IS YOUR WRITING BEING MIMICKED BY AI? UNVEILING IMITATION WITH INVISIBLE WATERMARKS IN CREATIVE WRITING

Anonymous authors
Paper under double-blind review

ABSTRACT

In-Context Learning (ICL) and efficient fine-tuning methods significantly enhance the efficiency of applying Large Language Models (LLMs) to downstream tasks. However, they also raise concerns about the imitation and infringement of authorial creative works. Current copyright protection methods for creative works predominantly focus on visual arts, leaving a critical research gap in the safeguarding of creative writing. In this paper, we propose **WIND** (**W**atermarking through **I**mplicit, **N**on-disruptive **D**isentanglement), a novel zero-watermarking, verifiable and implicit scheme designed to protect the originality of creative writing from unauthorized AI imitation. Specifically, we decompose creative essence into five key elements, which are extracted utilizing LLMs through a designed instance delimitation mechanism and consolidated into condensed-lists. These lists enable WIND to convert core copyright attributes into verifiable watermarks via implicit encoding within a disentanglement creative space, where 'disentanglement' refers to the separation of creative-specific and creative-irrelevant features. This approach, utilizing implicit encoding, avoids distorting fragile textual content. Extensive experiments demonstrate that WIND effectively verifies creative writing copyright ownership against AI imitation, achieving F1 scores above 98% and maintaining robust performance under stringent low false-positive rates where existing state-of-the-art text watermarking methods struggle. The code is available in the supplementary materials.

1 INTRODUCTION

In-context learning (ICL) has emerged as a revolutionary paradigm in natural language processing (NLP), as comprehensively surveyed by Dong et al. (2024). Large language models (LLMs) acquire extensive real-world knowledge through few-shot learning, as demonstrated by Brown et al. (2020), Wei et al. (2022), Liu et al. (2023a), Liu et al. (2024b), and OpenAI (2023). Simultaneously, advancements in efficient parameter fine-tuning methods, such as those proposed by Hu et al. (2022); Liu et al. (2021); Han et al. (2024), enable large language models (LLMs) to adapt effectively to specific downstream tasks with minimal data. These techniques have heightened concerns over copyright infringement of authors' creative works, resulting in a growing number of legal disputes. In the domain of creative writing, for instance, *Authors Guild of America* sued *OpenAI* for training its models on copyrighted texts without permission (AIa (2023)). Similarly, the *New York Times* also filed a lawsuit against *OpenAI* (Tim (2023)). Similar litigation has arisen in other creative domains, such as visual arts (Sar (2023); Get (2023)). Consequently, the protection of creative works has drawn significant attention from researchers Liu et al. (2023b); Tang et al. (2023); Maini et al. (2024).

Current methods for ensuring copyright protection in creative works primarily focus on digital watermarking, a widely studied and validated paradigm for safeguarding data and preventing infringement. Several studies, such as Chen et al. (2022); Salman et al. (2023); Shan et al. (2023), have explored scrambled watermarks, which involve embedding intentional signals into images to protect authors' creative visual arts. Alternatively, recent work by Huang et al. (2024b) explores verifiable watermarks using diffusion models to establish clear copyright boundaries for image style protection. Although current watermarking methods tailored to creating works focus primarily on visual arts, the preservation of creative writing remains underexplored.

To defend against unauthorized imitation by LLMs, we aim to construct a verifiable watermark that can authenticate the authorship of creative writing. Our core motivation is to verify whether a given text constitutes an unauthorized imitation of a specific author's protected creative signature. While this objective connects to the well-developed domain of authorship identification as examined in Huang et al. (2024a), our approach is fundamentally different. The former focuses on identifying the writer of a given text, whereas our work aims to proactively protect an author's creative essence by generating a verifiable watermark. This shifts the problem from mere classification to robust infringement verification. The process first requires the extraction of the distinctive writing traits that characterize the author. These unique features then form the basis for encoding a watermark that preserves the authorship identity. To this end, we draw on key insights from linguistic research by Vaezi & Rezaei (2019);

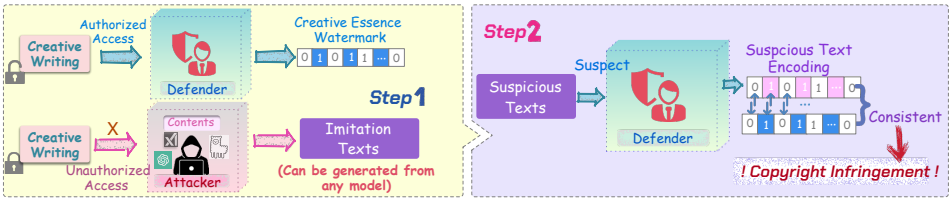


Figure 1: The application scenario of the implicitly verifiable watermark.

Tweedie & Baayen (1998); Lu (2010); Steen et al. (2010); Pennebaker & King (1999); Kao & Jurafsky (2012), we define the creative writing signature through five key elements: (1) vocabulary and word choice, (2) syntactic structure and grammatical features, (3) rhetorical devices and stylistic choices, (4) tone and sentiment, and (5) rhythm and flow. Leveraging this representation, we propose **WIND**, a **W**atermarking scheme based on **I**mplicit, **N**on-disruptive **D**isentanglement. The scheme maps above traits into a verifiable and implicit watermark within a disentangled creative space, preserving the fragile writing style while enabling reliable ownership verification.

Specifically, building on the findings of Leidinger et al. (2023) regarding the impact of prior knowledge selection on the parsing ability of LLMs, we first construct two prompt templates, incorporate contrastive learning, and develop an instance delimitation mechanism to select the most suitable prompt for each sample. Next, to disentangle creation-specific and creation-irrelevant features, we employ LLMs to extract the five elements to establish the creative writing essence, termed condensed-lists. Finally, we convert these lists into the disentangled space to calculate an anchor, which is then implicitly encoded as a verifiable watermark for creative writing.

The application scenario of the proposed WIND is shown in Figure 1. WIND (the Defender) generates a unique watermark via the extracted creative-specific features from the protected data. If an unauthorized attacker accesses protected data to generate imitation texts, the defender could detect infringement by verifying whether the suspect samples exhibit material similarity to the creative writing. This is achieved by measuring consistency through Hamming distance. Additionally, to meet practical needs and reduce computation costs, WIND is designed to perform effectively in few-shot scenarios. Our main contributions are summarized as follows:

- We present a novel, verifiable and implicit watermarking method, namely WIND, to protect creative writing copyrights from unauthorized AI imitation. To the best of our knowledge, this is the first work to formally decompose the abstract essence of creative writing and leverage it for copyright protection via watermarking.
- We create an instance delimitation mechanism to identify optimal prior knowledge, which facilitates the extraction of condensed-lists by LLMs. Subsequently, we establish a verifiable watermark domain for creative essence, moving beyond injecting signal methods that will potentially harm the fragile style.
- Extensive experiments confirm the method's effectiveness and robustness against a wide range of challenges, including state-of-the-art text watermarking techniques, robust attacks, and multi-level visual analysis.

2 Related Work

2.1 Creative Work Copyright Protection

Existing methods for detecting infringements of creative works mainly fall into two categories: membership inference (MI) and digital watermarking. For MI-based approaches, Shi et al. (2024) compare data generated before and after model training, while Maini et al. (2024) perform membership inference attacks (MIAs) in gray-box settings. However, these methods struggle when LLMs imitate the unique creation but modify irrelevant details, and gray-box model access limits real-world applications.

Digital watermarks for copyright protection are mainly of two types: scrambled and verifiable watermarks. Scrambled watermarks embed distorted signals in data to protect visual art but are vulnerable at the latent representation level. For example, Chen et al. (2022) reversibly transform images into adversarial examples. Salman et al. (2023) add imperceptible adversarial perturbations to protect images from malicious editing. Shan et al. (2023) apply subtle cloaks to images for similar protection. Verifiable watermarks, however, offer a stronger defense against unauthorized use. Huang et al. (2024b) address image style infringement in the text-to-image process. Current methods for protecting creative writing are insufficient.

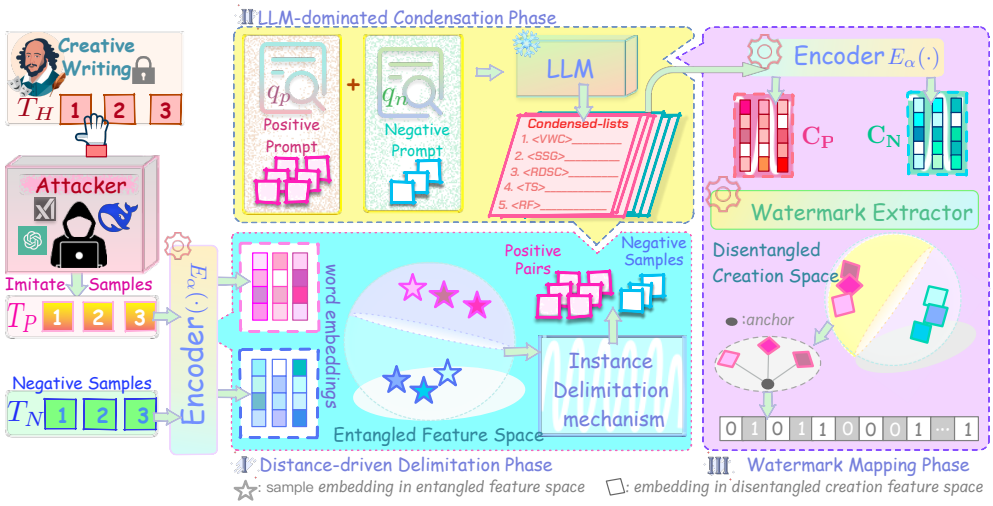


Figure 2: The overall framework of WIND, which consists of three main phases: (1) Distance-driven Delimitation, (2) LLM-dominated Condensation, and (3) Watermark Mapping. The numbers within the squares of $\boldsymbol{T_H}$, $\boldsymbol{T_P}$, and $\boldsymbol{T_N}$ represent different samples. Additionally, a star in the entangled feature space and a diamond in the disentangled space (of the same color) denote the same sample.

2.2 Text Watermarking for LLMs

Text watermarking in LLMs can be considered a protective scheme for the learned creative styles of LLMs, as demonstrated by He et al. (2022). These watermarks are typically embedded during the text generation phase, primarily through modifications to either token sampling or the model's internal logic. Christ et al. (2024) use random sequences for token sampling. Kuditipudi et al. (2023) apply Levenshtein distance for text-number matching. In contrast, the KGW method (Kirchenbauer et al. (2023)) biases green-listed tokens. Zhao et al. (2024) propose a universal green list, while Lu et al. (2024) weight high-entropy tokens. Meanwhile, Hu et al. (2023) introduce unbiased watermarks for attribution. However, these methods are insufficient to safeguard the creation of ideas. Therefore, in this paper, we introduce WIND, a novel framework that employs LLMs to extract the creative essence as condensed lists, which are then utilized to generate an implicit watermark for copyright verification without altering the fragile creative attributes.

3 Approach

3.1 Problem Formulation

Creative Writing Essence. We first define the protected creative writing ($\boldsymbol{T_H}$) as a collection of human-authored texts, such as *Shakespeare's Hamlet*. Next, we operationalize the abstract concept of "creative writing essence" by decomposing it into five concrete elements, grounded in linguistic research Vaezi & Rezaei (2019); Tweedie & Baayen (1998); Lu (2010); Steen et al. (2010); Pennebaker & King (1999); Kao & Jurafsky (2012), including vocabulary and word choice (VWC), syntactic structure and grammatical features (SSGF), rhetorical devices and stylistic choices (RDCS), tone and sentiment (TS), and rhythm and flow (RF). See Appendix A.1 for details.

Attackers. Attackers are equipped with two abilities. Firstly, they can gain unauthorized access to valuable data sets like books or weblogs, enabling them to imitate the creative essence. Furthermore, attackers can provide LLM APIs that effectively hide the details of their imitation behaviors.

Defender. Our defense objective is to prevent unauthorized AI imitation and verify copyright ownership. Our defender $D(\cdot)$ generates an implicit watermark to protect creative writing $\boldsymbol{T_H}$. To address statistical biases between human and machine-generated texts, we create two sets of machine-generated texts: $\boldsymbol{T_P}$, which mimics $\boldsymbol{T_H}$, and $\boldsymbol{T_N}$, which represents unprotected writing. For a suspicious text $\boldsymbol{T_{test}}$, we compute the distance between its corresponding watermark and the implicit watermark of $\boldsymbol{T_H}$ to obtain a probability pr, where $pr = 1$ indicates that the text imitates $\boldsymbol{T_H}$.

$$pr = \begin{cases} 1, & \text{if } d_h(D(\boldsymbol{T_{test}}), D(\boldsymbol{T_P})) < \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where d_h denotes Hamming distance and ϵ empirically is 1% of the length of watermark.

3.2 OVERVIEW

The training process of WIND is depicted in Figure 2. Our pipeline operates through three sequential phases designed to progressively isolate and encode the creative essence. The process begins with a Distance-driven Delimitation Phase, which constructs a feature space to identify samples that closely emulate the protected creative style from those that do not, thus preparing an optimal contextual prior for subsequent distillation. This is followed by an LLM-dominated Condensation Phase, which is the core of our disentanglement approach. Here, we leverage the powerful generative prior of a large language model, conditioned on the previously identified context, to interpret and distill the abstract creative essence into a structured set of concrete stylistic elements. Finally, the Watermark Mapping Phase transforms this structured stylistic information into a compact and verifiable zero-watermark, ensuring robust copyright verification.

3.3 DISTANCE-DRIVEN DELIMITATION PHASE

We employ the encoder with la layers and adjustable parameters α, denoted as $E_\alpha(\cdot)$, to compute word embeddings for $\boldsymbol{T_P}$ and $\boldsymbol{T_N}$. Each sentence $t_{pi} \in \boldsymbol{T_P}$ and $t_{nj} \in \boldsymbol{T_N}$ is mapped into a positive feature vector $\boldsymbol{p_i} \in \mathbb{R}^{b_i \times la}$ and a negative feature vector $\boldsymbol{n_j} \in \mathbb{R}^{b_j \times la}$, respectively, with b_i and b_j representing the number of words in t_{pi} and t_{nj}. Assuming both $\boldsymbol{T_P}$ and $\boldsymbol{T_N}$ contain num samples, their corresponding feature vector sets are denoted as $\boldsymbol{P} = [\boldsymbol{p_1}, \boldsymbol{p_2}, \ldots, \boldsymbol{p_{num}}]$ and $\boldsymbol{N} = [\boldsymbol{n_1}, \boldsymbol{n_2}, ..., \boldsymbol{n_{num}}]$, respectively. Both $\boldsymbol{P}$ and $\boldsymbol{N}$ inherently include creation-irrelevant features. In this context, texts from the $\boldsymbol{T_P}$ are considered positive, while all other texts are negative. Next, for any feature vector $\boldsymbol{x} \in \boldsymbol{P} \cup \boldsymbol{N}$, we use the cosine similarity function $sim(\cdot)$ to identify the most similar vector to x from the union of $\boldsymbol{P}$ and $\boldsymbol{N}$, i.e., $\boldsymbol{y_x^*} = \arg\max_{\boldsymbol{y} \in \boldsymbol{P} \cup \boldsymbol{N} \setminus \{\boldsymbol{x}\}} sim(\boldsymbol{x}, \boldsymbol{y})$, with the highest similarity expressed as $sim_x^* = sim(\boldsymbol{x}, \boldsymbol{y_x^*})$. Then the cross-entropy loss $\mathcal{L}_{ce}$ is calculated:

$$\mathcal{L}_{ce} = \frac{1}{2 \times num} \sum_{x \in \boldsymbol{P} \cup \boldsymbol{N}} \mathrm{H}(y_x, \hat{y}_x), \tag{2}$$

where $\mathrm{H}(\cdot)$ represents the entropy function. y_x is ground-truth of sample $\boldsymbol{x}$. $\hat{y}_x$ is the pseudo-label determined by the class of the most similar vector $\boldsymbol{y_x^*}$. Specifically, $\hat{y}_x = 1$ holds when $\boldsymbol{y_x^*} \in \boldsymbol{P}$, otherwise, $\hat{y}_x = 0$. Moreover, to emphasize the distinctions between positive and negative samples, we design a hyperparameter mar and utilize a contrastive loss function:

$$\mathcal{L}_{con} = \frac{1}{2 \times num} \Big(\sum_{\boldsymbol{x}, \boldsymbol{x'} \in \boldsymbol{P}} \|\boldsymbol{x} - \boldsymbol{x'}\|^2 + \sum_{\boldsymbol{x} \in \boldsymbol{N}, \boldsymbol{x''} \in \boldsymbol{P}} \max(0, mar - \|\boldsymbol{x} - \boldsymbol{x''}\|^2) \Big). \tag{3}$$

The importance of optimal reference instance selection in prompt engineering has been empirically established by Sahoo et al. (2024). Taking this viewpoint, we introduce an instance delimitation mechanism to select the optimal prior knowledge for each sample. Note that for each $\boldsymbol{x}$, the most similar vector $\boldsymbol{y_x^*}$ may come from either set $\boldsymbol{P}$ or $\boldsymbol{N}$. We construct two sets: one is the positive pair set $\boldsymbol{pp}$, and the other is the negative sample set $\boldsymbol{neg}$. Specifically, $\boldsymbol{pp}$ contains samples that closely emulate $\boldsymbol{T_H}$, where the most similar sample (with similarity exceeding a predefined threshold) is labeled as the positive instance. Each sample in $\boldsymbol{pp}$ is paired with its corresponding optimal prior knowledge $\boldsymbol{y_x^*}$, allowing better disentanglement of the creation-specific features that distinguish protected creative works from unprotected ones. In contrast, $\boldsymbol{neg}$ is composed of individual samples instead of pairs due to the diverse creative works in $\boldsymbol{T_N}$, whereas $\boldsymbol{T_P}$ sentences uniformly exhibit the creative essence. The assignments for $\boldsymbol{pp}$ and $\boldsymbol{neg}$ are formalized in the corresponding equations:

$$\boldsymbol{pp} = \{(\boldsymbol{x}, \boldsymbol{y_x^*}) \mid \boldsymbol{x} \in \boldsymbol{P} \cup \boldsymbol{N} \wedge \boldsymbol{y_x^*} \in \boldsymbol{P} \wedge sim_x^* > \sigma\}, \tag{4}$$

$$\boldsymbol{neg} = \{\boldsymbol{x} \mid \boldsymbol{x} \in \boldsymbol{P} \cup \boldsymbol{N} \wedge (\boldsymbol{y_x^*} \in \boldsymbol{N} \vee sim_x^* \leq \sigma)\}, \tag{5}$$

where σ is the pre-defined threshold, and sim_x^* denotes the similarity between $\boldsymbol{x}$ to the most similar sample. Notably, we design an instance delimitation mechanism rather than direct label prediction via $E_\alpha(\cdot)$, primarily to mitigate misclassification by $E_\alpha(\cdot)$, as demonstrated in Section 4.2.

3.4 LLM-DOMINATED CONDENSATION PHASE

To further disentangle the creative essence, we employ an LLM to extract condensed-lists composed of five elements, as detailed in Section 3.1. Specifically, we design two prompt templates, q_p and q_n, where q_p is designed for samples in the positive pair set $\boldsymbol{pp}$, and q_n is used for the negative sample set $\boldsymbol{neg}$. For each sample $t_m \in \boldsymbol{T_P} \cup \boldsymbol{T_N}$, we start by appending the sample to its corresponding prompt, creating a complete input sequence notated as $q||t_m$. Here, $q = q_p$ when $E_\alpha(t_m)$ is part of $\boldsymbol{pp}$ and $q = q_n$ for the samples in $\boldsymbol{neg}$. The concatenated input $q_i||t_m$ is then fed into a frozen-parameter LLM, denoted as $G(\cdot)$, which generates a condensed-list $\boldsymbol{c} = [e_1, e_2, \ldots, e_5]$ for each sample, where each e_i represents one of five key elements per sample. Prompt constructions are in Appendix A.2.

3.5 WATERMARK MAPPING PHASE

In the preceding stages, LLM is used to extract the condensed-lists. These lists are then further transformed into positive disentangle embeddings $\boldsymbol{C_P}$ and negative embeddings $\boldsymbol{C_N}$ through the encoder $E_\alpha(\cdot)$. It is worth noting that this encoder is identical to the one used in the first step. We then employ the sigmoid function $\theta(\cdot)$ and a learnable watermark matrix $\boldsymbol{M}_\gamma \in \mathbb{R}^{len \times la}$ to construct the watermark extractor, where γ denotes the learnable parameters and len is the fixed watermark length. Each condensed-list $\boldsymbol{c_m}$ is processed according to the formula:

$$\boldsymbol{w_m} = \theta(\boldsymbol{M}_\gamma \cdot E_\alpha(\boldsymbol{c_m})), \tag{6}$$

where $\boldsymbol{w_m} \in \boldsymbol{W}$ and $\boldsymbol{W} \in \mathbb{R}^{2num \times len}$. The anchor $\boldsymbol{a}$ is computed as $\boldsymbol{a} = \frac{1}{l}\sum_{i=1}^{i<=l} \boldsymbol{w_i}$ and l represents the length of $\boldsymbol{pp}$. Notably, $\boldsymbol{w_{au}} = [r(a_1), r(a_2), ..., r(a_{len})]$ denotes the verifiable and implicit watermark for the protected creative writing, where $r(\cdot)$ acts as a hard thresholding function, converting its input into a bit string. We anticipate that all samples imitating $\boldsymbol{T_H}$, after being mapped by $\boldsymbol{M}_\gamma$, will closely converge in a disentangled creative feature space. To quantify this convergence, we introduce a regularization penalty, denoted as $\mathcal{L}_o$, to measure the average distance between the positive samples and $\boldsymbol{a}$. The calculation is as follows:

$$\mathcal{L}_o = \frac{1}{l}\sum_{i=1}^{i<=l} \|\boldsymbol{w_i} - \boldsymbol{a}\|^2. \tag{7}$$

3.6 TRAINING PROCEDURE

For each instance $t_m \in \boldsymbol{T_P} \cup \boldsymbol{T_N}$, we assign $\boldsymbol{W_P} = \{\boldsymbol{w_m} | t_m \in \boldsymbol{T_P}\}$ to signify the vectors in the disentangled creation space. Ideally, all samples from $\boldsymbol{T_P}$ should be mapped to anchor $\boldsymbol{a}$. To rigorously evaluate the performance of the encoder $E_\alpha(\cdot)$ and the watermark matrix $\boldsymbol{M}_\gamma$, we employ Binary Cross-Entropy (BCE) loss:

$$\mathcal{L}_w = BCELoss(\boldsymbol{W_p}, \boldsymbol{a}). \tag{8}$$

Accordingly, the total loss for WIND is:

$$\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{con} + \mathcal{L}_o + \mathcal{L}_w. \tag{9}$$

The training procedure is summarized in Algorithm 1.

Algorithm 1: Training Procedure of WIND

Data: Protected creative writing $\boldsymbol{T_H}$, imitation texts $\boldsymbol{T_P}$, unprotected texts $\boldsymbol{T_N}$, encoder $E_\alpha(\cdot)$, similarity function $d(\cdot)$, watermark matrix $\boldsymbol{M}_\gamma$, sigmoid $\theta(\cdot)$, LLM $G(\cdot)$, prompts q_p, q_n, R episodes and ep epochs.
Result: Updated parameters α, γ.
for $epoch \leftarrow 1$ **to** ep **do**
  foreach $episode \in R$ **do**
    foreach $t_m \in \boldsymbol{T_P} \cup \boldsymbol{T_N}$ **do**
      $\boldsymbol{x} = E_\alpha(t_m)$, $\boldsymbol{y_x^*} = \text{argmax}_{\boldsymbol{y} \in \boldsymbol{P} \cup \boldsymbol{N} \setminus \{\boldsymbol{x}\}} sim(\boldsymbol{x}, \boldsymbol{y})$
    Construct $\boldsymbol{pp}, \boldsymbol{neg}$ using Eq.4 and 5, **foreach** $t_m \in \boldsymbol{T_P} \cup \boldsymbol{T_N}$ **do**
      $\boldsymbol{c_m} = \boldsymbol{x} \in \boldsymbol{pp}?G(q_p \mid t_m) : G(q_n \mid t_m)$
      Compute $\boldsymbol{w_m}$ (Eq.6)
    Compute $\mathcal{L}_{ce}$ (Eq.2), $\mathcal{L}_{con}$ (Eq.3), Calculate $\mathcal{L}_o$ (Eq.7), $\mathcal{L}_w$ (Eq.8)
    Update α, γ with overall loss $\mathcal{L}$ (Eq.9)

3.7 WATERMARK VALIDATION

The goal of watermark validation is to generate a verifiable watermark for a given suspicious text to confirm copyright ownership. During testing, upon receiving the input sentence t_{test}, we identify the most similar sample $\boldsymbol{y_{test}^*}$ from the training dataset. We then extract the condensed list $\boldsymbol{c_{test}}$, which consists of five elements, by leveraging the frozen-parameters LLM $G(\cdot)$ with the optimal combined input. The $\boldsymbol{c_{test}}$ is formulated as follows:

$$\boldsymbol{c_{test}} = G(q||t_{test}). \tag{10}$$

Here, q is the prompt template that depends on classification of t_{test} as either $\boldsymbol{pp}$ or $\boldsymbol{neg}$, based on the instance delimitation mechanism specified in Section 3.3. Subsequently, $\boldsymbol{c_{test}}$ is mapped into the disentangled feature space, facilitating the extraction of unique creation features represented as $\boldsymbol{w_{test}} = \theta(\boldsymbol{M}_\gamma \cdot E_\alpha(\boldsymbol{c_{test}}))$. This process quantifies the similarity that the tested sample t_{test} imitates the protected creative writing, formulated as follows:

$$\mathcal{P}(\boldsymbol{w_{test}}|\boldsymbol{a}) = \frac{\sum_{i=1}^{len} \mathbb{I}(r(w_{test}^i) = r(a^i))}{len}. \tag{11}$$

Herein, $\mathbb{I}(\cdot)$ symbolizes an indicator function, assuming a value of 1 contingent upon the equality $r(w^i_{test}) = r(a^i)$. To establish a robust mathematical foundation for copyright verification, $\mathcal{P}$ approaches 1 when t_{test} imitates creative writing and approaches 0 otherwise. See Appendix C for Algorithm 2.

4 EXPERIMENTS

4.1 DATASET AND EXPERIMENTAL SETTING

In our experiments, we use two stylistically distinct human-written datasets as protected creative writing $\boldsymbol{T_H}$: Shakespeare (SP) and ROCStories (ROC) (Zhu et al. (2023)). Additionally, the IMDB dataset (Dai et al. (2019)) serves as one of the negative creative works. For example, when the protected set $\boldsymbol{T_P}$ consists of LLM-rewritten ROC texts, the negative set $\boldsymbol{T_N}$ includes LLM-rewritten SP and IMDB texts. Rewritten texts are generated utilizing three language models: GPT-3.5-turbo-16k (GPT3.5) (Brown et al. (2020)), Grok-beta[1] (Grok), and OPT-1.3B (Zhang et al. (2022)) (OPT). Note that OPT is included solely in the watermark baseline. All tabulated values represent the mean results from three experimental trials. Unless otherwise specified, the default experiment uses 'Grok' to generate imitation texts and 10 samples from the protected and unprotected creative writing, respectively.

To evaluate performance, we measure the True Positive Rate (TPR), False Positive Rate (FPR; ideal: 0), and F1 score (F1). We employ SimCSE-RoBERTa proposed by Gao et al. (2021) as the encoder, and our model has a total parameter size of 356.41 M. The optimization is conducted using the AdamW (Loshchilov (2017)) optimizer, with the Encoder $E_\alpha(\cdot)$ learning rate dynamically adjusted from 5e-5 to 1e-7, and the learning rate of Watermark Extractor $\boldsymbol{M}_\gamma$ fixed at 1e-5. Implementation details of the chip, time complexity, hyperparameters and key statistics of the datasets are provided in the Appendix D.2 and D.1.

4.2 MAIN RESULTS

Our main experiment addresses two key research questions. **[Q1:] Is a specialized approach like WIND required for creative writing protection?** To explore this, we utilize three powerful fine-tuned classifiers sufficient for a nuanced style protection task, including BERT-base-uncased (BERT) (Devlin et al. (2019)), RoBERTa (Liu (2019)), and T5 (Raffel et al. (2020)). The results are presented in Table 1 and 2. There are three main findings: (1) Overall, WIND surpasses baseline models in safeguarding creative writing, while also exhibiting a lower standard deviation. (2) WIND achieves satisfactory F1 scores and minimal FPR with just six protected style samples, whereas the baseline models perform nearly at random guessing levels. (3) When using one LLM as $G(\cdot)$ to detect texts generated by another LLM, there is a slight performance degradation due to distribution differences in machine-generated texts. However, even with this, our WIND still demonstrates excellent performance.

Table 1: Performance assessment of WIND and baselines when safeguarding the creative writing SP. Each baseline model is fine-tuned with num samples from protected creative writing and negative texts. 'GPT3.5' and 'Grok' denote which LLM generates imitation texts. WIND-3.5, WIND-G and WIND-D (marked in blue) signify the use of GPT3.5, Grok, and DeepSeek-V3 (Liu et al. (2024a)) as $G(\cdot)$ to obtain creative essence. Results show means ± standard deviations over.

		GPT3.5			Grok		
num	Methods	F1	TPR	FPR	F1	TPR	FPR
6	BERT	$60.12_{6.69}$	$59.28_{0.24}$	$26.0_{7.51}$	$72.87_{9.88}$	$74.04_{2.12}$	$29.75_{2.04}$
	RoBERTa	$61.21_{7.43}$	$63.31_{11.06}$	$8.02_{6.51}$	$75.58_{6.68}$	$80.71_{7.3}$	$33.31_{1.28}$
	T5	$45.93_{2.7}$	$51.32_{4.14}$	$35.27_{4.29}$	$48.12_{4.26}$	$62.70_{4.55}$	$46.32_{1.19}$
	WIND-3.5	$94.72_{1.13}$	$90.01_{1.58}$	$\mathbf{2.99_{1.23}}$	$89.31_{4.67}$	$83.02_{6.05}$	$2.23_{2.02}$
	WIND-G	$\mathbf{94.73_{0.92}}$	$\mathbf{96.04_{1.65}}$	$7.27_{1.89}$	$93.59_{2.37}$	$92.04_{4.32}$	$2.61_{1.40}$
	WIND-D	$93.91_{3.05}$	$92.05_{2.10}$	$6.24_{4.56}$	$\mathbf{96.16_{2.01}}$	$\mathbf{93.33_{4.17}}$	$\mathbf{0.67_{0.94}}$
10	BERT	$68.32_{5.53}$	$64.71_{8.75}$	$3.34_{3.46}$	$75.62_{6.81}$	$90.75_{2.53}$	$53.68_{9.82}$
	RoBERTa	$88.71_{7.91}$	$95.02_{6.02}$	$25.7_{6.54}$	$76.97_{4.54}$	$89.59_{5.72}$	$38.13_{5.28}$
	T5	$67.34_{0.95}$	$91.38_{7.76}$	$78.04_{8.28}$	$56.91_{4.84}$	$58.08_{8.59}$	$15.73_{1.67}$
	WIND-3.5	$\mathbf{98.02_{0.88}}$	$96.02_{4.35}$	$2.03_{2.84}$	$95.22_{0.57}$	$90.71_{0.94}$	$1.34_{1.19}$
	WIND-G	$96.0_{1.6}$	$96.05_{1.62}$	$4.79_{1.92}$	$\mathbf{99.04_{1.13}}$	$\mathbf{99.32_{1.24}}$	$\mathbf{1.13_{1.28}}$
	WIND-D	$97.32_{2.67}$	$\mathbf{96.54_{1.98}}$	$\mathbf{0.67_{0.94}}$	$97.65_{2.08}$	$97.33_{2.49}$	$2.00_{1.63}$
20	BERT	$90.73_{5.25}$	$84.71_{9.76}$	$1.39_{1.88}$	$90.82_{2.68}$	$96.75_{0.59}$	$10.71_{8.66}$
	RoBERTa	$91.80_{5.79}$	$89.76_{3.82}$	$8.91_{3.80}$	$92.75_{1.04}$	$93.22_{5.36}$	$6.75_{2.28}$
	T5	$73.92_{7.34}$	$72.04_{8.31}$	$5.32_{4.07}$	$86.42_{3.87}$	$88.02_{8.51}$	$17.02_{2.34}$
	WIND-3.5	$\mathbf{98.51_{0.56}}$	$97.02_{1.57}$	$2.04_{2.80}$	$96.30_{1.18}$	$93.72_{1.85}$	$1.82_{0.96}$
	WIND-G	$96.32_{2.14}$	$\mathbf{97.35_{2.52}}$	$3.37_{1.91}$	$97.76_{1.42}$	$97.58_{0.91}$	$2.19_{1.17}$
	WIND-D	$97.89_{1.33}$	$96.58_{0.91}$	$\mathbf{0.49_{0.79}}$	$\mathbf{98.12_{0.92}}$	$\mathbf{97.63_{2.60}}$	$\mathbf{0.43_{1.02}}$

[1]https://console.x.ai

These experiments underscore the necessity of specialized approaches like WIND, which learn and verify unique stylistic signatures. Baseline models incorporate content irrelevant to creativity into their feature space, causing them to prioritize generic content over stylistic essence. This bias leads to protection failures and motivates our proposed instance delimitation mechanism for sample classification. Overall, WIND extracts a creative essence watermark that verifies copyright origin, delivering superior accuracy and reliability.

Table 2: Performance assessment of WIND and baselines when safeguarding the creative writing ROC.

		GPT3.5			Grok		
num	Methods	F1	TPR	FPR	F1	TPR	FPR
6	BERT	$65.03_{10.75}$	$65.31_{7.88}$	$33.79_{9.43}$	$61.21_{7.84}$	$71.75_{6.61}$	$49.32_{6.06}$
	RoBERTa	$66.81_{4.29}$	$88.02_{2.76}$	$76.73_{7.08}$	$86.43_{3.19}$	$\mathbf{99.31_{0.94}}$	$45.02_{1.34}$
	T5	$38.48_{2.12}$	$40.74_{3.13}$	$24.04_{2.37}$	$39.88_{4.01}$	$46.05_{2.59}$	$43.91_{3.87}$
	WIND-3.5	$94.47_{2.24}$	$96.66_{3.38}$	$7.08_{1.39}$	$97.27_{2.25}$	$97.63_{4.68}$	$1.97_{2.82}$
	WIND-G	$\mathbf{96.16_{1.57}}$	$95.03_{1.58}$	$\mathbf{4.32_{2.41}}$	$\mathbf{98.73_{1.68}}$	$97.24_{3.8}$	$\mathbf{0.39_{0.92}}$
	WIND-D	$92.31_{2.72}$	$\mathbf{96.89_{3.02}}$	$8.96_{4.60}$	$96.16_{2.00}$	$93.67_{3.77}$	$1.09_{0.23}$
10	BERT	$69.05_{3.81}$	$64.05_{4.27}$	$14.69_{7.80}$	$74.38_{4.91}$	$73.72_{3.55}$	$10.74_{4.39}$
	RoBERTa	$86.69_{2.58}$	$87.32_{3.78}$	$23.29_{4.34}$	$87.82_{1.54}$	$95.75_{2.93}$	$10.34_{5.21}$
	T5	$54.62_{7.69}$	$68.75_{4.52}$	$48.79_{3.90}$	$34.20_{6.15}$	$33.02_{3.07}$	$20.65_{3.18}$
	WIND-3.5	$\mathbf{97.43_{0.65}}$	$\mathbf{98.32_{1.72}}$	$\mathbf{3.38_{1.29}}$	$98.04_{2.02}$	$97.24_{2.46}$	$1.09_{1.65}$
	WIND-G	$94.05_{0.89}$	$98.01_{1.67}$	$6.98_{1.76}$	$\mathbf{98.91_{2.57}}$	$\mathbf{99.48_{2.13}}$	$1.25_{0.46}$
	WIND-D	$94.93_{4.28}$	$93.67_{3.54}$	$3.96_{0.78}$	$96.55_{1.98}$	$94.00_{2.83}$	$\mathbf{0.67_{0.94}}$
20	BERT	$96.05_{0.79}$	$96.02_{2.76}$	$4.02_{3.39}$	$96.42_{3.59}$	$96.02_{2.38}$	$4.19_{3.27}$
	RoBERTa	$87.05_{3.62}$	$90.08_{4.66}$	$16.41_{1.65}$	$94.79_{3.15}$	$94.73_{3.39}$	$3.70_{3.53}$
	T5	$86.21_{3.44}$	$90.76_{7.74}$	$22.02_{7.55}$	$85.27_{5.91}$	$90.73_{7.71}$	$21.72_{2.34}$
	WIND-3.5	$\mathbf{96.05_{0.21}}$	$\mathbf{96.08_{1.57}}$	$2.04_{0.24}$	$96.81_{0.47}$	$97.33_{2.67}$	$1.54_{1.42}$
	WIND-G	$99.66_{0.53}$	$99.27_{0.83}$	$\mathbf{0.33_{0.48}}$	$\mathbf{98.99_{0.26}}$	$\mathbf{98.62_{0.46}}$	$1.41_{0.32}$
	WIND-D	$95.36_{2.09}$	$94.76_{1.77}$	$2.14_{0.96}$	$97.60_{0.98}$	$95.33_{1.89}$	$\mathbf{0.00_{0.00}}$

Table 3: Comparison of WIND (marked in blue) with SOTA Watermarking Methods, where "SP" and "ROC" denote the protected creative writing, respectively. Post-arrow values show performance gaps.

	FPR@%10				FPR@%1			
	SP		ROC		SP		ROC	
	TPR	F1	TPR	F1	TPR	F1	TPR	F1
KGW	$93.87_{\downarrow 4.51}$	$92.92_{\downarrow 6.10}$	$97.17_{\downarrow 0.84}$	$95.24_{\downarrow 3.56}$	$89.80_{\downarrow 8.57}$	$94.62_{\downarrow 4.61}$	$88.03_{\downarrow 10.64}$	$94.13_{\downarrow 5.08}$
Unigram	$94.37_{\downarrow 4.01}$	$92.47_{\downarrow 6.55}$	$96.13_{\downarrow 1.88}$	$93.03_{\downarrow 5.86}$	$89.58_{\downarrow 8.79}$	$94.50_{\downarrow 4.73}$	$91.17_{\downarrow 7.50}$	$88.99_{\downarrow 10.22}$
EWD	$93.83_{\downarrow 4.55}$	$94.73_{\downarrow 4.29}$	$88.27_{\downarrow 9.74}$	$88.89_{\downarrow 10.00}$	$95.65_{\downarrow 2.72}$	$97.78_{\downarrow 1.45}$	$88.02_{\downarrow 10.65}$	$93.61_{\downarrow 5.60}$
SynthID	$78.89_{\downarrow 19.49}$	$75.38_{\downarrow 23.64}$	$85.33_{\downarrow 12.68}$	$86.78_{\downarrow 12.11}$	$78.52_{\downarrow 19.85}$	$79.03_{\downarrow 20.20}$	$84.71_{\downarrow 13.69}$	$69.15_{\downarrow 30.06}$
Unbiased	$38.14_{\downarrow 60.2}$	$51.35_{\downarrow 47.67}$	$50.14_{\downarrow 47.87}$	$62.50_{\downarrow 36.39}$	$14.23_{\downarrow 84.14}$	$24.56_{\downarrow 74.67}$	$16.00_{\downarrow 82.67}$	$27.59_{\downarrow 71.62}$
WIND-G	**98.38**	**99.02**	**98.01**	**98.89**	**98.37**	**99.23**	**98.67**	**99.21**

[Q2:] Is WIND superior to state-of-the-art watermarking methods? The baseline SOTA watermarking schemes we compare include: KWG (Kirchenbauer et al. (2023)), Unigram (Zhao et al. (2024)), EWD (Lu et al. (2024)), SynthID (Dathathri et al. (2024)), and Unibased (Hu et al. (2023)). We use OPT-1.3B to generate watermarked texts for protected creative writing and non-watermarked negative samples, with implementation details provided in Appendix D.2. For fairness, WIND's data is also generated by the same model. Additionally, we set the FPR below 10% and 1% for our recordings. Table 3 reveals that WIND substantially outperforms SOTA text watermarking methods in validating the creative essence watermark, primarily because our approach condenses creation-specific features into a verifiable and implicit watermark. Overall, as demonstrated in the main results, WIND maintains robust compatibility with detecting suspicious texts without being constrained by infringing models, whether black-box or white-box models.

4.3 Robustness Study

We evaluate the robustness of WIND against diverse attack methods. To safeguard creation integrity, attacks must avoid substantial disruptions from creative essence. Our attacks (Dugan et al. (2024)), including case swapping (Upper-Lower), common misspellings (Misspelling), number insertions (Number), adding \n\n between sentences (Add Paragraph), and utilization of Grok for sentence rewriting with creation retention (Rewrite), are designed with minimized creation impact. The first four methods use a 30% probability relative to each sample's length. Table 4 reveals that WIND maintains strong performance even under adversarial attacks, confirming its effectiveness in copyright validation for creative writing.

4.4 Ablation Study

We assess each component's impact via an ablation study (Table 5). The study involves five modifications: '$-\mathcal{L}_{con}$', which removes contrastive loss in the encoder; '$-\mathcal{L}_o$', which eliminates regularization penalty; '$-\boldsymbol{C}$', which skips

Table 4: Robustness study. Robustness attack outcomes are marked in pink.

	SP			ROC		
	F1	TPR	FPR	F1	TPR	FPR
Upper-Lower	$95.87_{\downarrow 3.07}$	$97.21_{\downarrow 2.11}$	$3.15_{\uparrow 2.02}$	$96.34_{\downarrow 2.57}$	$96.52_{\downarrow 2.96}$	$4.74_{\uparrow 3.49}$
Misspelling	$96.72_{\downarrow 2.32}$	$98.41_{\downarrow 0.91}$	$1.25_{\uparrow 0.08}$	$96.87_{\downarrow 2.04}$	$96.42_{\downarrow 3.06}$	$5.79_{\uparrow 3.54}$
Number	$97.63_{\downarrow 1.41}$	$97.47_{\downarrow 1.85}$	$2.50_{\uparrow 1.37}$	$98.19_{\downarrow 0.72}$	$98.91_{\downarrow 0.57}$	$2.14_{\uparrow 0.89}$
Rewrite	$94.25_{\downarrow 4.79}$	$96.53_{\downarrow 2.79}$	$3.47_{\uparrow 2.34}$	$95.60_{\downarrow 3.31}$	$94.37_{\downarrow 5.11}$	$2.89_{\uparrow 1.64}$
Add Paragraph	$97.75_{\downarrow 1.29}$	$96.92_{\downarrow 2.40}$	$0.71_{\downarrow 0.42}$	$97.75_{\downarrow 1.16}$	$96.33_{\downarrow 3.15}$	$2.97_{\uparrow 1.72}$
WIND-G	99.04	99.32	1.13	98.91	99.48	1.25

instance delimitation and LLM condensation phases, 'Froze α', where the encoder is frozen; and '$-q_p$', where samples skip instance delimitation mechanism and go straight to the LLM, bypassing encoder's selection of the best inference instance. Our results show that the exclusion of any component leads to a notable decline in model performance. In particular, omitting the delimitation mechanism and directly inputting samples to the LLM (row '$-q_p$') lowers the F1 score by about 16% (from 99.04 to 82.32), highlighting the critical role of the delimitation mechanism in providing high-quality input for effective LLM condensation. Moreover, Table 5 reveals inferior performance in BERT and RoBERTa compared to SimCSE-RoBERTa, attributed to reduced model anisotropy.

Table 5: Ablation study results are highlighted in pink, and various encoders are marked in blue.

	SP			ROC		
	F1	TPR	FPR	F1	TPR	FPR
$-\mathcal{L}_{con}$	$93.61_{\downarrow 5.43}$	$88.02_{\downarrow 11.3}$	$1.54_{\uparrow 0.41}$	$95.82_{\downarrow 3.09}$	$92.05_{\downarrow 7.43}$	$1.97_{\uparrow 0.72}$
$-\mathcal{L}_{o}$	$91.56_{\downarrow 7.48}$	$86.08_{\downarrow 13.24}$	$2.05_{\uparrow 0.92}$	$92.53_{\downarrow 6.38}$	$86.04_{\downarrow 13.44}$	$3.56_{\uparrow 2.31}$
$-\boldsymbol{C}$	$84.49_{\downarrow 14.55}$	$76.03_{\downarrow 23.29}$	$3.97_{\uparrow 2.84}$	$89.12_{\downarrow 9.79}$	$90.09_{\downarrow 9.39}$	$12.02_{\uparrow 10.77}$
Froze α	$86.23_{\downarrow 12.81}$	$86.07_{\downarrow 13.25}$	$14.01_{\uparrow 12.88}$	$86.16_{\downarrow 12.75}$	$82.09_{\downarrow 17.39}$	$18.05_{\uparrow 16.80}$
$-q_p$	$82.32_{\downarrow 16.72}$	$70.08_{\downarrow 29.24}$	$5.97_{\uparrow 4.84}$	$84.73_{\downarrow 14.18}$	$78.09_{\downarrow 21.39}$	$9.98_{\uparrow 8.73}$
BERT (WIND)	$91.27_{\downarrow 7.77}$	$88.76_{\downarrow 10.56}$	$6.35_{\uparrow 5.22}$	$92.13_{\downarrow 6.78}$	$87.51_{\downarrow 11.97}$	$4.79_{\uparrow 3.54}$
RoBERTa (WIND)	$94.94_{\downarrow 4.63}$	$92.79_{\downarrow 6.53}$	$4.54_{\uparrow 3.41}$	$93.67_{\downarrow 5.24}$	$94.37_{\downarrow 5.11}$	$5.93_{\uparrow 4.68}$
WIND-G	99.04	99.32	1.13	98.91	99.48	1.25

4.5 Creative Elements and Prompt Sensitivity

Single creative element. The results appear in Figure 3(a). Preserving only specific elements while omitting others leads to varying degrees of performance degradation, demonstrating that different elements represent core attributes of creative essence. For example, in the ROC dataset (composed of modern works), extracting only rhythm and flow (RF) features significantly reduces style extraction performance. This occurs because RF features, while prominent in poetry (a subset of SP), are not equally distinctive across the entire SP dataset.

Combined elements. To validate the impact of different combinations of creative elements, we test four representative combinations on SP and ROC. The results in Table 6 (marked in pink) demonstrate that adding more elements consistently improves performance, confirming their complementary roles. For example, on SP, the 3-element set (VWC+SSGF+TS) outperforms the 2-element one (SSGF+TS). On ROC, combinations with RF show smaller gains, aligning with conclusion of Figure 3(a). Together, these experiments confirm the efficacy and non-redundancy of the five-element set. Its ability to represent an author's potential multi-genre characteristics allows it to effectively condense the essence of creative writing, achieving comprehensive stylistic coverage where the omission of any component impairs performance.

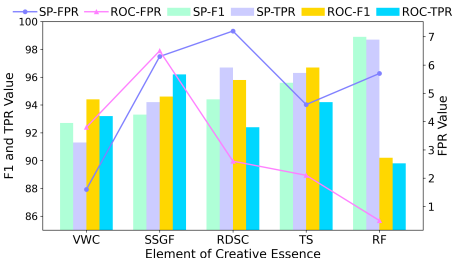
Table 6: Results for different combinations of creative elements and prompt sensitive.

	SP			ROC		
	F1	TPR	FPR	F1	TPR	FPR
SSGF+TS	$95.36_{\downarrow 3.68}$	$94.21_{\downarrow 5.11}$	$3.35_{\uparrow 2.22}$	$95.73_{\downarrow 3.18}$	$94.36_{\downarrow 4.26}$	$2.75_{\uparrow 1.34}$
VWC+RF	$96.46_{\downarrow 2.58}$	$93.92_{\downarrow 5.40}$	$4.99_{\uparrow 3.86}$	$94.07_{\downarrow 4.84}$	$91.75_{\downarrow 6.87}$	$3.26_{\uparrow 1.85}$
VWC+SSGF+TS	$97.05_{\downarrow 1.99}$	$97.81_{\downarrow 1.51}$	$3.81_{\uparrow 2.68}$	$97.58_{\downarrow 1.33}$	$96.27_{\downarrow 2.35}$	$1.06_{\downarrow 0.35}$
RDCS+TS+RF	$97.76_{\downarrow 1.28}$	$98.59_{\downarrow 0.73}$	$3.13_{\uparrow 2.00}$	$95.08_{\downarrow 3.83}$	$93.46_{\downarrow 5.16}$	$3.07_{\uparrow 1.66}$
q_{pmid}	$98.79_{\downarrow 0.25}$	$99.27_{\downarrow 0.05}$	$1.75_{\uparrow 0.62}$	$98.54_{\downarrow 0.37}$	$98.26_{\downarrow 0.36}$	$1.13_{\downarrow 0.28}$
q_{plow}	$98.03_{\downarrow 1.01}$	$98.75_{\downarrow 0.57}$	$2.69_{\uparrow 1.56}$	$97.91_{\downarrow 1.00}$	$98.27_{\downarrow 0.35}$	$2.52_{\uparrow 1.11}$
WIND-G	99.04	99.32	1.13	98.91	98.62	1.41

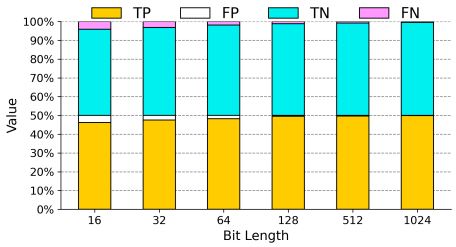
Exploring prompt sensitivity. To assess whether WIND's performance relies on prompt template quality during the LLM condensation phase, we systematically modify the templates (see Appendix A.2 for details). First,

we eliminate the task description, including the role statement and general overview (*"You are an excellent linguist..."*), to produce template q_{pmid}. Next, we remove illustrative examples from the creative elements description, yielding template q_{plow}. The negative sample template q_n remains unchanged, as it already serves as a minimal configuration. Results in Table 6 (blue cells) show that as long as the prompt template retains its core components (task definition, the five creative elements, and output requirements), it effectively guides the large model in disentangling style-specific features. Thus, WIND demonstrates robustness to variations in prompt design.

4.6 Further Explorations



(a) Effectiveness of five elements

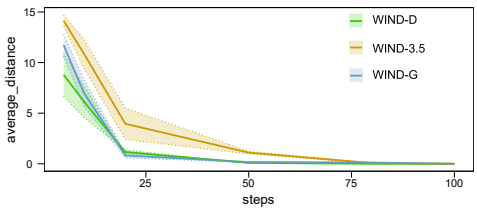


(b) Bit length study when protecting SP.

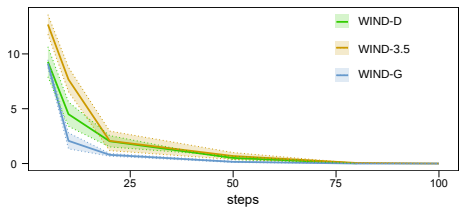
Figure 3: Further exploration. Performance of WIND-G as an illustrative case.

Impact of bit length. We investigate the counts of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) across different watermark lengths, visualized using stacked histograms (see Figure 3(b)). Notably, both FP and FN gradually decrease as the watermark bit length increases. This trend can be attributed to the ability of longer watermarks to encapsulate more distinctive features.

Impact of the regularization penalty. As shown in Figure 4, the average distance converges to zero during training under $\mathcal{L}_o$, demonstrating the regularization penalty's effectiveness in narrowing the protected creative domain. WIND-3.5 maintains a higher region than WIND-D and WIND-G, reflecting GPT-3.5's weaker consistency in disentangling the protected style. The ribbon for WIND-D is slightly wider than that for WIND-G, which also indicates a greater standard deviation when the sample size (num) is 10, consistent with Subsection 4.2.



(a) SP



(b) ROC

Figure 4: The effectiveness of regularization penalty, where area within the dashed line represents the std deviation.

Other explorations. We begin by comparing the performance of WIND against other types of methods, providing subsequent analysis. We then systematically examine the effect of varying sample sizes on WIND's performance, supported by an in-depth case study on condensing style lists by the LLMs. Additionally, we investigate the generalization capability of WIND. For further details, please refer to Appendix D.3.

5 Conclusion and Limitation

In this paper, we introduce WIND, a verifiable and implicit watermarking scheme designed to protect the copyright of creative writing. First, we decompose the abstract concept of creative essence into five elements and propose an instance delimitation mechanism to guide LLMs in generating condensed-lists. These lists are then projected into a disentangled creation space for watermark mapping. Crucially, WIND preserves the integrity of the creative essence while operating independently of the infringement model, ensuring broad compatibility with imitation texts. Experimental results demonstrate WIND's superior performance in both practicality and copyright verification robustness. Currently, WIND's limitations lie primarily in its disentanglement performance; feature work focuses on optimizing the feature space to more effectively represent the core attributes of creative writing. Furthermore, although the set of five creative elements is proven effective, justified, and non-redundant, this fixed set may struggle to accommodate highly niche or unconventional styles. To enhance flexibility, the framework can be extended by incorporating additional style-specific elements to address a broader range of expressions.

6 Ethics Statement

This research adheres to the ICLR Code of Ethics and maintains high standards of research integrity throughout all stages of the study. No human subjects are involved, and all datasets utilized are publicly available and do not contain personally identifiable information. The methodologies and findings are presented transparently, with careful consideration given to potential risks, including misuse, bias, and fairness. We have taken steps to mitigate any unintended harm by rigorously evaluating model outputs for bias and ensuring compliance with data privacy and legal requirements. There are no conflicts of interest or external sponsorships influencing this work.

7 Reproducibility statement

We are committed to ensuring the reproducibility of our results. Core experimental code is provided in the supplementary materials. Detailed descriptions of experimental settings and procedures can be found in Section 4 of the main paper, with additional implementation and experimental details presented in Appendix D. These resources collectively enable independent verification and reproduction of our findings.

References

Authors guild v.openai inc.(1:23-cv-08292)[db/ol].(2023-09-19)[2024-04-20]. 2023. URL https://www.courtlistener.com/docket/67810584/authors-guild-v-openai-inc/.

Getty images vs. stability ai: A landmark case in copyright and ai, 2023. 2023. URL https://www.bakerlaw.com/getty-images-v-stability-ai/.

Sarah silverman and authors sue openai and meta over copyright infringement. 2023. URL https://www.nytimes.com/2023/07/10/arts/sarah-silverman-lawsuit-openai-meta.html.

The times sues openai and microsoft over a.i. use of copyrighted work. 2023. URL https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html.

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

Kejiang Chen, Xianhan Zeng, Qichao Ying, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. Invertible image dataset protection. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 01–06. IEEE, 2022.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pp. 1125–1139. PMLR, 2024.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuan-Jing Huang. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5997–6007, 2019.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1107–1128, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.64. URL https://aclanthology.org/2024.emnlp-main.64/.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024, pp. 12463–12492. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.674. URL https://doi.org/10.18653/v1/2024.acl-long.674.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, 2021.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting llms with binoculars: zero-shot detection of machine-generated text. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 17519–17537, 2024.

Xuanli He, Qiongkai Xu, Lingjuan Lyu, Fangzhao Wu, and Chenguang Wang. Protecting intellectual property of language generation apis with lexical watermark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 10758–10766, 2022.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Baixiang Huang, Canyu Chen, and Kai Shu. Can large language models identify authorship? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 445–460, 2024a.

Junqiang Huang, Zhaojun Guo, Ge Luo, Zhenxing Qian, Sheng Li, and Xinpeng Zhang. Disentangled style domain for implicit z-watermark towards copyright protection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b.

Justine Kao and Dan Jurafsky. A computational analysis of style, affect, and imagery in contemporary poetry. In *Proceedings of the NAACL-HLT 2012 workshop on computational linguistics for literature*, pp. 8–17, 2012.

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

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2023.

Alina Leidinger, Robert van Rooij, and Ekaterina Shutova. The language of prompting: What linguistic properties make a prompt successful? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9210–9232, 2023.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023a.

Qingyi Liu, Jinghui Qin, Wenxuan Ye, Hao Mou, Yuxuan He, and Keze Wang. Adaptive prompt routing for arbitrary text style transfer with pre-trained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18689–18697, 2024b.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019.

Yixin Liu, Hongsheng Hu, Xuyun Zhang, and Lichao Sun. Watermarking text data on large language models for dataset copyright protection. *arXiv preprint arXiv:2305.13257*, 2023b.

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Xiaofei Lu. Automatic analysis of syntactic complexity in second language writing. *International journal of corpus linguistics*, 15(4):474–496, 2010.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. An entropy-based text watermarking detection method. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11724–11735, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.630. URL https://aclanthology.org/2024.acl-long.630.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. Llm dataset inference: Did you train on my dataset? *arXiv preprint arXiv:2406.06443*, 2024.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 24950–24962, 2023.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. MarkLLM: An open-source toolkit for LLM watermarking. In Delia Irazu Hernandez Farias, Tom Hope, and Manling Li (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 61–71, Miami, Florida, USA, November 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.emnlp-demo.7.

James W Pennebaker and Laura A King. Linguistic styles: language use as an individual difference. *Journal of personality and social psychology*, 77(6):1296, 1999.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Mądry. Raising the cost of malicious ai-powered image editing. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 29894–29918, 2023.

Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by {Text-to-Image} models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 2187–2204, 2023.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *12th International Conference on Learning Representations, ICLR 2024*, 2024.

Gerard J Steen, Aletta G Dorst, Tina Krennmayr, Anna A Kaal, and J Berenike Herrmann. A method for linguistic metaphor identification. 2010.

Ruixiang Tang, Qizhang Feng, Ninghao Liu, Fan Yang, and Xia Hu. Did you train on my dataset? towards public dataset protection with cleanlabel backdoor watermarking. *ACM SIGKDD Explorations Newsletter*, 25(1):43–53, 2023.

Fiona J Tweedie and R Harald Baayen. How variable may a constant be? measures of lexical richness in perspective. *Computers and the Humanities*, 32:323–352, 1998.

Maryam Vaezi and Saeed Rezaei. Development of a rubric for evaluating creative writing: a multi-phase research. *New Writing*, 16(3):303–317, 2019.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. In *The Twelfth International Conference on Learning Representations*, 2024.

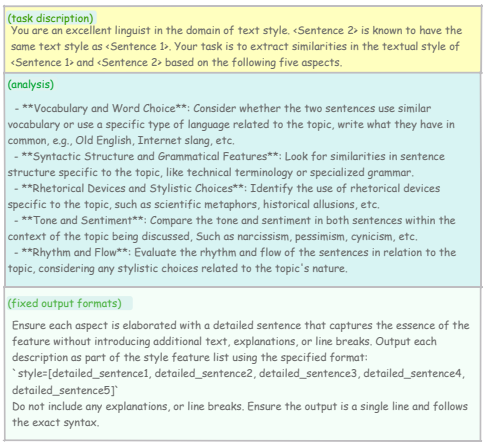
Xuekai Zhu, Jian Guan, Minlie Huang, and Juan Liu. Storytrans: Non-parallel story author-style transfer with discourse representations and content enhancing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14803–14819, 2023.

A Prompt Templates

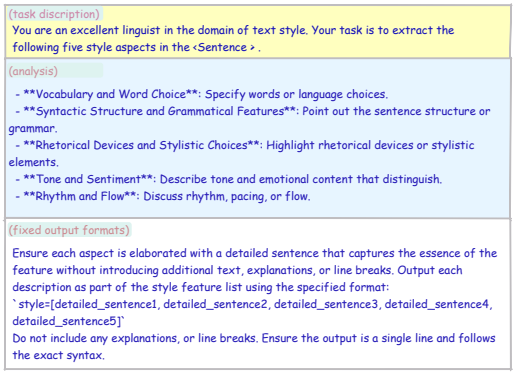
A.1 Five Elements of Creative Essence

- **Vocabulary and Word Choice (VWC).** The type of language used, such as Old English or Internet slang.
- **Syntactic Structure and Grammatical Features (SSGF).** The specific structure of the language, such as technical terminology and specialized grammar.
- **Rhetorical Devices and Stylistic Choices (RDCS).** The use of rhetorical devices, like scientific metaphors or historical allusions, that are particular to the topic.
- **Tone and Sentiment (TS).** The emotional context of the topic, such as narcissism, pessimism, and cynicism.
- **Rhythm and Flow (RF).** The rhythm and flow of sentences, considering stylistic choices based on the topic's nature.

A.2 Construction of Prompts



(a) Details of q_p



(b) Details of q_n

Figure 5: Prompt construction details.

B Mathematical Rationale for Disentanglement via Instance-Aware LLM Conditioning

This section provides a formal justification for WIND's disentanglement mechanism, focusing on how the framework isolates creative-style features through instance-aware conditioning of LLMs.

Problem Setup. Let $\boldsymbol{T_P}$ represent a text sequence containing both creative-specific features $\boldsymbol{cs}$ and creative-irrelevant features $\boldsymbol{ci}$ (e.g., semantic meaning, narrative content). The disentanglement objective is to learn a mapping $\mathcal{F}$ that approximates:

$$\mathcal{F}(t) \approx \boldsymbol{cs} = (e_1, e_2, e_3, e_4, e_5), \tag{12}$$

where each e_i corresponds to one of the five creative elements defined in Section 3.1.

Instance-Aware Disentanglement Mechanism. The disentanglement process operates through optimal context selection. For each test sample t_{test}, we identify its most similar vector in the learned feature space:

$$\boldsymbol{y^*_{test}} = \arg \max_{t \in \boldsymbol{P} \cup \boldsymbol{N}} sim(E_\alpha(t_{test}), E_\alpha(t)). \tag{13}$$

Let $\boldsymbol{y}^*_{\boldsymbol{test}}$ be the feature vector of the text t_{sim}. Then, if $t_{test} \in \boldsymbol{T_P}$, the condition $sim^*_{t_{test}} > \sigma$ ensures with high probability that $t_{sim} \in \boldsymbol{T_P}$. This process can be interpreted through Bayesian inference: the LLM generates the condensed list $\boldsymbol{c}_{\text{test}}$ given the test text and the context from the prompt and demonstration example.

$$\mathcal{P}(\boldsymbol{c_{test}} \mid t_{test}, q_p, t_{sim}) \propto \mathcal{P}(t_{test} \mid \boldsymbol{c_{test}}) \cdot \mathcal{P}(\boldsymbol{c_{test}} \mid q_p, t_{sim}, t_{sim} \in \boldsymbol{T_P}). \tag{14}$$

Here, the likelihood $\mathcal{P}(t_{test} \mid \boldsymbol{c_{test}})$ reflects the LLM's knowledge of how the abstract features in $\boldsymbol{c_{test}}$ manifest in text. The prior $\mathcal{P}(\boldsymbol{c_{test}} \mid q_p, t_{sim} \in \boldsymbol{T_P})$ is constrained by the prompt template q_p and the demonstration example t_{sim} (where $t_{sim} \in \boldsymbol{T_P}$), effectively filtering out content-specific variations and guiding the extraction of style-relevant features. For samples where $t_{test} \in \boldsymbol{T_N}$ or $sim^*_{test} \leq \sigma$, the corresponding Bayesian formulation is:

$$\mathcal{P}(\boldsymbol{c_{test}}|t_{test}, q_n, t_{sim}) \propto \mathcal{P}(t_{test}|\boldsymbol{c_{test}}) \cdot \mathcal{P}(\boldsymbol{c_{test}}|q_n, t_{sim}), \tag{15}$$

where the prior $\mathcal{P}(\boldsymbol{c_{test}}|q_n, t_{sim})$ guides the LLM to extract features divergent from the protected creative essence.

Convergence to Disentangled Representations. The regularization term $\mathcal{L}_o$ (Equation 7) ensures consistent mapping of style features. This loss function drives all style-similar samples toward a compact cluster, effectively disentangling style features from other document characteristics.

Conclusion. WIND's mathematical foundation demonstrates how instance-aware conditioning enables precise separation of style features from content features. The framework achieves disentanglement through: (1) optimal context selection via similarity measures in a learned feature space, (2) Bayesian inference with informative priors provided by demonstration examples, and (3) regularization that enforces consistency in the disentangled representation space.

C Watermark Validation Algorithm

Algorithm 2: Watermark Validation Procedure

Input: Suspicious text t_{test}, encoder $E_\alpha(\cdot)$, LLM $G(\cdot)$, watermark matrix $\boldsymbol{M}_\gamma$, sigmoid $\theta(\cdot)$, prompts q_p, q_n, implicitly verifiable watermark $\boldsymbol{a}$.
Output: Similarity score $\mathcal{P}(\boldsymbol{w_{test}}|\boldsymbol{a})$.

Step 1: Identify classification of t_{test} as $\boldsymbol{pp}$ or $\boldsymbol{neg}$ using 4 and 5;
Step 2: Select prompt q based on classification:

$$q \leftarrow \begin{cases} q_p, & \text{if } t_{test} \in \boldsymbol{pp} \\ q_n, & \text{if } t_{test} \in \boldsymbol{neg} \end{cases}$$

Step 3: Generate condensed list:

$$\boldsymbol{c_{test}} = G(q \mid t_{test})$$

Step 4: Extract disentangled style feature:

$$\boldsymbol{w_{test}} = \theta\big(\boldsymbol{M}_\gamma \cdot E_\alpha(\boldsymbol{c_{test}})\big)$$

Step 5: Compute similarity score:

$$\mathcal{P}(\boldsymbol{w_{test}}|\boldsymbol{a}) = \frac{1}{len} \sum_{i=1}^{len} \mathbb{I}\big(r(w^i_{test}) = r(a^i)\big)$$

Return $\mathcal{P}(\boldsymbol{w_{test}}|\boldsymbol{a})$

D Experiments Appendix

D.1 Details of Datasets

Statistical details of the datasets are summarized in Table 7. For instance, when the protected creative writing is 'ROC', the protected set $\boldsymbol{T_P}$ comprises machine-generated texts where LLMs (i.e., Grok, GPT3.5, and OPT) transform human-written SP texts into ROC outputs. The same applies when protecting 'SP'. Texts in IMDB datasets are sentiment-transformed (a variant of style transfer) by LLMs. In the training process, we randomly sample num instances from $\boldsymbol{T_H}$ and creative works to construct $\boldsymbol{T_P}$ and $\boldsymbol{T_N}$ respectively, following the same process for validation. Importantly, the datasets for training, validation, and testing are strictly non-overlapping.

D.2 Implementation Details

Our model is deployed on a Mac OS Sonoma platform equipped with an Apple M1 Pro chip. This system utilizes an integrated GPU rather than a discrete one. While efficient for its intended use, the integrated architecture does not expose explicit GPU-level metrics such as memory usage or processing time. As a result, it is not feasible to collect GPU-specific statistics during training. Instead, we report the total wall-clock time as a proxy for performance. For

Table 7: Statistics of the employed dataset.

	T_H	Negative Samples	GPT3.5		Grok		OPT	
			Size	AVG_l	Size	AVG_l	Size	AVG_l
Train	SP	ROC+IMDB	200	58	200	65	200	69
	ROC	SP+IMDB	200	43	200	39	200	32
Test	SP	ROC+IMDB	120	61	120	69	120	58
	ROC	SP+IMDB	120	40	120	42	120	37

example, a representative training run on the SP dataset with 10 samples completed in approximately 23 minutes on the aforementioned hardware. Besides, the hyperparameter mar is 0.5 empirically, and the threshold of the delimitation mechanism is 0.8.

For the baseline watermarking methods, the green list ratio is set to 0.5. The sum of green tokens in the text can be approximated by a normal distribution with a variance δ^2 of 2.0, and the z-score threshold is 4.0. Detailed personalized parameters for these baseline models are provided in MarkLLM (Pan et al. (2024)).

D.3 Further Explorations

To further solidify our claims and ensure a thorough comparison, the performance of WIND is evaluated against two modern classifier types: (1) AIGT detectors, where we evaluate and adapt two representative systems (DetectGPT Mitchell et al. (2023) and Binoculars Hans et al. (2024)) to distinguish protected from unprotected writing styles, and (2) an LLM classifier that receives 10 in-context examples (5 protected, 5 unprotected) and classifies new samples. Table 8 compares these baselines with our method.

Table 8: Baseline results for AIGT detectors, LLM classifier and our WIND.

	SP			ROC		
	F1	TPR	FPR	F1	TPR	FPR
GPT-3.5	$43.58_{\downarrow 55.46}$	$35.72_{\downarrow 63.60}$	$28.22_{\uparrow 27.09}$	$26.70_{\downarrow 72.21}$	$35.82_{\downarrow 62.80}$	$52.18_{\uparrow 50.77}$
DetectGPT	$38.25_{\downarrow 60.79}$	$40.67_{\downarrow 58.65}$	$71.80_{\uparrow 70.67}$	$46.72_{\downarrow 52.19}$	$35.28_{\downarrow 63.34}$	$15.75_{\uparrow 14.34}$
Binoculars	$55.03_{\downarrow 44.01}$	$64.50_{\downarrow 34.82}$	$69.83_{\uparrow 68.70}$	$47.12_{\downarrow 51.79}$	$44.63_{\downarrow 53.99}$	$44.82_{\uparrow 43.41}$
WIND-G	99.04	99.32	1.13	98.91	98.62	1.41

Additionally, we evaluate representative AIGT detectors and modern LLM classifiers on the creative writing style protection task. AIGT detectors underperform WIND by over 40%, as our task requires distinguishing between AI-generated texts in different styles, where all samples are machine-generated. While standard detectors can identify AI-generated content, they struggle to recognize specific stylistic variations. Similarly, LLM classifiers achieve results comparable to AIGT detectors, reinforcing our key argument: generic LLM classification falls short for creative style protection.

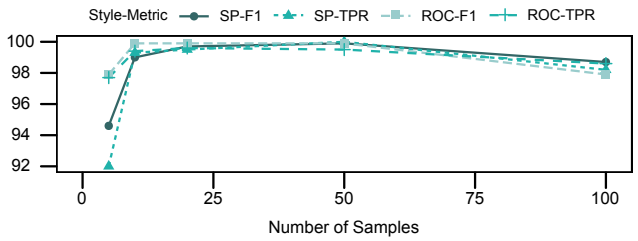


Figure 6: The performance when num changes.

With varying numbers of training samples in the protected creative writing, experimental results in SP and ROC (as shown in Figure 6) reveal that F1 and TPR increase at different rates as num changes. However, the model's performance slightly declines when num approaches 50.

Table 9 summarizes the results of our validation of generalization. The findings demonstrate that WIND's performance remains consistent even when the test data and training data are sourced from different LLMs.

As shown in Figure 7, which presents a case study of the LLM-dominated condensation phase, the strategic design of the instance delimitation mechanism significantly enhances the model's ability to disentangle the creation-specific features.

Table 9: We investigate the generalization of WIND. The notation 'Grok→GPT3.5' indicates that the model is trained on data generated by Grok but tested on data generated by GPT3.5; the same applies to 'Grok→GPT3.5'. This experiment preserves the 'ROC', and WIND-D is trained and tested exclusively on Grok-generated texts.

num		F1	TPR	FPR
6	GPT3.5 → Grok	96.72	96.23	1.67
	Grok → GPT3.5	95.03	94.67	3.83
	WIND-D	96.16	93.67	0.00
10	GPT3.5 → Grok	97.12	96.23	0.24
	Grok → GPT3.5	96.32	95.33	0.33
	WIND-D	96.55	94.00	0.67

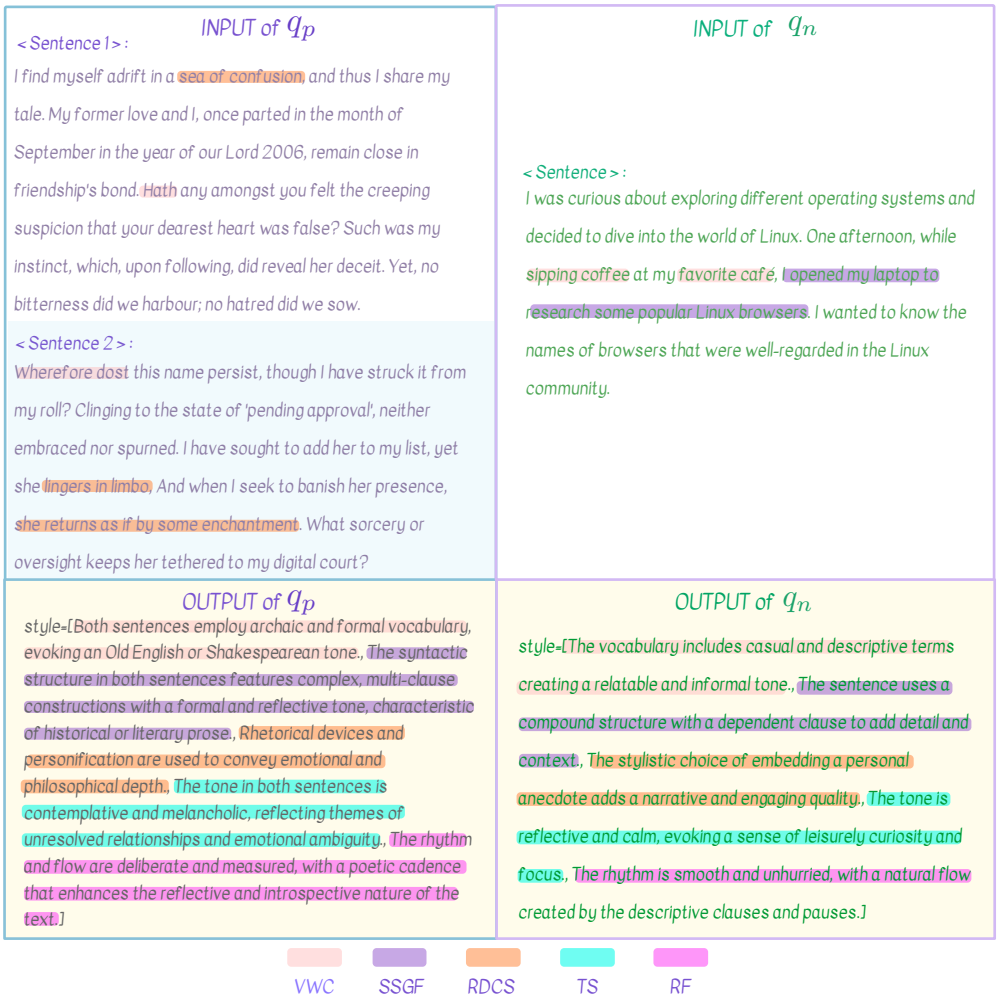


Figure 7: Case study when protecting 'SP'. A sample pair from $\boldsymbol{pp}$ (<Sentence 1> and <Sentence 2>) and a sample from $\boldsymbol{neg}$ (<Sentence>) are combined with the prompt templates q_p and q_n as input. DeepSeek-V3 generates the OUTPUT: five distinct stylistic key points, each highlighted in a unique color.

E THE USE OF LARGE LANGUAGE MODELS (LLMS)

We confirm that all research content and key contributions in this work are entirely original and solely attributable to the listed authors. During the manuscript preparation phase, we employed ChatGPT exclusively for minor language polishing of pre-existing textual content (e.g., grammar correction and sentence structure refinement). No generative AI was used for any substantive research activities, including but not limited to: ideation, technical analysis, data interpretation, or results generation. All authors have thoroughly verified the accuracy of the presented work and assume full responsibility for its scholarly integrity.