families. For each setting, we fix the training size at 12,000 instances and vary the number of training generators $N \in \{2, 3, 4, 5\}$, distributing samples evenly across the selected categories to control for data volume. As shown in Table 3, our method consistently benefits from increased diversity and demonstrates stronger scalability than recent alternatives. In the unseen OPT detection scenario, under low-diversity training ($N = 2$), the margin over the strongest method is minimal (0.3% over BERT), but under high-diversity training ($N = 5$) it expands substantially (5.1% over BERT). These findings confirm that training diversity, rather than training volume, is the key driver for learning invariant features and achieving robust cross-generator generalization.

Model	Accuracy (%) / F_1-Measure			
	$N = 2$	$N = 3$	$N = 4$	$N = 5$
OPT Test				
BERT$_{\text{NAACL'19}}$	72.8/72.2	73.1/69.9	76.7/74.9	78.1/76.1
IT:CLUST$_{\text{ACL'22}}$	69.4/67.7	68.4/69.8	70.1/68.1	68.4/68.1
PECOLA$_{\text{ACL'24}}$	70.1/68.1	**76.8/75.5**	80.5/80.0	81.1/80.9
SCRN$_{\text{ACL'24}}$	69.2/68.3	70.7/67.4	72.5/69.9	76.1/75.9
Ours	**73.1/74.7**	73.9/75.1	**80.5/80.4**	**83.2/82.3**
FLAN-T5 Test				
BERT$_{\text{NAACL'19}}$	51.2/50.2	52.9/51.1	55.0/53.9	58.1/57.8
IT:CLUST$_{\text{ACL'22}}$	53.1/53.7	54.4/53.1	54.2/53.0	53.6/52.2
PECOLA$_{\text{ACL'24}}$	**56.7/56.3**	54.2/50.2	57.4/57.1	61.3/60.1
SCRN$_{\text{ACL'24}}$	53.6/52.1	54.1/53.2	56.8/55.9	57.8/57.4
Ours	54.7/52.6	**57.0/55.1**	**61.1/58.4**	**64.5/62.4**

Table 3: Cross-generator generalization on OPT and FLAN-T5. Models are trained with N categories under a fixed training size of 12,000 samples and evaluated on an unseen LLM category.

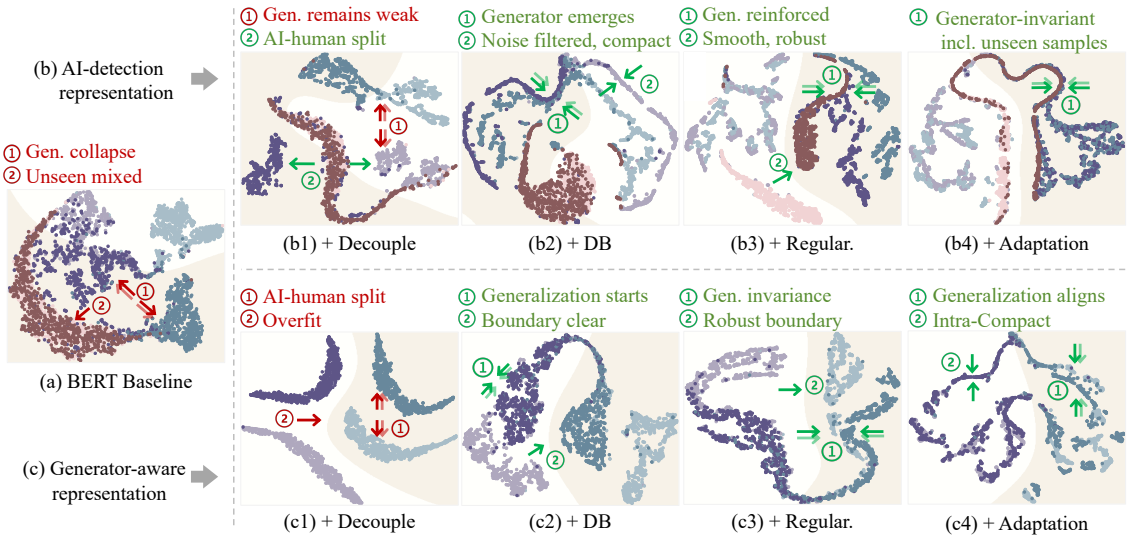


Figure 3: Progressive t-SNE visualization of disentangled feature spaces. The BERT baseline (a) collapses, with unseen samples (red) mixed into training clusters (blue/purple). After decoupling, the AI-detection (b) and generator-aware (c) branches begin to separate (b1/c1), but generalization remains unstable and generator-sensitive. With DB encoding (b2/c2), semantic noise is reduced, yielding more compact clusters. Cross-view regularization (b3/c3) smooths boundaries and enhances robustness. Finally, discriminator-guided adaptation (b4/c4) yields clear human–AI separation in the detection branch and generator-invariant alignment in the auxiliary branch.

4.3 Generalization via Progressive Cumulative Design

We visualize the evolution of latent features as modules are progressively added. Models are trained on GPT and BigScience categories and evaluated on the unseen FLAN-T5 one, with t-SNE projections shown in Figure 3. The BERT encoder produces entangled and poorly aligned representations, while each additional module: decoupling, DB encoding, cross-view regularization, and adaptation incrementally enhances compactness, robustness, and ultimately cross-generator generalization.

4.4 Analysis of Core Modules

To gain deeper insights into generalization, we study how individual modules influence the representation space along three dimensions: disentanglement, robustness, and compactness.

4.4.1 Filtering Generator Bias via Bottlenecked Encoding

To assess the contribution of the DB encoding module, we compare the AI-detection representations $\mathbf{a}$ learned by a conventional MLP encoder and our DB encoder, both trained on GPT, BigScience, and FLAN-T5 sets. As shown in Figure 4, both models successfully separate human from AI texts, but the MLP encoder exhibits clear generator bias with divided clusters. In contrast, our DB encoder mitigates such bias, yielding compact and generator-invariant features that support invariant feature learning and substantially enhance cross-generator generalization.

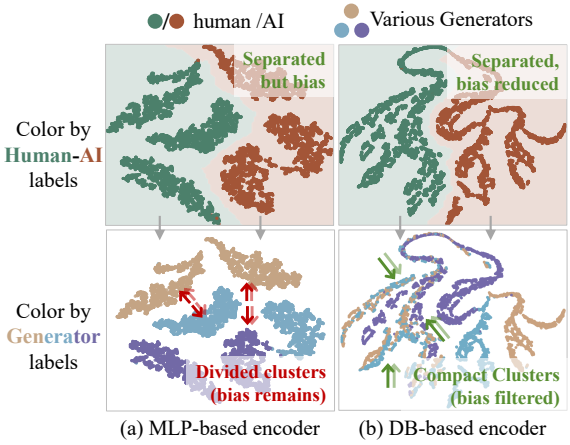
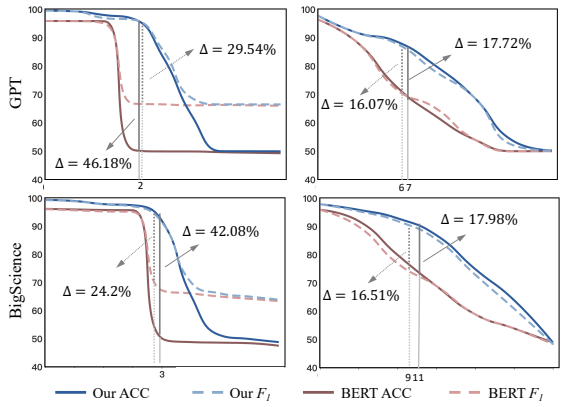


Figure 4: T-SNE visualizations of AI-detection features $\mathbf{a}$ from MLP and DB encoders. With human-AI coloring (top), the encoders separate authenticity. With generator coloring (bottom), MLP features split into distinct generator clusters, while DB suppresses generator bias, yielding compact, generator-invariant representations.

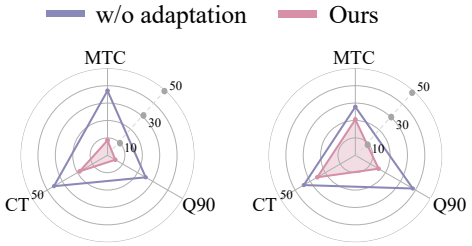
4.4.2 Enhancing Robustness via Cross-View Regularization

To evaluate the robustness benefits of the cross-view regularization module, we inject two types of



(a) Generator-aware Noise Attack (b) Gaussian Noise Attack

Figure 5: Robustness under generator-aware and Gaussian perturbations. Perturbation-based regularization enhances our method's resilience, maintaining superior accuracy over the BERT baseline in both structured and random noise settings.



(a) Generator-aware (b) AI-detection

Figure 6: Intra-class compactness with vs. without adaptation. Adaptation consistently improves feature compactness across tasks, yielding lower dispersion metrics and more task-aligned representations.

noise into the AI-detection features: (1) generator-aware perturbations simulating stylistic interference, and (2) Gaussian noise introducing unstructured variation. We compare the performance of model variants under both noise conditions. As shown in Figure 5, our model consistently maintains significantly higher classification accuracy, with improvements of up to 46.18% (on GPT) under generator-style perturbations and 17.98% (on BigScience) under Gaussian noise. These results confirm that cross-view regularization enhances robustness by reducing sensitivity to both structured and unstructured perturbations, thereby improving generalization under distributional shifts.

4.4.3 Improving Compactness via Adaptation

To evaluate the impact of the discriminator-guided adaptation stage, we analyze intra-class compactness of the learned features. Specifically, we compute three metrics: MeanToCenter, Covariance Trace, and 90th Percentile Pairwise Distance, on AI-detection and generator-aware representations extracted from GPT category. As shown in Figure 6, models trained with the adaptation module consistently exhibit lower values across all metrics, indicating tighter clustering and reduced semantic noise. These findings suggest that the adaptation module facilitates alignment, leading to more compact and discriminative feature structures, thereby enhance model generalization.

Components					FLAN-T5	OPT
BERT	Disen.	DB	Regular.	Adapt.	ACC / F1	ACC / F1
✓					60.9 / 52.6	82.1 / 84.2
✓	✓				62.0 / 53.3	83.7 / 83.5
✓	✓	✓			65.2 / 60.2	87.1 / 86.0
✓	✓	✓	✓		66.8 / 63.8	88.3 / 87.4
✓	✓	✓	✓	✓	**68.9 / 64.1**	**89.1 / 88.3**

Table 4: Incremental ablation results on two unseen LLM categories.

4.5 Ablation Study

We conduct an incremental ablation study on the FLAN-T5 and OPT as the held-out evaluation category set (see Table 4). For each test case, the model is trained on the remaining 6 categories and evaluated on the selected one to measure generalization. Adding DB-based encoding provides the largest gain by filtering generator noise and clarifying semantics. Cross-view regularization improves robustness by stabilizing representations under perturbations, and discriminator-guided adaptation further refines decision boundaries for compact, task-aligned features. Overall, each module contributes additive improvements, with DB-based encoding as the most influential.

5 Conclusion

We propose a structured disentanglement framework for detecting AIGT content from unseen generators, tackling distribution shifts and stylistic artifacts. By combining DB-based encoding, cross-view regularization, and discriminator-guided adaptation, our method suppresses generator-aware noise and improves generalization. Extensive experiments demonstrate state-of-the-art cross-generator performance, highlighting the scalability of disentanglement for AIGT detection.

Limitations

While our framework achieves strong cross-generator generalization, several limitations remain. One concern lies in interpretability: although feature visualizations provide some intuition, the specific linguistic or structural cues driving detection decisions are not yet fully understood. Another issue is sensitivity to hyperparameter choices, which may require careful tuning to maintain stable performance across settings. In addition, robustness against adaptive or adversarial attacks has not been systematically examined, and strengthening this aspect would be essential for reliable deployment in real-world applications.

References

Josh Achiam and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *In The Twelfth International Conference on Learning Representations.*

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jiaqi Chen, Xiaoye Zhu, Tianyang Liu, Ying Chen, Chen Xinhui, Yiwen Yuan, Chak Tou Leong, Zuchao Li, Long Tang, Lei Zhang, and 1 others. 2025. Imitate before detect: Aligning machine stylistic preference for machine-revised text detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23559–23567.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35.

Sebastian Gehrmann and 1 others. 2019. Gltr: Statistical detection and visualization of generated text. *In Proceedings of the 57th Annual Meeting ofthe Association for Computational Linguistics: System Demonstrations.*, pages 111–116.

Jiazhi Guan, Yi Zhao, Zhuoer Xu, Changhua Meng, Ke Xu, and Youjian Zhao. 2024. Adversarial robust safeguard for evading deep facial manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 118–126.

Hanxi Guo, Siyuan Cheng, Xiaolong Jin, Zhuo Zhang, Kaiyuan Zhang, Guanhong Tao, Guangyu Shen, and Xiangyu Zhang. 2024. Biscope: Ai-generated text detection by checking memorization of preceding tokens. *Advances in Neural Information Processing Systems*, 37:104065–104090.

Beizhe Hu, Qiang Sheng, Juan Cao, Yuhui Shi, Yang Li, Danding Wang, and Peng Qi. 2024. Bad actor, good advisor: Exploring the role of large language models in fake news detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 22105–22113.

Xiaomeng Hu and 1 others. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in neural information processing systems*, 36:15077–15095.

Guanhua Huang, Yuchen Zhang, Zhe Li, Yongjian You, Mingze Wang, and Zhouwang Yang. 2024. Are ai-generated text detectors robust to adversarial perturbations? *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Zhiwei Jiang, Tianyi Gao, Yafeng Yin, Meng Liu, Hua Yu, Zifeng Cheng, and Qing Gu. 2023. Improving domain generalization for prompt-aware essay scoring via disentangled representation learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12456–12470.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36:27469–27500.

Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and LINGYU DUAN. 2022. Uncertainty modeling for out-of-distribution generalization. In *International Conference on Learning Representations*.

Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. 2018. Domain generalization via

conditional invariant representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Aiwei Liu, Leyi Pan, Xuming Hu, Shuang Li, Lijie Wen, Irwin King, and Philip S. Yu. 2024a. An unforgeable publicly verifiable watermark for large language models. In *The Twelfth International Conference on Learning Representations*.

Shengchao Liu, Xiaoming Liu, Yichen Wang, Zehua Cheng, Chengzhengxu Li, Zhaohan Zhang, Yu Lan, and Chao Shen. 2024b. Does detectgpt fully utilize perturbation? bridging selective perturbation to fine-tuned contrastive learning detector would be better. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE.

Rui Shao and 1 others. 2023. Detecting and grounding multi-modal media manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6904–6913.

Eyal Shnarch, Ariel Gera, Alon Halfon, Lena Dankin, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2022. Cluster & tune: Boost cold start performance in text classification. *In Annual Meeting of the Association for Computational Linguistics.*, page 7639–7653.

Kanishka Silva, Ingo Frommholz, Burcu Can, Frédéric Blain, Raheem Sarwar, and Laura Ugolini. 2024. Forged-gan-bert: Authorship attribution for llm-generated forged novels. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 325–337. Association for Computational Linguistics.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 8384–8395.

Christoforos Vasilatos, Manaar Alam, Talal Rahwan, Yasir Zaki, and Michail Maniatakos. 2023. Howkgpt: Investigating the detection of chatgpt-generated university student homework through context-aware perplexity analysis. *arXiv preprint arXiv:2305.18226*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4: Multi-generator, Multi-domain, and Multilingual Black-Box Machine-Generated Text Detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Zhenhua Wang and 1 others. 2025. A novel benford's law-driven approach for detecting machine-generated text. *ACM Transactions on Information Systems*.

Wenjie Wei, Yanyue Zhang, Jinyan Li, Panfei Liu, and Deyu Zhou. 2025. Cross-domain fake news detection based on dual-granularity adversarial training. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9407–9417.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on llm-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

A Appendix

A.1 Data Preprocessing

We adopt the pre-processed dataset from the MAGE benchmark (Li et al., 2024), which provides both generator categories and domain labels. To ensure consistency with the original distribution, we perform balanced sampling for each category by aligning with the proportions of both generator and domain labels. This design controls for domain effects and encourages the model to learn generalization across generators rather than being confounded by domain-specific biases. For clarity and to avoid redundancy, we exclude several highly similar models, including T0-11B, T0-3B, OPT-1.3B, OPT-125M, OPT-13B, OPT-30B, and OPT-350M.

Section	Train	Test
4.2.1 Main Results (Table 2)	Leave-One-Generator-Out	
4.2.2 Diversity (Table 3)	BigScience GPT LLaMA GLM EleutherAI	OPT FLAN-T5
4.3 Progress Gen. (Figure 3)	BigScience GPT	FLAN-T5
4.4.1 Bias Filter (Fig. 4) A.3 Bias Filter (Fig. 7)	BigScience GPT FLAN-T5	-
4.4.2 Robustness (Figure 5)	BigScience GPT	-
4.4.3 Compactness (Figure 6)	GPT	-
4.5 Ablation Study (Table 4)	$\bigcup_{i \neq j} \text{Test}_i$	FLAN-T5 OPT
A.4 Adver. Attacks (Table 6)	$\bigcup_{i \neq j} \text{Test}_i$	OPT
A.5 Generalization (Figure 8)	$\bigcup_{i \neq j} \text{Test}_i$	OPT FLAN-T5 LLaMA

Table 5: Overview of training and testing setups across different experiments.

A.2 Overview of Training and Testing Setups

Table 5 summarizes the training and testing setups used across all experiments. For each section, we specify the generator categories included in the training set and those held out for testing. In particular, Section 4.2.1 adopts a leave-one-generator-out protocol, while subsequent experiments examine generalization under varying generator diversity, progressive training scenarios, robustness to bias and adversarial perturbations, and ablation analyses. This overview facilitates a clear understanding of how different experimental objectives are mapped to corresponding train–test splits.

A.3 Futher Analysis on Bottleneckd Encoding

To complement the main analysis of AI-detection features, we further examine the generator-aware branch. Figure 7 illustrates the t-SNE visualizations of generator-aware representations under the same experimental setup (GPT, BigScience and FLAN-T5). In the MLP-based baseline, generator-aware features are entangled with authenticity cues, indicating that this branch implicitly encodes AI-vs-human distinctions. In contrast, the DB-based encoding yields clearer generator separation while

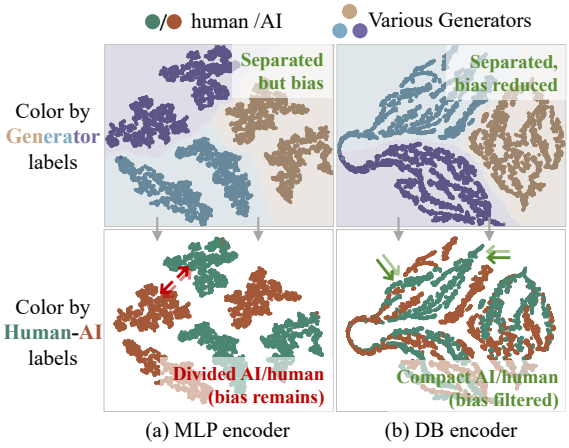


Figure 7: MLP vs. DB encoder representations of generator-aware branch g. With Human–AI coloring (top), both separate distinct generators, but MLP retains human-AI bias while DB filters it. With generator coloring (bottom), MLP forms divided human and AI samples, whereas DB suppresses bias and yields compact, generator-aware clusters.

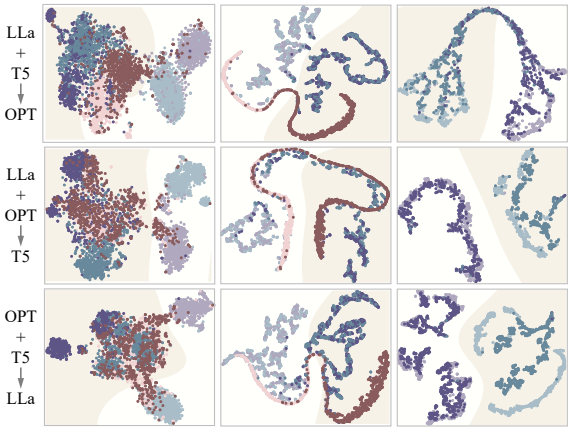
Model	BERT	SCRN	Ours
Metric	ACC / F_1	ACC / F_1	ACC / F_1
Original	82.1 / 84.2	81.3 / 79.2	**89.1 / 88.3**
Delete	68.8 / 72.6	67.6 / 64.9	**80.0 / 79.9**
Swap	**66.9** / 64.2	62.6 / 59.9	66.5 / **69.9**
Insert	66.9 / 65.1	62.6 / 60.9	**67.5 / 70.6**
Replace	66.4 / 65.3	63.5 / 66.9	**74.5 / 75.6**
Average	67.3 / 66.8	64.1 / 63.2	**72.1 / 74.0**

Table 6: Performance comparison among BERT, SCRN, and our method under various perturbation types on unseen OPT category.

suppressing unintended authenticity leakage. This result confirms that our disentanglement framework not only produces more generator-invariant AI-detection features but also enforces semantic purity in the generator-aware branch, thereby strengthening the overall robustness to unseen generators.

A.4 Robustness under Input-Level Adversarial Attacks

To evaluate robustness under input-level perturbations, we apply four token-level noises (deletion, insertion, repetition, replacement) to the test data. As shown in Table 6, our method consistently outperforms BERT and SCRN across all settings. The improvements are most pronounced under deletion: our model achieves 80.0% accuracy, an absolute gain of 11.2% over BERT, and a 15.0% F_1 gain over SCRN. Moreover, while BERT and SCRN



(a) BERT output (b) AI-detection $\boldsymbol{a}$ (c) Generator-aware $\mathbf{g}$

Figure 8: T-SNE visualization of the representation distributions on LLaMA, OPT and FLAN-T5 categories. Models are trained on two generators (blue/purple) and evaluated on the held-out one (red). Dark/light shades for AI and human samples. The baseline BERT representations (a) show good separation for seen generators but entanglement on unseen ones. With disentanglement, the AI-detection branch (b) yields clearer human–AI separation with reduced generator bias, while the generator-aware branch (c) explicitly isolates generator-specific features.

drop by 13.3% and 13.7% respectively compared to clean inputs, our method degrades by only 9.1%, confirming its stronger resilience to adversarial perturbations.

A.5 Generalization Gains from Feature Disentanglement

To assess cross-generator generalization, we conduct a leave-one-generator-out analysis on three MAGE categories (LLaMA, OPT, FLAN-T5). In each run, models are trained on two generators and tested on the remaining unseen one, with representation distributions visualized via t-SNE (Figure 8). As shown in panel (a), BERT features separate human and AI samples on seen generators but fail to generalize to the unseen one, where generator entanglement persists. With disentanglement, the AI-detection branch (panel b) enlarges the human–AI margin while suppressing generator bias, and the generator-aware branch (panel c) captures generator distinctions explicitly. These results demonstrate that disentanglement isolates task-relevant semantics from generator artifacts, leading to improved cross-generator generalization.