symmetry

MDPI

Article

An Effective Framework for Intellectual Property Protection of NLG Models

Mingjie Li *, Zichi Wang and Xinpeng Zhang

School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China; wangzichi@shu.edu.cn (Z.W.); xzhang@shu.edu.cn (X.Z.)
* Correspondence: mingjie8699@126.com

Abstract: Natural language generation (NLG) models combined with increasingly mature and powerful deep learning techniques have been widely used in recent years. Deployed NLG models in practical applications may be stolen or used illegally, and watermarking has become an important tool to protect the Intellectual Property (IP) of these deep models. Watermarking technique designs algorithms to embed watermark information and extracts watermark information for IP identification of NLG models can be seen as a symmetric signal processing problem. In terms of IP protection of NLG models, however, the existing watermarking approaches cannot provide reliable and timely model protection and prevent illegal users from utilizing the original performance of the stolen models. In addition, the quality of watermarked text sequences generated by some watermarking approaches is not high. In view of these, this paper proposes two embedding schemes to the hidden memory state of the RNN to protect the IP of NLG models for different tasks. Besides, we add a language model loss to the model decoder to improve the grammatical correctness of the output text sequences. During the experiments, it is proved that our approach does not compromise the performance of the original NLG models on the corresponding datasets and outputs high-quality text sequences, while forged secret keys will generate unusable NLG models, thus defeating the purpose of model infringement. Besides, we also conduct sufficient experiments to prove that the proposed model has strong robustness under different attacks.

Keywords: natural language generation; ownership protection; RNN; deep neural networks

Citation: Li, M.; Wang, Z.; Zhang, X. An Effective Framework for Intellectual Property Protection of NLG Models. *Symmetry* **2023**, *15*, 1287. https://doi.org/10.3390/sym15061287

Academic Editors: Deming Lei and Alexander Zaslavski

Received: 23 April 2023
Revised: 3 June 2023
Accepted: 16 June 2023
Published: 20 June 2023

1. Introduction

In recent years, many artificial intelligence (AI) domains such as natural language processing [1–3] and computer vision [4–6] have been combined with deep neural networks (DNN), and the corresponding model performance has been significantly improved. Since the generation of DNN model requires massive computational and data resources [7], these well-trained models are regarded significant intellectual property (IP) for model owners. Therefore, it is necessary to protect IP of the DNN models from infringement in order to maintain the competitive advantage of the owners in the open market.

DNN model extraction attacks [8–10] have been proposed in the past few years. These attacks are generally implemented by learning the outputs of these victim DNN models, and the purposes are to imitate the function of the victim DNN models. The DNN model extraction attacks have confirmed that the functionality of DNN models can be stolen through exploiting well-crafted queries [10]. After the DNN models are stolen, unauthorized users can use the extracted DNN models to avoid service charges and ultimately lead to IP infringement. Therefore, it is a challenge to prevent DNN model extraction attacks while maintaining the performance of DNNs models for authorized users.

In view of this, several IP protection approaches of DNN models [11–14] have been proposed, and IP protection of DNN models has become an important research area in recent years. Ideally, the IP protection approaches of DNN models do not degrade the

performance of the original models as much as possible, and must also be able to withstand ambiguity and removal attacks. Digital watermarking is a symmetric signal processing technique that embeds watermark information into the original carrier through a designed algorithm without affecting its normal use, and can extract the embedded watermark information through an extraction algorithm to achieve carrier copyright protection. Watermarking technique is the most common approach to protect IP of DNN models. Recent watermarking approaches [11–13] usually utilize trigger sets to embed invisible watermarks on DNN models before distributing them to authorized users. When a DNN model is suspected of being stolen, the model owner can perform IP claim with the help of the trigger set.

Current watermarking approaches for protecting DNN models either follow traditional digital watermarking framework or focus on protecting the IP of DNN models that perform classification tasks [15]. Unfortunately, these watermarking approaches are not applicable for natural language generation (NLG) models that including machine translation (MT), image captioning (IC), text summarization (TS) and other technologies because text data is quite different from other data. For example, watermarks for computer vision tasks are generally designed images, which of course are not applicable for textual data for general NLG tasks, e.g., MT takes text as input and generates coherent text as output. In addition, DNN models output labels for classification tasks, whereas general NLG models output sequences. Moreover, DNN models are about finding decision boundaries based on the data content for classification tasks, while the NLG models need to fully understand the data content and connect it with the language models to generate coherent text [15].

Although NLG model backdoors [16–18] can be used as watermarks for model IP protection, these model backdoors generally utilize input data that is different from the corpus data as triggers, therefore, they are usually easy to detect and cause malicious behavior of NLG models, which is not conducive to the corresponding application [7]. Recent works on IP protection of NLG models [7,19] using watermarking techniques have been proposed. T. Xiang et al. [7] propose a semantic and robust watermarking approach for NLG models that make use of unharmful phrase pairs as watermarks for ownership protection of NLG models. Recent work [19] presents a novel watermarking approach which embeds watermarks by performing lexical modification to the original outputs of the NLG models. X. He et al. [20] propose a conditional watermarking framework to protecting the IP of NLG models. This watermarking approach reduces the distortion of the overall text word distribution, while changing the selection of conditional words as much as possible. However, these model IP protection approaches can only identify and verify the ownership of the corresponding NLG models, the owner may have to depend on government investigations and enforcement actions to prevent IP infringement. It remains questionable whether these approaches provide reliable and timely protection of models.

Considering the drawbacks of current IP protection approaches of NLG model, we propose two embeding schemes to the hidden memory state of the RNN to protect the IP of NLG models for different tasks. The experimental results demonstrate that our proposed approach hardly degrades the performance of the original NLG model on the corresponding dataset, while forged secret keys will immediately generate unusable NLG models. That is, the NLG models protected by our proposed approach will not work unless the valid secret keys are provided, thus immediately preventing the illegal use of the models at no additional costs and defeating the purpose of model infringement. In addition, we also conduct relevant experiments to prove that the proposed model has strong robustness under different attacks. Main contributions of this paper:

(1) In view of the existing watermarking approaches can not provide reliable and timely model protection for NLG model, we propose two schemes to protect the IP of NLG model for different tasks.

(2) We have conducted sufficient experiments to prove that the proposed approach hardly degrades the performance of the original NLG model while forged secret keys will immediately generate unusable NLG models.

(3) The experimental results show that the proposed approach can not only prove the ownership of the protected NLG model, but also has strong robustness.

The rest of this paper is structured as follows: we provide the preliminary and related work on NLG model IP protection in Section 2. Then, Section 3 elaborates the proposed NLG model IP protection approach in detail. After that, we conduct sufficient experiments and evaluate the proposed approach based on the analysis of the experimental results in Section 4. In Section 5, we summarize our proposed NLG model IP protection approach and the future work is prospected.

2. Preliminary and Related Work

In this section, we describe the preliminary and progress of related work of IP protection approaches of DNN models in recent years, so as to introduce our proposed IP protection approach.

2.1. Digital Watermarking

Digital watermarking is a branch of information hiding field [21], which uses a designed algorithm to embed identification digital information in the carrier without affecting the use of the original carrier to achieve the purpose of protecting the ownership of the digital media, and the embedded digital information is as undetectable and unmodified as possible by attackers [22]. And even with unauthorized copies or minor modifications to the watermarked file, the owners can still extract the watermarks to prove their ownership. Digital watermarking is one of the most suitable technologies to protect IP, verify ownership, track contents of the digital media and ensure authentication and security [23].

Digital watermarking can be roughly divided into video watermarking, image watermarking, audio watermarking and text watermarking according to the watermarking carrier. Because there is a lot of redundant information in the video file, the video watermarking generally embeds the watermarking sequences in it to achieve the purpose of copyright protection. Image watermarking generally embeds watermarks in the time domain or frequency domain of the image by changing the pixel or texture information of the image to protect image ownership. Audio watermarking generally embeds the watermarks by modifying the significant bits in the time domain or frequency domain of the sound. Text watermarking is generally achieved by adjusting the document structure and the semantic structure of the text [21]. Digital watermarking technology is being updated, and mature digital watermarking will be migrated to new carriers in the future.

2.2. IP Protection of DNN Models

Early work on IP protection of DNN models is proposed by embedding watermarks into the weight parameters through parameter regularizers as white-box protection during training [11]. In order to verify the ownership, the owner needs to access the model parameters to extract the embedded watermarks. Subsequent work [12,24–26] uses the black-box as the setting to remedy this issue. A set of trigger set images are produced in the form of random image and label pairs in this setting. In the training process, the feature distributions of these images are far away from the labeled samples. During the verification process, the watermarks can be extracted remotely without accessing the model weights. Recently proposed IP protection approaches of DNN models are applicable to both white-box and black-box settings [13,27]. Rouhani et al. [27] embed watermarks in the designated DNN layers by integrating two regularization loss terms, namely Gaussian Mixture Model agent loss and commonly used binary cross-entropy loss. This approach is robust against attacks such as fine-tuning, but does require more computation. The closest to our work is L. Fan et al. [13] add a "passport" layer to the protected DNN models to enable the DNN models IP validation. If an attacker utilizes a forged passport, the performance of the proposed model will greatly decrease. This approach requires the DNN model owner to maintain the confidentiality of the passport layer weights.

Recent work on IP protection of the NLG models has also been proposed. NLG model backdoors [16–18] can be used as watermarks for model IP protection. However, NLG model backdoors are usually easy to detect and cause malicious behavior of NLG models, which is not conducive to the corresponding application [7]. T. Xiang et al. [7] propose a semantic and robust watermarking approach for NLG models that make use of unharmful phrase pairs as watermarks for ownership protection of NLG models. Experimental results demonstate that the watermarks can maintain its verifiability even after multiple model modifications. Recent work [19] presents a novel watermarking approach which embeds watermarks by performing lexical modification to the original outputs of the NLG models. This approach achieves identifiable performance with less semantic losses. X. He et al. [20] propose a conditional watermarking framework to protecting the IP of NLG models, which reduces the distortion of the overall text word distribution, while changing the selection of conditional words as much as possible. However, these model IP protection approaches can only identify and verify the ownership of the corresponding NLG models. It remains questionable whether these approaches provide reliable and timely protection of models.

3. Proposed IP Protection Approach of NLG Models

At present, the NLG models for MT, IC and TS tasks are mainly implemented by sequence-to-sequence (seq2seq) model [28]. The seq2seq model consists of an encoder and a decoder, and its purpose is to model a conditional probability $p(\mathbf{y}|\mathbf{x})$, here $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ represents the source input and $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ represents the output sequence. Specifically, first the encoder of the seq2seq model projects $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ to hidden states $\mathbf{h} = \{h_1, h_2, \ldots, h_n\}$. Afterwards, hidden states $\mathbf{h} = \{h_1, h_2, \ldots, h_n\}$ are converted into a fixed-length vector $\mathbf{c}$. Finally, output sequence $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$ can be decoded one by one from the vector $\mathbf{c}$. It should be pointed out that the seq2seq model for MT and TS tasks generally uses the LSTM encoder because the input is text data, while the CNN encoder is generally used for the IC task because the input is image.

In view of the drawbacks of current NLG model IP protection approaches, we propose two embeding schemes to the hidden memory state of the RNN of the model decoder to protect the IP of NLG models for different tasks.

3.1. IP Protection Approach Generation of NLG Models

The encoder of model based on seq2seq generally uses LSTM for MT and TS tasks. Firstly, the encoder projects the input $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ to hidden state $\mathbf{h} = \{h_1, h_2, \ldots, h_n\}$. Afterwards, the hidden state $\mathbf{h}$ is converted into a fixed-length vector $\mathbf{c}$. For IC tasks, the model based on Seq2Seq generally uses CNN as the encoder. The encoder first extracts features from the input image through CNN, and then converts it into a fixed-length vector $\mathbf{c}$. Next, we propose the scheme $\mathbb{E}(\mathbf{K}, \mathbf{h}', e)$ to embed secret key into the hidden memory state of the RNN of the seq2seq model decoder to protect the IP of NLG models for different tasks.

$$\mathbb{E}(\mathbf{K}, \mathbf{h}', e) = \begin{cases} \mathbf{K} \oplus \mathbf{h}', & \text{if } e = \oplus, \\ \mathbf{K} \otimes \mathbf{h}', & \text{otherwise.} \end{cases} \tag{1}$$

where the embedded secret key $\mathbf{K} = \{k_i\}_{i=1}^{m}$ with m is the size of the RNN hidden state of the decoder, $k_i \in \mathbb{R} : -1 \leq k_i \leq 1$, $\mathbf{h}' = \{h'_1, h'_2, \ldots, h'_n\}$ is the hidden state of the model decoder. The whole embedding process

$$P_{\mathbb{E}}(\mathbf{D}, \mathbf{s}, \mathbb{N}[\cdot], L) = \mathbb{N}[\mathbf{W}, \mathbf{s}] \tag{2}$$

is a RNN learning process that takes $\mathbf{D}$ as input training data, and additionally together with optionally signature $\mathbf{s}$, and outputs the protected model $\mathbb{N}[\mathbf{W}, \mathbf{s}]$ by minimizing a given loss L, $\mathbf{W}$ represents the model weights.

During the experiments, we use string conversion to generate the secret key. However, since the binary vectors of different alphanumeric conversions are generally close, we perform transformation $\mathbb{T}$ on the converted binary vector $\mathbf{B}_1$ to generate the final key $\mathbf{K}$:

$$\mathbb{T}(\mathbf{B}_1, \mathbf{B}_2) = \mathbf{B}_1 \otimes \mathbf{B}_2 = \mathbf{K} \tag{3}$$

where $\mathbf{B}_2$ is the binary vector sampled by the model owner from -1 or 1.

To further strengthen the whole model, we add the sign loss regularization into the model loss function:

$$L_s(\mathbf{h}', \mathbf{S}, \gamma) = \sum_{i=1}^{m} \max(\gamma - h_i' s_i, 0) \tag{4}$$

where $\mathbf{S} = \{s_i\}_{i=1}^{m}$ with $s_i \in \{-1, 1\}$, m is the size of the hidden state of the decoder. γ is a hyperparameter, which is introduced into the sign loss in order to make the hidden state of the decoder have magnitude greater than 0, this is similar to [13]. The main difference between our approach and [13] is that we do not embed the signature in the model weights, but in the hidden state of the decoder. A major reason is that we found the embedding signature in the model weights is generally vulnerable to channel permutation attacks.

In addition, in order to improve the grammatical correctness of the output sequence, we add a language model loss to the model decoder to fine-tune the model during the training process [29]. Specifically, we first trained AWD-LSTM [30] separately on the used training set, which is effective and widely used in recent years [31–33]. After that, we fix the weights of the trained AWD-LSTM model to calculate the likelihood of the output sequence. The sequence with higher likelihood is more syntactically similar to the target sequence in training process. The Figure 1 shows the overview of IP protection approach generation process for different NLG tasks based on seq2seq model.

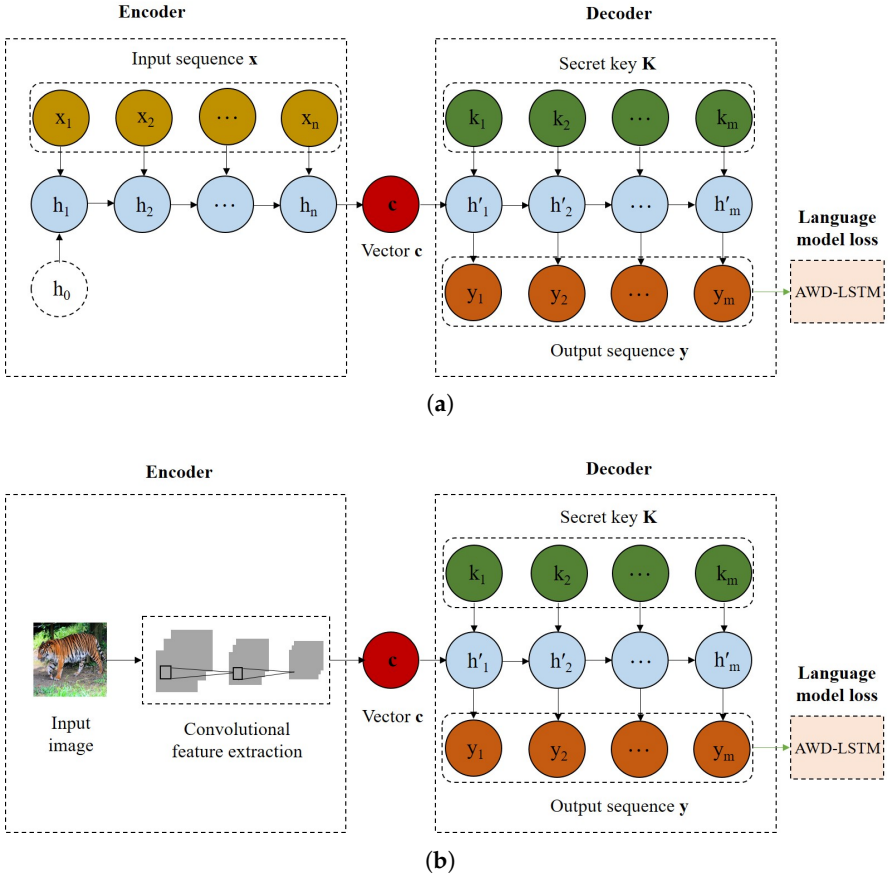


Figure 1. Overview of IP protection approach generation process for different NLG tasks based on seq2seq model. (**a**) IP protection approach generation for MT and TS tasks. (**b**) IP protection approach generation for IC task.

3.2. IP Verification of NLG Models

In this paper, combined with the proposed IP protection approach generation of NLG models, we use two approaches to verify the IP of the NLG models, namely secret key-based IP verification and signature-based IP verification, the overview of the NLG model verification process is shown as Figure 2. In terms of secret key-based IP verification, the user can depend on the secret key is public key or private key to verify the IP of the NLG models. Formerly, trained NLG model and the corresponding public key is provided to the user. In the process of model inference, the public key is needed as part of the input of NLG model to ensure the model performance. In this case, the NLG model IP can be directly verified using the provided key. Latter, a private key is embedded into the NLG model. During model inference, only text data or images are needed as model input. The model owner needs to access to the NLG model and extract the private key from the decoder RNN cells for model IP verification. For the signature-based IP verification, in view of a unique signature is embedded through sign loss regularization during training, the model owner is required to access to the NLG model and extract the sign of the hidden state from the RNN cells of decoder, then compare it with the embedded signature.

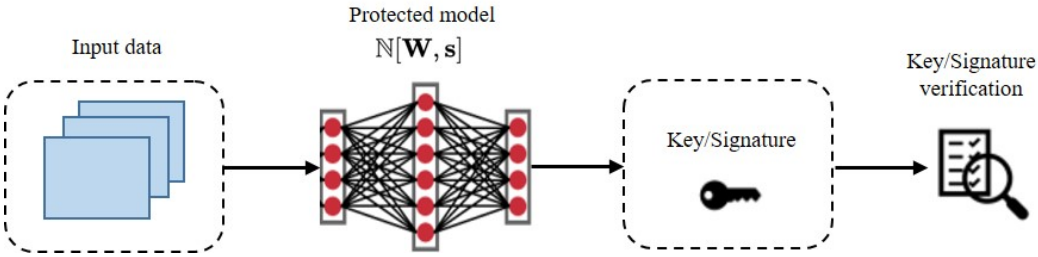


Figure 2. Overview of the NLG model verification process.

4. Experiments

In this section, we conduct relevant experiments to evaluate the proposed IP protection approach of NLG models on the corresponding datasets for different NLG tasks, and compare them with the baseline models.

4.1. Datasets and Models

Datasets. In the experiments, we perform three common NLG models namely MT, IC, and TS on the respective datasets to evaluate the proposed IP protection approach. For the MT task, we perform German to English translation on the the commonly used WMT14 dataset. Moses [34] is employed to pre-process all corpora of the WMT14 dataset and all text contains case. We evaluate the proposed IP protection approach for the IC task on MS-COCO dataset and employ the split provided by [35]. For the TS task, we use the CNN/DM dataset and recycle the dataset version preprocessed by [36]. Following the work [37] that utilizes a small amount of test data to evaluate the performance of the IP protection approach, we randomly select 200 pairs of samples from the test set of different NLG model to form the respective test set. Details of each dataset for different NLG models are listed in Table 1.

Table 1. Statistics of datasets used for different NLG models.

Dataset	Train	Dev	Test	Task
WMT14	4.5 M	3 K	200	Machine Translation
MSCOCO	567 K	25 K	200	Image Captioning
CNN/DM	287 K	13 K	200	Text Summarization

Models. For the MT task, we use the model proposed by [38] as the baseline model. This model adopts the method of introducing attention mechanism into seq2seq model, and achieves good translation effect. It should be noted that the original model in reference [38]

performs English-French translation tasks on WMT14, we only used the model in [38], but we performed English-German translation tasks on WMT14 during the experimental process (including model training and testing). The state-of-the-art work on IC [39] is used as the baseline model for the IC task. This popular model also used the encoder-decoder framework, where a CNN encode an image into a fixed-length vector, and LSTM is used to generate the captions. L. Fan et al. [13] add a "passport" layer to the protected DNN model to enable the DNN model IP validation. It is closer to our proposed approach and will also serve as a baseline model for IC tasks to compare with our approach. For the TS task, we adopt the framework developed by R. Nallapati et al. [40] as the baseline model, which model abstractive TS using attentional RNN Encoder–Decoder and achieve state-of-the-art performance.

4.2. Basic Settings and Evaluation Metrics

Basic Settings. The seq2seq model we used is an RNN Encoder–Decoder for the MT task in the experiments, the encoder and decoder both have 1000 hidden units. During the experiments, we use stochastic gradient descent (SGD) algorithm with Adadelta [41] to train the model, the learning rate is 0.001. The epoch is set to 50 and the minibatch is 80 sentences. For the IC task, we employ ResNet-50 [42] pre-trained on the ImageNet as the encoder. For the decoder, we use LSTM with 512 hidden units. The attention loss factor is set to 0.01. In addition, we optimize the model using Adam [43] with β_1 of 0.9, β_2 of 0.998 and ϵ of 1×10^{-5}. The epoch is set to 20 and the batch size is 32. We use attentional RNN Encoder–Decoder for the TS task. The encoder and decoder both have 400 hidden units. We leverage Adadelta [41] with a learning rate of 0.001 to train the model and set the batch size to 50. We employ NVIDIA TITAN RTX 24GB GPU for accelerating these NLG models training.

Evaluation Metrics. For different NLG tasks, we use BLEU [44] as the metric to evaluate the overall output text sequence quality of the models. BLEU can be used to evaluate the similarity between the reference text sequence and the generated text sequence. In general, a larger BLEU value indicates a higher output text sequence quality of the corresponding model. GErr [45] is the number of increased grammatical errors in the generated text sequence compared to the reference text sequence of the different NLG models. For the different NLG tasks, we hope the GErr to be as small as possible. SPICE [46] is used as an important evaluation metric for model performance for the IC task. It is generally believed that a larger SPICE value corresponds to a higher quality of the output text sequence of the IC model. The ROUGE value [47] is an important performance evaluation metric for the TS model. For the TS model, we hope the ROUGE value to be as large as possible.

4.3. Performance Evaluation for Different NLG Tasks

We conduct experiments on the respective datasets to evaluate the performance of the proposed models for different NLG tasks, and compare them with the corresponding baseline models. Table 2 and Figure 3 show the examples of our proposed model and the baseline models for the MT and IC tasks on the corresponding dataset, respectively. It can be clearly observed from Table 2 and Figure 3 that the text sequences generated by our proposed models are very close to the baseline models, while the image captions generated by "Passport" [13] are quite different from the baseline model, which are very brief compared to the baseline model and our proposed models. This preliminarily shows that the quality of the text sequences generated by our proposed models has a certain guarantee.

Table 2. Examples of different models on WMT14 dataset for the MT task. "$\mathbf{K}_{\oplus}$" and "$\mathbf{K}_{\otimes}$" represent the "$\mathbf{K} \oplus \mathbf{h}'$" and "$\mathbf{K} \otimes \mathbf{h}'$" embedding approaches, respectively (same below).

Source Text	Sie achten auf gute Zusammenarbeit zwischen Pony und Führer und da waren Fenton und Toffee die Besten im Ring.
Baseline [38]	They pay attention to good cooperation between pony and guide, and Fenton and Toffee were the best in the ring.
$\mathbf{K}_{\oplus}$	They pay attention to a good cooperation between pony and guide, and Fenton and Toffee were the best in the ring.
$\mathbf{K}_{\otimes}$	They pay attention to good cooperation between pony and guide and Fenton and Toffee were the best in the ring.

(a) a blue smart car is parked in a parking lot.
(b) a blue smart car parked in a parking lot.
(c) a blue smart car is parked in a parking lot.
(d) a blue smart car is parked.

(a) a dog sitting on a chair in front of a tv.
(b) a dog sitting on a chair in front of a tv.
(c) a dog sitting on a chair in front of a tv.
(d) a dog sitting in front of a tv.

(a) a white bird flying over a body of water.
(b) a white bird is flying over a body of water.
(c) a white bird flying over a body of water.
(d) a bird flying over a body of water.

Figure 3. Captions generated by (**a**) baseline model [39], (**b**) $\mathbf{K}_{\oplus}$, (**c**) $\mathbf{K}_{\otimes}$, (**d**) "Passport" [13] on MS-COCO dataset. It can be observed that the image captions generated by our proposed models are very close to the baseline models, while the image captions generated by "Passport" [13] are quite different from the baseline model.

Performance of our proposed models and the baseline models on WMT14, MSCOCO and CNN/DM datasets for the MT, IC and TS tasks are shown in Tables 3–5, respectively. It can be seen from Tables 3–5 that the overall performance of our proposed models is similar to the baseline models, while in terms of increased grammatical errors(GErr), the proposed models significantly outperform the corresponding baseline models. This is mainly attributed to the trained AWD-LSTM model included in the decoder of our proposed models. The sequence with higher likelihood is more syntactically similar to the target sequence in training process, that is, the GErr of sequence generated by our approach is lower. In addition, it can be observed from Table 4 that the performance of the "Passport" [13] model on MS-COCO is significantly worse than our models and the baseline model for the IC task. For example, the SPICE of the "Passport" [13] model dropp 16.86% while our proposed models drop at most 3.7% on MS-COCO dataset.

Table 3. Performance of different models on WMT14 dataset for the MT task. The baseline is the model proposed by [38].

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	GErr
Baseline	29.82	21.65	14.30	10.95	1.06
$\mathbf{K}_{\oplus}$	28.54	21.51	13.42	11.08	0.86
$\mathbf{K}_{\otimes}$	28.33	20.85	13.04	9.40	0.82

Table 4. Performance of different models on MS-COCO dataset for the IC task. The baseline is the model proposed by [39] and "Passport" is the model proposed by [13].

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	SPICE	GErr
Baseline	71.74	57.30	39.89	30.92	18.51	1.02
$\mathbf{K}_{\oplus}$	70.48	57.42	39.58	28.08	18.30	0.83
$\mathbf{K}_{\otimes}$	70.17	56.91	39.43	29.94	17.85	0.86
Passport [13]	65.31	51.27	37.06	27.37	15.39	1.05

Table 5. Performance of different models on CNN/DM dataset for the TS task. The baseline is the model proposed by [40].

Model	ROUGE-1	ROUGE-2	ROUGE-L	GErr
Baseline	39.08	18.45	35.29	1.14
$\mathbf{K}_{\oplus}$	38.61	18.49	34.87	0.90
$\mathbf{K}_{\otimes}$	38.75	18.06	33.95	0.93

4.4. Human Evaluation for Different NLG Tasks

To further verify the performance of our proposed models and the baseline models on the corresponding datasets for the different tasks, we employ the common platform namely Amazon Mechanical Turk (https://www.mturk.com (accessed on 2 March 2023)) for human evaluation to verify the naturality of the output text sequences generated by our proposed models and the corresponding baseline models. In the experiment, we randomly select 200 output text sequences generated by different models and ask annotators to give corresponding naturality scores to evaluate the naturality of the output text sequences. Specifically, for each output text sequence, we ask annotators to give naturality scores from $\{1, 2, 3\}$, which stand for "machine generated", "uncertain" and "human written" respectively, and use the average of the naturality scores provided by these annotators as the final naturality score. The higher the naturality score, the more natural the output text sequence. The average naturality scores of the output text sequences generated by our proposed models and the corresponding baseline models on corresponding datasets for the different tasks is shown as Table 6. From Table 6 we can clearly see that the output text sequences generated by both our proposed models and the corresponding baseline models get high natural scores, while the text sequences generated by the proposed models generally have higher natural scores than the corresponding baseline models for the different tasks.

Table 6. The average naturality scores of the output text sequences generated by our proposed models and the corresponding baseline models on corresponding datasets for the different tasks.

NLG Tasks	Datasets	Models	Naturality Score
MT	WMT14	Baseline [38]	2.420
		$\mathbf{K}_{\oplus}$	2.435
		$\mathbf{K}_{\otimes}$	2.430
IC	MS-COCO	Baseline [39]	2.345
		$\mathbf{K}_{\oplus}$	2.340
		$\mathbf{K}_{\otimes}$	2.360
TS	CNN/DM	Baseline [40]	2.265
		$\mathbf{K}_{\oplus}$	2.325
		$\mathbf{K}_{\otimes}$	2.305

4.5. Protection against Forged Key and Signature

To further evaluate the performance of our proposed models, we assume that an attacker somehow accesss into the protected NLG models and tries to attack these models with forged keys and signatures. Figure 4 shows the performance of our proposed models for different NLG tasks on the corresponding datasets. It is obvious from Figure 4a,c,e that in general, the performance of NLG models will degrade when forged keys are used. For example, even if the dissimilarity between the real key and the fake key is only 20%, the BLEU-1 of our proposed "$\mathbf{K}_{\oplus}$" model drops by almost 50% for the MT task on WMT14 dataset. Table 7 shows examples of different models on WMT14 dataset for the MT task when our proposed models under forged key attack. As shown in Table 7, the proposed models can generate translated text sequences very similar to the baseline model given the correct key. However, The translation sentences generated by the proposed models using fake keys with 25% and 50% dissimilarity to the real key are either incomplete or very brief and meaningless. This shows that our proposed IP protection approach is effective to against the forged key attack.

In practice, if the key of the proposed model is exposed to the attacker, we can also use the signature embedded in the model as proof of ownership. In this case, the attacker will try to forge the signature to attack the model by changing the sign of the signature. From Figure 4b,d,f, we can observe that the performance of our proposed models degrades to varying degrees when the forged signatures are used. For example, even if the dissimilarity between true and fake keys increases by 10%, the SPICE of our proposed "$\mathbf{K}_{\otimes}$" model drops by almost 13% for the IC task. When the dissimilarity between true and fake keys increases to 60%, the performance of the model has become so poor that it is almost useless. This demonstates our proposed IP protection approach is also resilient against to the forged signature attack.

Table 7. Examples of different models on WMT14 dataset for the MT task. The baseline is the model proposed by [38]. "$\mathbf{K}_{\oplus}$" represents the proposed "$\mathbf{K} \oplus \mathbf{h}'$", "$\mathbf{K}_{\oplus} - 25$" and "$\mathbf{K}_{\oplus} - 50$" represent the proposed "$\mathbf{K}_{\otimes}$" model using the forged key with 25% and 50% dissimilarity to the real key, respectively.

Source Text	Der Renditeabstand zwischen Immobilien und Bundesanleihen sei auf einem historisch hohen Niveau.
Baseline	The return gap between real estate and federal bonds is historically high.
$\mathbf{K}_{\oplus}$	The yield gap between real estate and federal bonds is at a historically high level.
$\mathbf{K}_{\oplus} - 25$	The gap between real estate and federal bonds is at a historically high level.
$\mathbf{K}_{\oplus} - 50$	The gap between real estate and federal bonds.

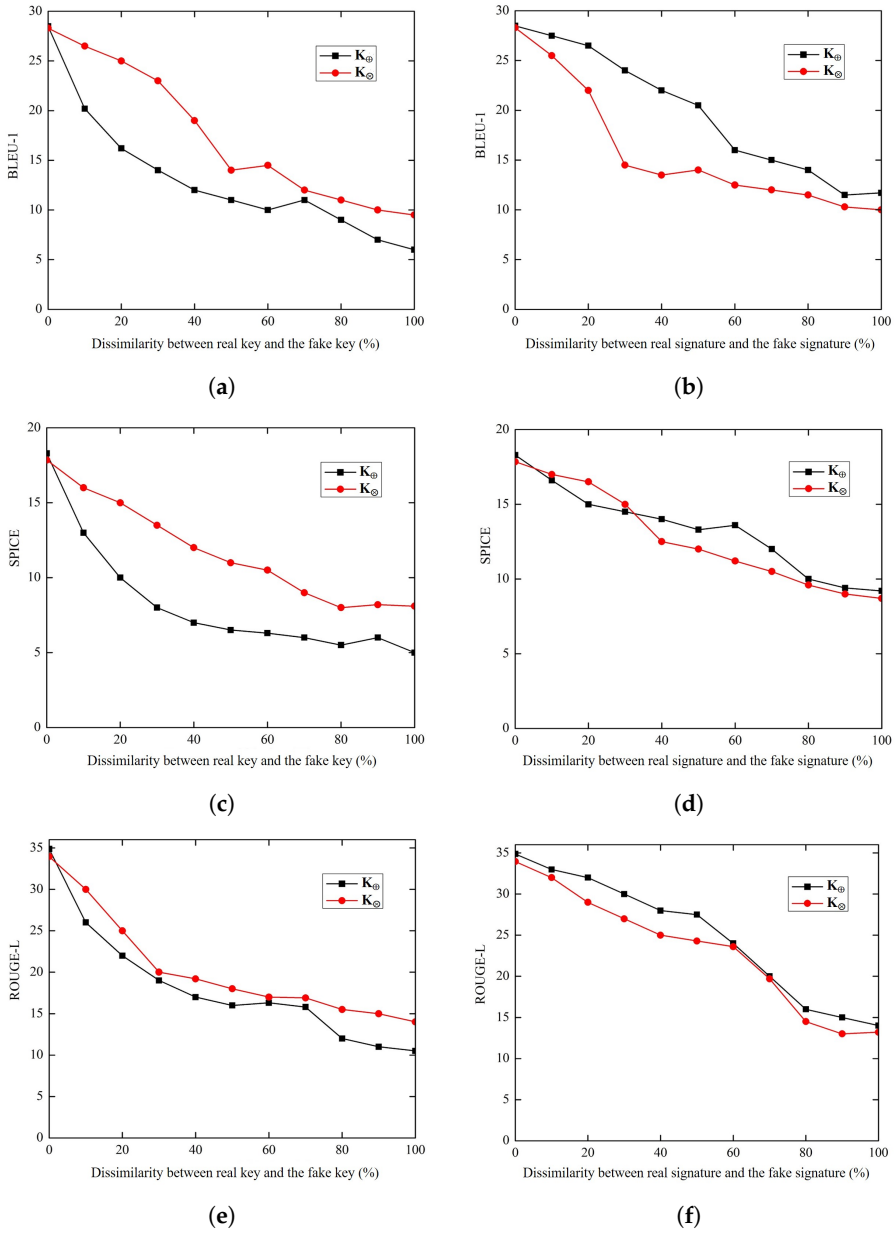


Figure 4. Performance of the proposed models with forged keys and signatures for different NLG tasks compared with the baseline models. (**a**) BLEU-1 scores with forged key on WMT14 dataset for the MT task. (**b**) The BLEU-1 scores with forged signature on WMT14 dataset for the MT task. (**c**) SPICE scores with forged key on MS-COCO dataset for the IC task. (**d**) SPICE scores with forged signature on MS-COCO dataset for the IC task. (**e**) ROUGE-L scores with forged key on CNN/DM dataset for the TS task. (**f**) ROUGE-L scores with forged signature on CNN/DM dataset for the TS task.

4.6. Robustness

The aforementioned experimental results and analysis have demonstrated that our proposed IP protection approach is effective to against the forged key and signature attacks. In practical applications, attackers may try different approaches to perform removal attacks on our proposed models. Next, we conduct experiments to evaluate the robustness of the proposed models for the removal attacks.

4.6.1. Pruning

Model parameter pruning is a commonly used technique in deep learning model training in recent years, mainly to reduce the weights and computation of DNN model.

However, in practical applications, an attacker may exploit it to remove the signatures embedded in the proposed models. In the experiments, we employ class-blind pruning [48] approach to evaluate the robustness of the proposed models for the removal attacks. Figure 5 shows the performance and signature detection rates of the proposed models with different model parameter pruning rates for different NLG tasks. It can be observed from Figure 5 that the proposed model can maintain good performance and high signature detection rate even with a high model parameter pruning rate. For instance, even if the model parameter pruning rate reaches 60%, BLEU-1 of our proposed "$\mathbf{K}_{\otimes}$" model drops by almost 10% and the signature detection rate can reach about 88% for the MT task. Table 8 shows examples of different models on WMT14 dataset for the MT task when our proposed "$\mathbf{K}_{\otimes}$" model with parameter pruning rate of 60%. According to Table 8, we can clearly observe that even if the parameter pruning rate of the proposed "$\mathbf{K}_{\otimes}$" model is 60%, the translated text sequence generated by the model is still similar to the baseline model. In conclusion, the signature detection rates and performance of the proposed models can be guaranteed even in the case of high model parameter pruning rates.

Table 8. Examples of different models on WMT14 dataset for the MT task. The baseline is the model proposed by [38]. "$\mathbf{K}_{\oplus} - 60$" represents the proposed "$\mathbf{K}_{\oplus}$" model with parameter pruning rate of 60%.

Source Text	Die neue Saison in der Falkenberger Discothek "Blue Velvet" hat begonnen.
Baseline	The new season in the Falkenberg discotheque "Blue Velvet" has begun.
$\mathbf{K}_{\oplus}$	The new season in Falkenberg discotheque "Blue Velvet" has begun.
$\mathbf{K}_{\oplus} - 60$	The season in Falkenberg discotheque "Blue Velvet" has begun.

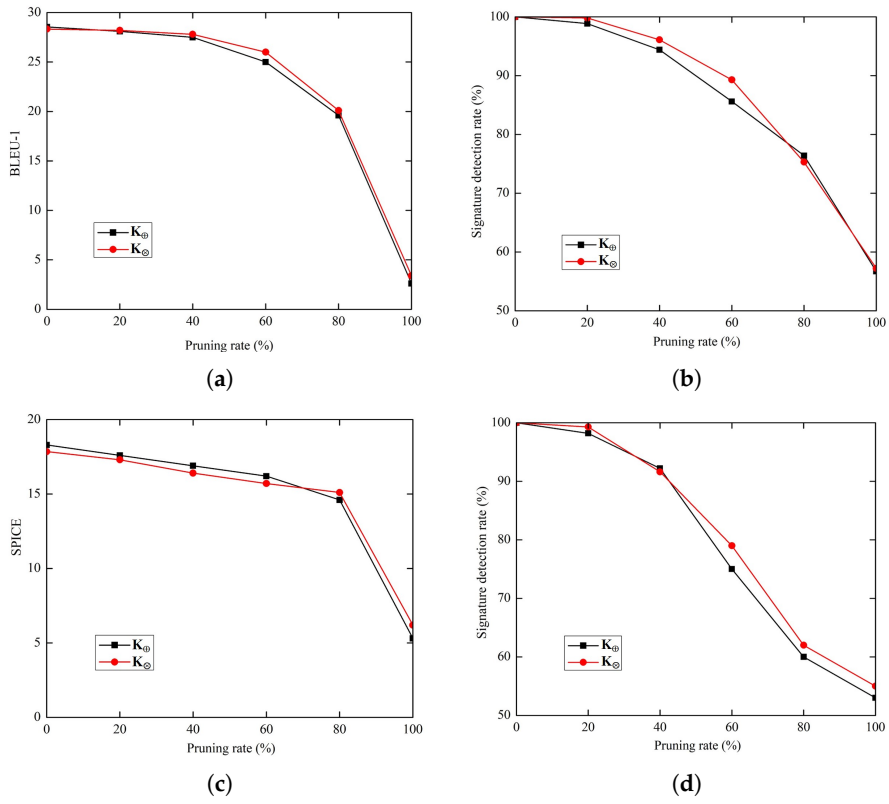
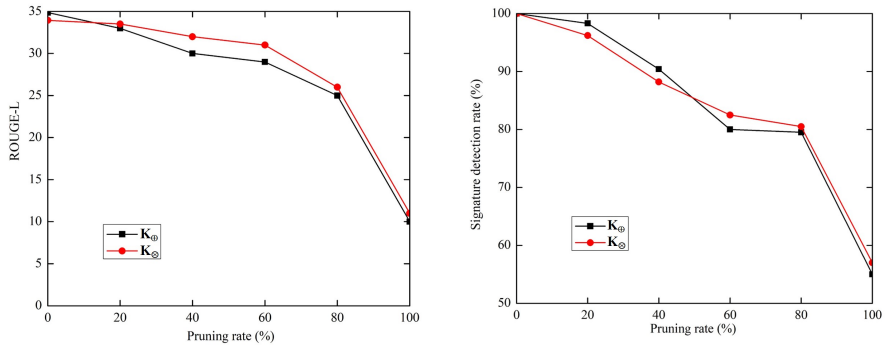


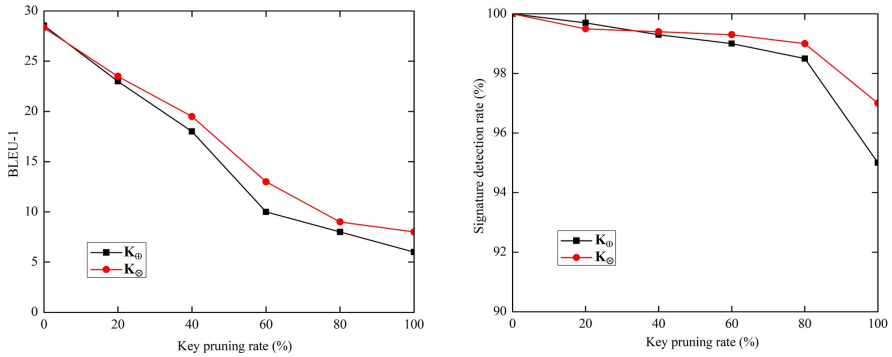
Figure 5. *Cont.*



(**e**) (**f**)

Figure 5. Performance and signature detection rates of the proposed models with different model parameter pruning rates for different NLG tasks. (**a**) BLEU-1 scores on WMT14 dataset for the MT task. (**b**) Signature detection rates on WMT14 dataset for the MT task. (**c**) SPICE scores on MS-COCO dataset for the IC task. (**d**) Signature detection rates on MS-COCO dataset for the IC task. (**e**) ROUGE-L scores on CNN/DM dataset for the TS task. (**f**) Signature detection rates on CNN/DM dataset for the TS task.

In practice, an attacker with a general understanding of our proposed model and knows the key is in place may try to prune the embedded key. Figure 6a,c,e show the performance of the proposed models with different key pruning rates for different NLG tasks. we can see that the performance of our proposed models gradually degrades with the increase of the key pruning rates, which is mainly due to the fact that the original embeded keys are changed more and more with the increase of the key pruning rates. In addition, we also conduct related experiments to test the signature detection rates of the proposed models with different key pruning rates for different NLG tasks on the corresponding dataset. The experimental results are shown as Figure 6b,d,f). It can be observed that the signature detection rates of the proposed models can reach more than 93% for different NLG tasks under different key pruning rates. Therefore, the secret key and signature embedded in our proposed models training process can protect the model IP against the key pruning attack.



(**a**) (**b**)

Figure 6. *Cont.*

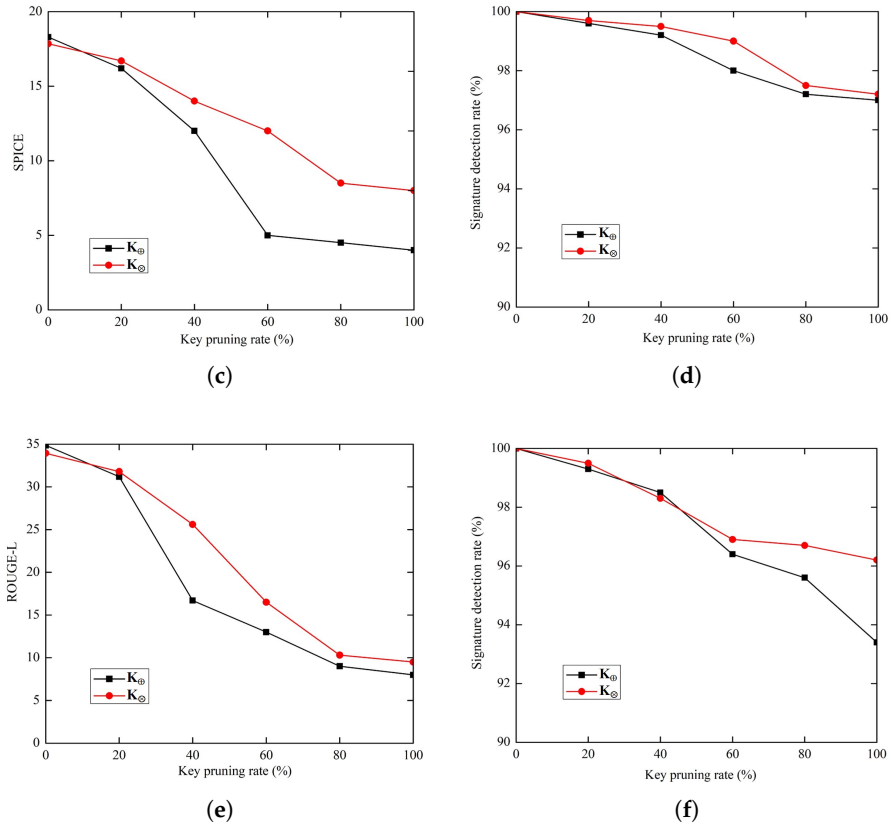


Figure 6. Performance and signature detection rates of the proposed models with different key pruning rates for different NLG tasks. (**a**) BLEU-1 scores on WMT14 dataset for the MT task. (**b**) Signature detection rates on WMT14 dataset for the MT task. (**c**) SPICE scores on MS-COCO dataset for the IC task. (**d**) Signature detection rates on MS-COCO dataset for the IC task. (**e**) ROUGE-L scores on CNN/DM dataset for the TS task. (**f**) Signature detection rates on CNN/DM dataset for the TS task.

4.6.2. Fine-Tuning

An attacker may fine-tune the stolen NLG model with a different dataset to obtain another model that almost inherits the performance of the original model while attempting to remove the signature embedded in the model. Table 9 shows the performance and signature detection rates of the proposed models after fine-tuning on different datasets for different NLG tasks. It can be seen that in the original NLG tasks, our proposed models can detect signatures with almost 100% accuracy. In addition, Table 9 shows that after fine-tuning the proposed models with different datasets, the model can achieve similar performance as the baseline models, but the signature detection rates drop by nearly 25%. This is a defect of the proposed approach, but in general it does not compromise the IP protection of the NLG model since we still have the embedded key as proof of ownership. On the whole, the secret key embedded in the proposed models works jointly with the signature can protect model ownership for fine-tuning with different datasets.

Table 9. Performance and signature detection rates of the proposed models after fine-tuning on different datasets for the different tasks. "Fine-tuning" in bracket indicates that the proposed model is fine-tuned on a dataset different from the original dataset.

	Metric	BLEU-1		Signature detection rate (%)	
MT	Datasets	WMT14	WMT17 (Fine-tuning)	WMT14	WMT17 (Fine-tuning)
	Baseline	29.82	26.60	-	-
	$\mathbf{K}_{\oplus}$	28.54	26.53	100	73.40
	$\mathbf{K}_{\oplus}$	28.37	25.89	99.99	72.62
	Metric	SPICE		Signature detection rate (%)	
IC	Datasets	MS-COCO	Flickr30k (Fine-tuning)	MS-COCO	Flickr30k (Fine-tuning)
	Baseline	18.51	13.12	-	-
	$\mathbf{K}_{\oplus}$	18.30	13.06	100	72.25
	$\mathbf{K}_{\otimes}$	17.85	13.08	100	74.38
	Metric	ROUGE-L		Signature detection rate (%)	
TS	Datasets	CNN/DM	Gigaword (Fine-tuning)	CNN/DM	Gigaword (Fine-tuning)
	Baseline	35.29	33.17	-	-
	$\mathbf{K}_{\oplus}$	34.87	33.02	99.99	75.09
	$\mathbf{K}_{\otimes}$	33.95	32.52	99.98	77.47

In practice, if an attacker has full knowledge of the proposed model, including the model training process, specific training parameters, datasets and embedded key/signature, it is possible to fine-tune the proposed model with a different key/signature according to the same training steps. The performance and signature detection rates of the proposed models after key/signature fine-tuning on the corresponding datasets for the different tasks are shown as Table 10. As shown in Table 10, the performance of the proposed models after key/signature fine-tuning slightly degrades compared to the original protected models. In addition, the signature detection rates of the proposed models can be maintained at around 70%. It can be seen that when the attacker has full knowledge of the proposed models and uses key/signature fine-tuning, the model IP can still be easily protected. However, this scenario is difficult to happen in practice.

Table 10. Performance and signature detection rates of the proposed model after key/signature fine-tuning on the corresponding datasets for the different tasks. "Attack" in bracket indicates the proposed model after key/signature fine-tuning.

	Metric	BLEU-1		Signature detection rate (%)	
MT	Datasets	WMT14	WMT14 (Attack)	WMT14	WMT14 (Attack)
	$\mathbf{K}_{\oplus}$	28.54	25.85	100	70.66
	$\mathbf{K}_{\otimes}$	28.37	24.70	99.99	70.34
	Metric	SPICE		Signature detection rate (%)	
IC	Datasets	MS-COCO	MS-COCO (Attack)	MS-COCO	MS-COCO (Attack)
	$\mathbf{K}_{\oplus}$	18.30	16.71	100	69.18
	$\mathbf{K}_{\otimes}$	17.85	14.46	100	68.73
	Metric	ROUGE-L		Signature detection rate (%)	
TS	Datasets	CNN/DM	CNN/DM (Attack)	CNN/DM	CNN/DM (Attack)
	$\mathbf{K}_{\oplus}$	34.87	30.82	99.99	72.24
	$\mathbf{K}_{\otimes}$	33.95	30.97	99.98	69.05

5. Conclusions

NLG models have received extensive attention and applications due to their increasing mature in combination with powerful DNN in recent years. However, the deployed NLG models may be stolen or used illegally in practical applications. Several recent works on IP protection of NLG models mostly use watermarks to identify and verify model ownership. However, it remains questionable whether these watermarking approaches provide reliable and timely protection of models. In addition, the quality of the watermarked text sequences generated by some watermarking approaches is not high. In this paper, we propose two embeding schemes to the hidden memory state of the NLG model decoder RNN to protect the IP of the models for different tasks, and we add a trained AWD-LSTM to the model decoder to fine-tune the model during the training process to improve the grammatical correctness of the output text sequences. Experimental results show that our proposed IP protection approach is able to maintain the original performance of the NLG model and outputs high-quality text sequences, while forged keys will generate unusable NLG models, thus defeating the purpose of model infringement. We also conduct sufficient experiments to demonstrate the robustness of the proposed IP protection of NLG models under various attacks. Table 9 shows that after fine-tuning the proposed models with different datasets, the model can achieve similar performance as the baseline models, but the signature detection rates drop by nearly 25%. In future work, we will optimize the proposed watermarking model to further improve its robustness, so that the model can effectively resist various common attacks. In addition, we will attempt to generalize similar model IP protection approach to the large language model represented by ChatGPT.

Author Contributions: Validation, Z.W. and X.Z.; Writing—original draft, M.L. All authors have read and agreed to the published version of the manuscript.

Funding: National Natural Science Foundation of China (NSFC) under Grant U22B2047 and Grant U1936214.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, W.E.; Sheng, Q.Z.; Alhazmi, A.; Li, C. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol. (TIST)* **2020**, *11*, 1–41. [CrossRef]
2. Mittal, V.; Gangodkar, D.; Pant, B. Exploring The Dimension of DNN Techniques For Text Categorization Using NLP. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 497–501.
3. Leyh-Bannurah, S.R.; Tian, Z.; Karakiewicz, P.I. Deep learning for natural language processing in urology: State-of-the-art automated extraction of detailed pathologic prostate cancer data from narratively written electronic health records. *JCO Clin. Cancer Inform.* **2018**, *2*, 1–9. [CrossRef] [PubMed]
4. Maurya, S.; Singh, V.; Verma, N.K. Condition-based monitoring in variable machine running conditions using low-level knowledge transfer with DNN. *IEEE Trans. Autom. Sci. Eng.* **2020**, *18*, 1983–1997. [CrossRef]
5. Vijayakumar, K.; Kadam, V.J.; Sharma, S.K. Breast cancer diagnosis using multiple activation deep neural network. *Concurr. Eng.* **2021**, *29*, 275–284. [CrossRef]
6. Qin, P.; Zhang, J.; Zeng, J.; Liu, H.; Cui, Y. A framework combining DNN and level-set method to segment brain tumor in multi-modalities MR image. *J. Abbr.* **2019**, *23*, 9237–9251. [CrossRef]
7. Xiang, T.; Xie, C.; Guo, S.; Li, J.; Zhang, T. Protecting Your NLG Models with Semantic and Robust Watermarks. *arXiv* **2021**, arXiv:2112.05428.
8. Wallace, E.; Stern, M.; Song, D. Imitation Attacks and Defenses for Black-box Machine Translation Systems. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 5531–5546.
9. Krishna, K.; Tomar, G.S.; Parikh, A.P. Thieves on Sesame Street! Model Extraction of BERT-based APIs. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
10. He, X.; Lyu, L.; Xu, Q.; Sun, L. Model Extraction and Adversarial Transferability. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Mexico City, Mexico, 6–11 June 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 2006–2012.

11. Uchida, Y.; Nagai, Y.; Sakazawa, S.; Satoh, S. Embedding Watermarks into Deep Neural Networks. In Proceedings of the 2017 Acm on International Conference on Multimedia Retrieval, Bucharest, Romania, 6–9 June 2017; pp. 269–277.
12. Zhang, J.; Gu, Z.; Jang, J.; Wu, H.; Stoecklin, M.P.; Huang, H.; Molloy, I. Protecting Intellectual Property of Deep Neural Networks with Watermarking. In Proceedings of the 2018 on Asia Conference on Computer and Communications Security, Incheon, Republic of Korea, 4–8 June 2018; Volume 10, pp. 159–172.
13. Fan, L.; Ng, K.W.; Chan, C.S.; Yang, Q. DeepIP: Deep Neural Network Intellectual Property Protection with Passports. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 6122–6139. [CrossRef] [PubMed]
14. Wu, H.; Liu, G.; Yao, Y.; Zhang, X. Watermarking Neural Networks with Watermarked Images. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 2591–2601. [CrossRef]
15. Lim, J.H.; Chan, C.S.; Ng, K.W. Protect, show, attend and tell: Empowering image captioning models with ownership protection. *Pattern Recogn.* **2022**, *122*, 108285. [CrossRef]
16. Chen, X.; Salem, A.; Backes, M.; Ma, S.; Zhang, Y. Badnl: Backdoor attacks against nlp models. In Proceedings of the ICML 2021 Workshop on Adversarial Machine Learning, Virtual Event, China, 24 July 2021.
17. Shen, L.; Ji, S.; Zhang, X.; Li, J. Backdoor Pre-trained Models Can Transfer to All. In Proceedings of the Conference on Computer and Communications Security, Virtual Event, Republic of Korea, 15–19 November 2021; pp. 3141–3158.
18. Dai, J.; Chen, C.; Li, Y. A backdoor attack against lstm-based text classification systems. *IEEE Access* **2019**, *7*, 138872–138878. [CrossRef]
19. He, X.; Xu, Q.; Lyu, L.; Wu, F.; Wang, C. Protecting intellectual property of language generation apis with lexical watermark. *Proc. AAAI Conf. Artif. Intell.* **2022**, 10758–10766. [CrossRef]
20. He, X.; Xu, Q.; Zeng, Y.; Lyu, L.; Wu, F.; Li, J.; Jia, R. CATER: Intellectual Property Protection on Text Generation APIs via Conditional Watermarks. In Proceedings of the 36th Conference on Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022.
21. Kamaruddin, N.S.; Kamsin, A.; Por, L.Y.; Rahman, H. A review of text watermarking: Theory, methods, and applications. *IEEE Access* **2018**, *6*, 8011–8028. [CrossRef]
22. Kaur, M.; Mahajan, K. An existential review on text watermarking techniques. *Int. J. Comput. Appl.* **2015**, *120*, 29–32. [CrossRef]
23. Singh, P.; Chadha, R.S. A survey of digital watermarking techniques, applications and attacks. *Int. J. Eng. Innov. Technol. (IJEIT)* **2013**, *2*, 165–175.
24. Adi, Y.; Baum, C.; Cisse, M.; Pinkas, B.; Keshet, J. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In Proceedings of the 27th USENIX Security Symposium (USENIX Security 18), Baltimore MD, USA, 15–17 August 2018; pp. 1615–1631.
25. Le Merrer, E.; Perez, P.; Trédan, G. Adversarial frontier stitching for remote neural network watermarking. *Neural Comput. Appl.* **2020**, *32*, 9233–9244. [CrossRef]
26. Quan, Y.; Teng, H.; Chen, Y.; Ji, H. Watermarking deep neural networks in image processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1852–1865. [CrossRef] [PubMed]
27. Darvish Rouhani, B.; Chen, H.; Koushanfar, F. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, Providence, RI, USA, 13–17 April 2019; pp. 485–497.
28. Vaswani, A.; Shazeer, N.; Parmar, N. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
29. Shetty, R.; Schiele, B.; Fritz, M. A4nt: Author attribute anonymity by adversarial training of neural machine translation. In Proceedings of the 27th USENIX Security Symposium (USENIX Security 18), Baltimore, MD, USA, 15–17 August 2018.
30. Merity, S.; Keskar, N.S.; Socher, R. Regularizing and optimizing lstm language models. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
31. Howard, J.; Ruder, S. Universal language model fine-tuning for text classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), Melbourne, Australia, 15–20 July 2018.
32. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.G.; Le, Q.; Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy, 28 July–2 August 2019.
33. Carlini, N.; Liu, C.; Erlingsson, U.; Kos, J.; Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In Proceedings of the 28th USENIX Security Symposium (USENIX Security 19), Santa Clara, CA, USA, 14–16 August 2019.
34. Koehn, P.; Hoang, H.; Birch, A. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, Prague, Czech Republic, 24–29 June 2007; Association for Computational Linguistics: Stroudsburg, PA, USA, 2007; pp. 177–180.
35. Karpathy, A.; Li, F.-F. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3128–3137.
36. See, A.; Liu, P.J.; Manning, C.D. Get to the point: Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1073–1083.

37. Szyller, S.; Atli, B.G.; Marchal, S.; Asokan, N. Dawn: Dynamic adversarial watermarking of neural networks. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual Event, China, 20–24 October 2021; pp. 4417–4425.
38. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014), Banff, AB, Canada, 14–16 April 2014.
39. Xu, K.; Ba, J.; Kiros, R.; Cho, K. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2048–2057.
40. Nallapati, R.; Zhou, B.; Gulcehre, C.; Xiang, B. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In Proceedings of the Conference on Computational Natural Language Learning (CoNLL 2016), Berlin, Germany, 11–12 August 2016.
41. Zeiler, M.D. ADADELTA: An adaptive learning rate method. *arXiv* **2012**, arXiv:1212.5701.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
43. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
44. Papineni, K.; Roukos, S.; Ward, T. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002; Volume 10, pp. 311–318.
45. Li, D.; Zhang, Y.; Peng, H. Contextualized perturbation for textual adversarial attack. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics, Online, 6–13 June 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 5053–5069.
46. Anderson, P.; Fernando, B.; Johnson, M. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 382–398.
47. Lin, C.Y. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2004; pp. 74–81.
48. See, A.; Luong, M.-T.; Manning, C.D. Compression of neural machine translation models via pruning. *arXiv* **2016**, arXiv:1606.09274.