

## A IMPLEMENTATION OF MINE

To reduce the computational cost of MINE, we propose a simplified version called sMINE. The main idea is to simplify the structure by fixing the transformer encoder of MINE and reusing the one in the watermarking encoder. In the training loop to calculate mutual information, other parts of the neural network are optimized except for the transformer encoder.

This method significantly accelerates the training process and reduces GPU memory consumption (see Sec. F for details). However, the efficiency gain mildly hurts model's performance. The evaluation results under distortion constraint  $D_1 = 1$  are shown in Fig. 7. Although the difference in capacity and transparency is small, a relatively large decline occurs in robust capacity. It may be because of an inexact estimation of mutual information using the simplified structure. Hence the result again verifies the importance of mutual information payoff in improving model's robustness performance.

## B COMPARISON WITH EXISTING STRATEGIES

To highlight our features, we compare our method against other representative text watermarking strategies under the principle of the information-hiding framework. Comparison baselines include rule-based methods CALS [26] and IF [27], as well as the paraphrasing method AWT [1].

As shown in Tab. 4, 'ours' refers to the proposed canonical framework with all key components in the system, including attacker  $A$ , encoder  $f$ , decoder  $\phi$ ,  $D_{S,X}$ ,  $D_{X,Y}$  constraints, and side information  $K$ . The proposed framework is the only one that is fully in accord with the principle, with others missing one or more key components. Both AWT [1] and CALS [26] have not considered potential attacks in their design, resulting in weaker robustness and vulnerability towards removal attacks. IF [27] considers basic removal attacks, such as insertion, substitution and deletion of words. By contrast, our watermarking system involves an omniscient attacker who plays its attack strategy against known defense in the training loop. By pitting against such an attacker, our watermarking strategy becomes more resistant to removal attacks.

Considering the unique characteristics of natural language, we decompose  $f$  into two parts, one dealing with the position of the embedding and the other selecting candidate words. Although the two parts are jointly handled by transformers in AWT and ours, they are explicitly separated in rule-based methods. CALS uses the synchronicity and substitutability tests to locate target words and then generate corresponding watermarks, whereas IF determines watermark positions either by finding keywords or computing NLI entailment scores. Both of them leverage pre-trained infill models like BERT to embed the watermarks by prediction of candidate words.

With regard to the design of distortion constraints, as all methods are aware of the semantics preservation in texts, they all impose  $D_{S,X}$  constraint. Particularly for CALS and IF, the constraint is implicitly expressed by the candidate selection rules as the infill model mainly considers the semantically similar substitutes. Both AWT and ours adopt the reconstruction loss between the original and watermarked texts as well as the discriminator loss. But our discriminator is stronger at differentiation for an additional condition

$M$ . In addition, we can explicitly control the distortion constraint by  $D_1$ . More importantly, our knowledge distillation loss effectively boosts the likelihood of synonyms, encouraging more diversified and natural reconstructed patterns. For  $D_{X,Y}$ , since no attacker is involved in AWT and CALS, the two methods do not impose the constraint. IF uses the edit distance between the corrupted and the original texts for this constraint, whereas ours adopt the reconstruction loss between  $S$  and  $Y$ , which mediately fulfills the  $D_{X,Y}$  constraint.

Most existing methods ignore the side information  $K$  in their design, but as pointed out by Moulin and O'Sullivan [19],  $K$  plays an important role in the watermarking system. The side information provides a source of randomness that is known to the decoder and enables the use of randomized encoding strategies which could lead to improved transmission performance. Besides, the side information may offer more about the original text to the decoder. If the system does not use side information, the attacker (knowing the encoding strategy used) would be able to decode the watermark and might then be able to remove it from the watermarked text. We consider several types of side information in our system: the original text, a half of the original text, the token-wise difference between  $S$  and  $X$ , as well as the locations of watermarks.

The last row of Tab. 4 points out the fundamental issue in the current design. Security by obscurity means that the method requires to obscure the detail of the security approach to defend attacks, which is widely discouraged in the community. AWT assumes the attacker cannot access the exact models of the encoder and decoder. Hence in facing white-box attacks (Sec.V-D in Abdelnabi and Fritz [1]), the watermark prediction accuracy drops to 50% which is close to random guess. It is even riskier to make the embedding rules public in CALS and IF, as the adversary could easily determine the embedding positions to launch the removal attack. IF explicitly requires a private infill model fine-tuned on potential attacks for robustness. But hiding the infill model, which is a part of  $f$ , obviously violates the principle. In contrast, our watermarking system is designed against an attacker who is fully aware of the watermarking strategies, models, and all data distributions except for the side information and unwatermarked texts. Our algorithm and the trained watermarking models are publicly accessible.

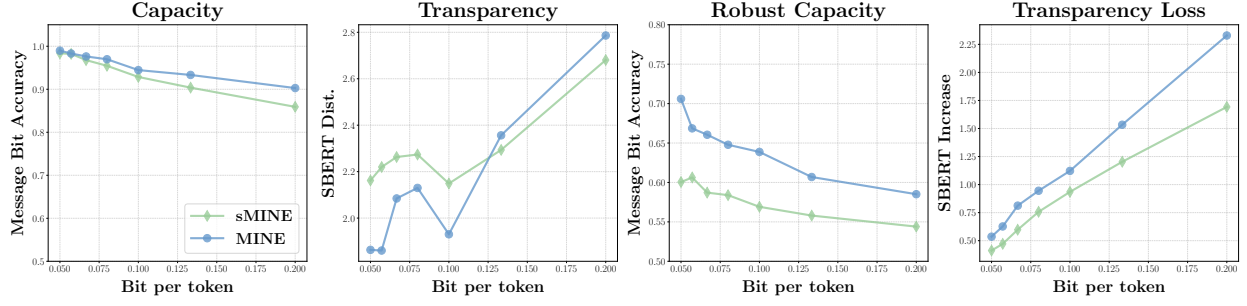
In conclusion, the proposed framework incorporates all key components of the information-hiding framework and avoids the design of security by obscurity, thereby demonstrating itself as a principled approach.

## C CAPABILITY OF REMOVAL ATTACKS

We consider the following adversaries trying to remove watermarks from the watermarked text: denoising auto encoder [DAE; 24], MINE attacker, and adaptive attacker. All attackers are trained on WikiText-2 training set, and are evaluated on WikiText-2 test set. DAE attack has been used by AWT [1] to evaluate robustness, while MINE attack and adaptive attack are introduced by us to specifically target at the watermarking strategy.

We adopt a transformer structure as the backbone of the attacker model, inheriting the DAE setting in AWT, except that we use the pre-trained BERT tokenizer. The detailed implementation of each attack differs by the knowledge and loss functions of the attacker:



Figure 7: Performance comparison of a simplified version of MINE and MINE. (Setup:  $K = \text{None}$ ,  $D_1 = 1$ )

		AWT	CALS	IF	OURS
$f, \phi$	A	None	None	insert/sub./del.	transformer structure
	position candidate	transformer structure	sync.+sub. test infill model	NLI/keyword infill model	transformer structure
	$D_{S,X}$	$L_{\text{rec}}(S, X), L_{\text{disc}}(S, X)$	no need	no need	$L_{\text{rec}}(S, X), \max(0, -L_{\text{disc}}(X, S M) - D_1)$
	$D_{X,Y}$	None	None	edit distance	$L_{\text{rec}}(Y, S)$
	$K$	None	None	None	optional
	Security by obscurity	✓	✓	✓	✗

Table 4: Different watermarking methods examined in the information-hiding framework.

• *DAE* model is trained to denoise noisy input sequences which are created by random embedding dropout and word replacement in the original inputs. The loss function is a reconstruction loss between  $Y$  and the original  $S$ .

• *MINE attack's* payoff is exactly the negative of the watermarking party's payoff in Eq. (4) with the distortion constraint Eq. (10). To further enhance the attacker's strength, we introduce the negative of  $L_{\text{bit}}$  in Eq. (7) into the attacker's loss to interrupt message reconstruction.

• *Adaptive attacker* simply adopts the reconstruction loss between  $Y$  and  $S$  as the training loss (Eq. (13)). Its distortion constraint is implicitly implied by the loss function.

To compare the strength of the three attack methods, we apply them against two watermarking schemes, i.e., AWT and ours, with the results presented in Tab. 5. The Meteor drop and SBERT increase are transparency loss caused by the removal attack, indicating the 'cost' of the attacker, and the bit accuracy (corrupted) refers to the bit accuracy of the watermark extracted from corrupted text under the removal attack. From the perspective of the attacker, the lower these three metrics are, the stronger the attack is. We can see that the adaptive attacker has an exceeding performance compared to other attacks by most metrics. Although MINE attack is the optimal strategy of the attacker in theory, it shares a similar performance to DAE, under which the bit accuracy of the corrupted text is high as above 0.8. This is mostly due to the complexity in controlling the optimization of multiple terms. Adaptive attack is the most effective as it integrates the attacker's payoff and distortion constraints into one reconstruction loss, which is straightforward to optimize in practice. Other attacks such as insertion, deletion, substitution, etc., are omitted here for a high cost in transparency.

Since the adaptive attack exhibits the strongest strength in all the removal attacks, we use it as the in-loop attacker channel, as illustrated in the final paragraph of Sec. 4.2. Also, we adopt the adaptive attack to evaluate the robust capacity of watermarking schemes in the upcoming sections by default.

Table 5: Comparison of the three removal attacks. Arrows indicate the desirable directions. (Ours setup: BPT= 0.05,  $D_1 = 0$ ,  $K=\text{None}$ )

Attacks	Bit Acc (corrupted) ↓		Meteor Drop ↓		SBERT Increase ↓	
	AWT	Ours	AWT	Ours	AWT	Ours
DAE	0.865	0.962	0.052	<b>0.049</b>	1.058	<b>1.190</b>
MINEAtk	0.876	0.975	0.033	0.056	0.606	1.224
AdaptAtk	<b>0.615</b>	<b>0.714</b>	<b>0.026</b>	0.056	<b>0.512</b>	1.227

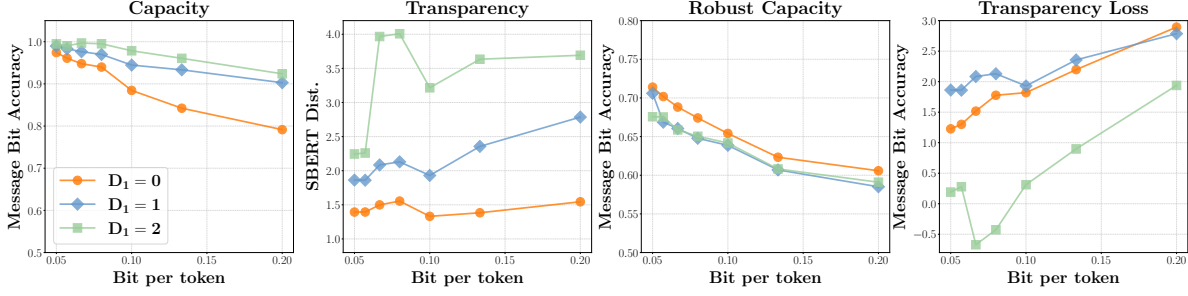
## D ABLATION STUDY

### D.1 Tradeoffs in Capacity, Transparency and Robustness

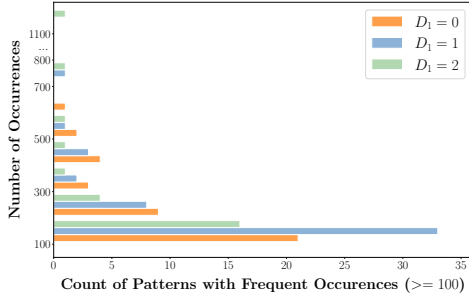
Intuitively, by adjusting the value of constraint  $D_1$  in Eq. (9), we would be able to manipulate model's transparency performance. In this section, we experimentally show how different values of  $D_1$  affect the tradeoffs among capacity, transparency and robustness.

We draw values of  $D_1$  from  $\{0, 1, 2\}$  and illustrate the results in Fig. 8. Clearly, larger values of  $D_1$  result in worse transparency performance and an improved capacity. This is intuitive that the encoder is allowed to make more aggressive changes to the original text given a larger distortion constraint, and these changes help the decoder accurately decode the watermarks. Hence the result





**Figure 8: The capacity, transparency and robustness of our watermarking system under different distortion constraints ( $D_{S,X}$ ). A larger  $D_1$  indicates a looser transparency constraint. (Setup: K=None)**



**Figure 9: Distribution of top-frequent watermark patterns with different distortion constraints. (Setup: BPT= 0.05, K=None)**

indicates that a better capacity can be achieved at the compromise of transparency.

Meanwhile, the interplay between transparency and robustness is more complicated. In Sec. 5.2, two rule-based baselines show a tradeoff between robustness and transparency, while for our framework, the same conclusion does not hold. As  $D_1$  becomes larger, it turns out that the adaptive attacker is more capable of watermark removal, reducing the message bit accuracy to below 0.7 with insignificant transparency losses. As aforementioned, a larger  $D_1$  warrants more aggressive changes and we analyze such changes are also subject to more precise detection by the attacker.

As we examine the occurrences of different watermarking patterns, we find that certain types of patterns may appear more frequently at a larger  $D_1$ . Counts of patterns with number of occurrence more than 100 are summarized in Fig. 9. Here each bin represents the number of patterns with occurrence respectively falling into the region  $[100, 200)$ ,  $[200, 300)$  and so forth. For  $D_1 = 2$ , the most and the second most common patterns appeared over 1100 and 700 times, respectively. With the decrease of  $D_1$ , the distribution of patterns becomes flatter. In particular, no patterns occur more than 700 times at  $D_1 = 0$ . As a result, a larger  $D_1$  makes it easier for the adaptive attacker to learn the mapping from  $X$  to  $S$ . Nevertheless, it does not necessarily indicate a positive correlation between transparency and robustness, since the diversity of watermark patterns may play a part. Compared to AWT, texts generated by our framework contain more diversified watermark patterns

(see Appendix H for details), which can escape the detection of the attacker but suffers transparency compromises.

In conclusion, experimental results verify that  $D_1$  effectively controls model's transparency performance, and serves as a tunable knob in the tradeoff between capacity and transparency. However, the relation between robustness and transparency is much more complicated, and requires closer examination in deployment.

## D.2 Effect of Knowledge Distillation

The module of knowledge distillation plays a pivotal role in enhancing the naturalness of the watermarked text. We observe that the reconstruction loss of Eq. (2) alone is insufficient to ensure naturalness, as it solely controls the distortion from the original text. It means any kind of change made to a single token would lead to the same penalty, regardless of the text semantics. Our knowledge distillation module serves as a complement selecting the most suitable candidate as substitute which brings the minimal change to the text semantics.

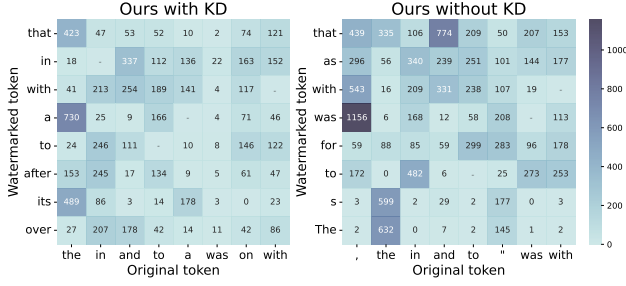
To show its effectiveness, we compare the texts generated by our system with and without knowledge distillation. We set  $D_1 = 1$  to permit changes to the original text. The most frequent watermarking patterns, i.e., how each token is modified to another, are given in Fig. 10. Notably, in the setting where knowledge distillation is absent, punctuation including ‘,’ and ‘”’ are frequently replaced with words like ‘was’ and ‘with.’ Such patterns are unnatural in the sense that they break the grammar rules leading to alteration of text semantics. In the case where knowledge distillation is present, the most common watermarking patterns include ‘the’→‘a’, ‘the’→‘that’, and switch of prepositions. While these patterns look more natural, they follow a flatter distribution than the case without. Without knowledge distillation, a single pattern could be repeated many times, e.g., the single quotation mark to ‘was’ occurs over 1000 times, which is both unnatural and fragile to break. Hence knowledge distillation not only naturalizes the watermarked text, but also smoothens the pattern distributions. More examples of watermarking patterns can be found in Appendix H.

## E SETUP FOR OWNERSHIP VERIFICATION

Steps of watermark embedding of the text owner:

**Step 1.** Split the text into sequences of length  $M$ .





**Figure 10: Most frequent watermark patterns.** The horizontal axis denotes the eight tokens that are replaced most, and the vertical axis indicates the top-8 frequently-chosen substitutes. Value of each cell represents the number of occurrences that the corresponding original token is replaced by the watermarked one. (Setup:  $D_1 = 1$ ,  $K = \text{None}$ )

**Step 2.** Embed messages of length  $L$  into each sequence using the encoding API. In our experiments,  $L = 4$ ,  $M \in \{20, 30, 40, 50, 60, 70, 80\}$ .

**Step 3.** Concatenate all watermarked sequences back into a long text and release it to the public.

Steps of ownership verification:

**Step 4.** Split the text into sequences of length  $M$ .

**Step 5.** Extract messages of length  $L$  from each sequence using the decoding API. If the owner is associated with some side information, the information is fed into the API as well.

**Step 6.** Perform hypothesis test to verify the ownership.

The hypothesis test is a binomial upper-tail test based on null and alternative hypotheses. The null hypothesis  $H_0$  assumes the text is not watermarked by the owner, while the alternative  $H_1$  assumes the text is watermarked by the owner. Assuming  $H_0$  is true, by random guess, number of bits match between extracted messages and ground-truth follows a binomial distribution with probability  $p = 0.5$ . Assume in the extracted message, there are  $k$  bits matching with the ground-truth. Following the common practice of upper-tail tests, we calculate the  $p$ -value, namely the probability of obtaining a statistic that is no less than  $k$ . In the case where  $n$  sequences are aggregated and there are  $nL$  bits of messages in total, we define  $p$ -value as

$$P(x \geq k | H_0 \text{ is True}) = \sum_{i=k}^{nL} \binom{nL}{i} 0.5^{nL}. \quad (14)$$

## F TRAINING COST

In this section, we state the computational overhead of experiments, with specific focus on the time and GPU memory consumption for each task. All experiments were conducted on a machine equipped with several NVIDIA GeForce RTX 3090 graphics cards but only one GPU was demanded for each experiment. The CPU is Intel(R) Xeon(R) Gold 6240C CPU @ 2.60GHz and the memory is 256GB.

The time cost for each training epoch, as well as the GPU memory consumption is shown in Table 6. ‘GPU memory’ is the maximum GPU memory consumption during training. With respect to side

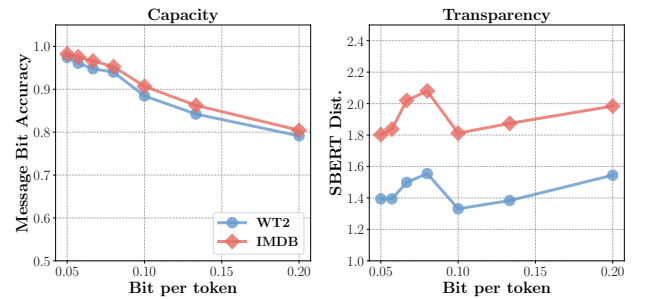
information, we list the case with the highest time cost. We observe that the training of MINE contributes most to the training time except which ours has a similar running time performance to AWT.

**Table 6: Training time and GPU memory consumption.**

Models	Time	GPU memory
AWT	140s	16.60GB
AWT (fine-tuning)	230s	21.92GB
Ours	940s	22.72GB
Ours w.o. MINE	357s	19.47GB
Ours + sMINE	540s	21.23GB
Ours + sMine + K=Loc	704s	22.24GB

## G TRANSFERABILITY TO UNSEEN DATASETS

In this section, we show the transferability of our watermarking system to unseen datasets. Specifically, we evaluate the performance of our trained watermarking system on IMDB [15] dataset. The IMDB dataset contains 50,000 movie reviews (25,000 in training and 25,000 in testing set), with sentiment labels as positive or negative, though the label is not used in our experiments. We use a part of the IMDB testing set as the input text, and evaluate the capacity and transparency of our watermarking system. The results are shown in Fig. 11, with comparison to those on WikiText-2 dataset. We can see that our watermarking system remains good message bit accuracy on the IMDB dataset, while the transparency is slightly shy of that on WikiText-2 dataset. Overall, our watermarking system is transferable to unseen datasets.



**Figure 11: Transferability of our watermarking system to IMDB dataset.** (Setup:  $K = \text{None}$ ,  $D_1 = 0$ )

## H EXAMPLE DEMONSTRATIONS

We provide several examples of our watermarking system in Tab. 7 and Tab. 8. The first column is the input text, the second column is the watermarked text, the third and fourth columns are respectively the SBERT distance and Meteor score, which measure the difference between the original and watermarked texts.

Tab. 7 shows good examples of watermarked text. The watermarked text is semantically similar to the original text. Tab. 8 shows



Table 7: Good examples of the watermarked text. Changed tokens are underlined.

Input	Output	SBERT	Meteor
first to note the breadth of Du Fu’s achievement, writing in [UNK] that his predecessor, " united in his work traits which <u>previous</u> men had displayed only singly ". He mastered all <u>the</u> forms of Chinese poetry : Chou says that in every form he " either made outstanding advances or contributed outstanding examples ". Furthermore, his poems use a wide range of registers, from <u>the</u> direct and [UNK]	first to note the breadth of Du Fu’s achievement, writing in [UNK] that his predecessor, " united in his work traits which <u>earlier</u> men had displayed only singly ". He mastered all <u>his</u> forms of Chinese poetry : Chou says that in every form he " either made outstanding advances or contributed outstanding examples ". Furthermore, his poems use a wide range of registers, from <u>his</u> direct and [UNK]	0.77	0.96
##s comments that, " it is amazing that Tu Fu is able to use so immensely stylized a form <u>in</u> so natural a manner ". [SEP] [SEP] = = Influence = = [SEP] [SEP] According to the [UNK] Britannica, Du <u>Fu</u> ’s writings are considered by many literary critics to be <u>among</u> the greatest of all time, <u>and</u> it states " his dense, compressed language makes use <u>of</u>	##s comments that, " it is amazing that Tu Fu is able to use so immensely stylized a form <u>with</u> so natural a manner ". [SEP] [SEP] = = Influence = = [SEP] [SEP] According to the [UNK] Britannica, Du <u>Fu with a</u> writings are considered by many literary critics to be <u>with</u> the greatest of all time, <u>with</u> it states " his dense, compressed language makes use <u>with</u>	1.36	0.94

Table 8: Bad examples of the watermarked text. Changed tokens are underlined.

Input	Output	SBERT	Meteor
to the [UNK] and self - consciously literary. This variety is [UNK] even within individual works : Owen identifies the, " rapid stylistic and thematic shifts " <u>in</u> poems which enable the poet to represent different [UNK] of <u>a</u> situation, while Chou uses the term " juxtaposition " as the major analytical tool <u>in</u> her work. Du Fu is noted for having written more on [UNK] and painting <u>than</u>	to the [UNK] and self - consciously literary. This variety is [UNK] even within individual works : Owen identifies the, " rapid stylistic and thematic shifts " <u>to</u> poems which enable the poet to depict different [UNK] of <u>to</u> situation, while Chou uses the term " kuxtaposition " as the major analytical tool <u>of</u> her work. Du Fu is noted for having written more on [UNK] and painting <u>to</u>	3.32	0.93

a bad example of watermarked text. In both two tables, there are patterns like ‘previous’→‘earlier,’ ‘represent’→‘depict,’ which would be close to the transparency performance of synonym substitutions under human evaluation. However, in Tab. 8, the watermarked text

contains some grammatical errors like “different [UNK] of to situation.” It is a common problem in learning-based watermarking methods. Although an adversarial discriminator and knowledge distillation have been applied to alleviate this problem, it is still hard to completely eliminate these issues.