Converging Symbolic and Semantic Spaces in Linguistic Steganography

Anwesha Halder
College of Engineering Bhubaneswar

***Abstract*—Prior research on language steganography, such as synonym replacement and sampling-based techniques, typically involved the intentional manipulation of observable symbols to hide sensitive information, which raised security concerns. In this letter, we investigated generation-based linguistic steganography in latent space by encoding hidden messages in the selection of implicit characteristics (semanteme) of natural language, hence preventing simple operations on seen symbols. We put out a brand-new rejection sampling-based linguistic semantic steganography framework. In particular, we used a semantic classifier for extraction and a controlled text generation model for embedding. A model based on BERT and CTRL is used in experiments to provide further quantitative evaluation. The results show that our method can achieve nearly flawless imperceptibility and satisfactory efficiency.**

***Index Terms*—Controllable text generation, linguistic steganography, rejection sampling, semantic steganography.**

I. INTRODUCTION

STEGANOGRAPHY [1] is the art and science of communicating in such a way that the presence of secret messages cannot be detected. The research and exploration of modern steganography began in 1984 when Simmons proposed *the Prisoners' Problem* [2]: Alice and Bob are in jail and locked up in separate cells, where the only communication channel is monitored by a warden Eve. They attempt to hatch an escape plan furtively together by means of communicating with seemingly innocuous steganographic carriers (stegocarriers) with secret messages concealed inside. Once Eve perceives anything beyond the range of what is expected, she will cut off the channel and the plan will fail. Hence, the problem of steganography can be described as: how can Alice and Bob make the stegocarriers *secure* enough so as not to arouse Eve's suspicion?

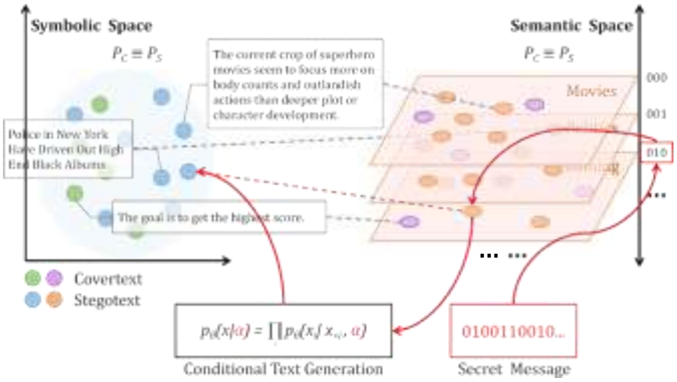

Symbolic Space
$P_C = P_S$
Police in New York Have Driven Out High End Black Albums
The current crop of superhero movies seem to focus more on body counts and outlandish actions than deeper plot or character development.
The goal is to get the highest score.
Semantic Space
$P_C = P_S$
Movies
000
001
010
...
... ...
Covertext
Stegotext
$p_\theta(x|\alpha) = \prod p_\theta(x_i | x_{<i}, \alpha)$
Conditional Text Generation
0100110010...
Secret Message

Fig. 1. This figure is a sketch map of the procedure of linguistic semantic steganography. Firstly secret message is mapped into a discrete semantic space and then the corresponding semantic vector α is fed to the conditional text generation module to yield stegotext x.

There are multiple points of view on the measurement of *security* in steganography. One of the most widely accepted is the *information-theoretic security* proposed by Cachin in 1998 [3], which is the relative entropy (Kullback–Leibler divergence, KLD) between distributions of covercarriers P_C and stegocarriers P_S. The lower the entropy, the higher the security. A minimum value zero appears when the steganographic system (stegosystem) is perfectly secure, which means covercarriers and stegocarriers are statistically undistinguishable, or rather, imperceptible.

Modern steganography is generally understood to deal with digital media such as images [4], audios [5], videos [6] and text [7]. Thereinto, text is prevalent in our daily life, which makes it practical to serve as carriers. In 2004, Bennett [8] summarized two strategies of linguistic steganography: modification-based steganography and generation-based steganography. Synonym substitution is a common way of linguistic modification steganography [9]–[11], where Alice replaces some words in a covertext with their synonyms according to a special set of rules to conceal secret messages. However, since text has limited information redundancy compared to images and audios [12], it can result in syntactic and semantic unnaturalness easily to conduct a substitution operation directly in the observed symbolic space (token space) [13]. As for linguistic generation steganography, researchers usually utilized autoregressive language model and employed a steganographic sampling (stegosampling) algorithm to embed secret messages during the generation procedure [14]–[21]. It may mitigate the unnatural issue as the output probability distribution of the current token is affected by all historical decisions to maintain the fluency of stegotext. Nevertheless, it is still a straightforward *manipulation* of the observed symbols. Stegosampling algorithms will inevitably do damage to the explicit distribution of tokens [17], [18] and result in a gap between stegotext and covertext.

As monitoring party of the public channel, Eve is usually assumed to have full knowledge about the distribution of covertext P_C, which means huge potential security problems for symbolic steganography that does harm to the explicit distribution of tokens. Therefore, it is more reasonable to conceal secret messages in an implicit way, for example, to manipulate a latent space $\mathcal{Z}$. Secret messages are firstly mapped into a latent variable z and then fed to a conditional generative model $p_\theta(x|z)$ to yield stegocarrier x. As long as the prior distribution $p(z)$ keeps unchanged, it will generate stegocarriers with the same distribution as that generated by innocent users according to $p_\theta(x) = \int p_\theta(x|z)p(z)$, which provides a safeguard for the information-theoretic security $D_{KL}(P_C, P_S)$. There have been attempts of such strategy in the field of image steganography. Liu *et al.* [22] utilized ACGAN [23] to embed secret messages into the class labels of generated images. Chen *et al.* [24] employed variational auto-encoders (VAEs) [25] and flow-based models [26] to conduct latent space steganography. As for linguistic steganography, latent space such as semanteme space, sentiment space and so on can also be employed to conceal secret messages.

In this letter, we consider about avoiding explicit symbolic manipulation and strive to take a step towards linguistic semantic steganography. Our research is carried out from the perspective of information-theoretic security, which aims to enhance the imperceptibility of stegotext. We proposed a novel framework of linguistic semantic steganography, which is illustrated in Fig. 1. We divided the implicit semantic space into a finite number of separate zones and encoded secret messages in the selection of semantic zones. Rejection sampling strategy based on controllable text generation model is employed to generate stegotext and semantic classifier is adopted to ensure a completely correct extraction. To verify the effectiveness of the framework, we implemented a model for quantitative evaluation. Experimental results show that the proposed method is able to achieve satisfactory efficiency and nearly perfect imperceptibility.

II. Novel Framework of Linguistic Semantic Steganography

In this section, we demonstrate a novel framework of linguistic semantic steganography, which takes advantage of the semanteme α of text x to conceal secret messages. The basic idea is to adopt a controllable text generation model $p_\theta(x|\alpha)$ and a semantic classifier model $p_\varphi(\alpha|x)$ for Alice and Bob respectively. Controllable text generation is the task of learning distribution $p(x|\alpha)$ of text x conditioned on semanteme α. It can be factorized with the chain rule of probability as follow

$$p(x|\alpha) = \prod_i p(x_i|x_{<i}, \alpha), \qquad (1)$$

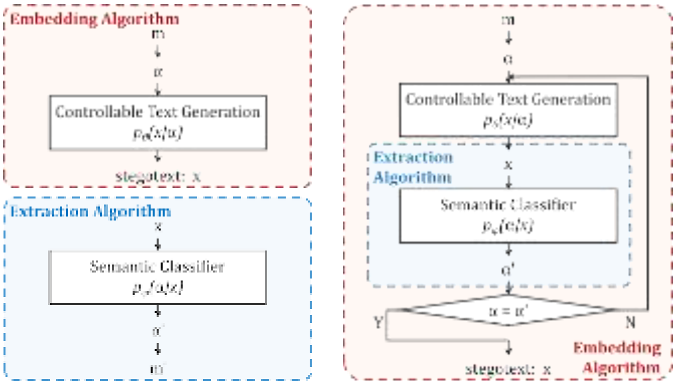
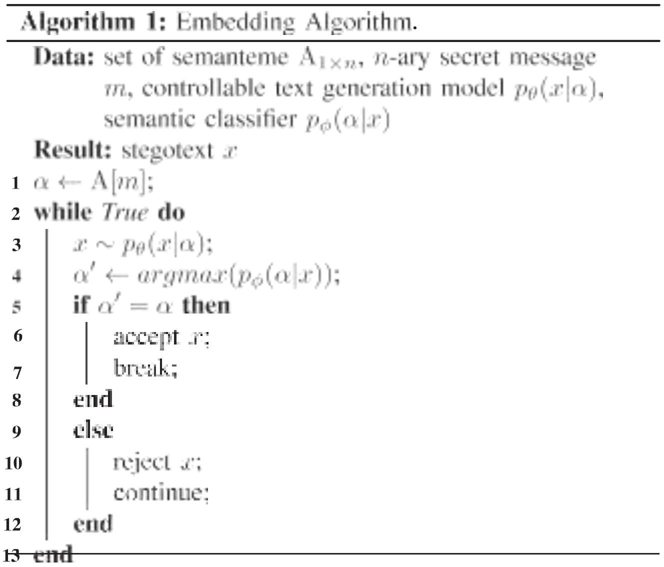


Fig. 2. This figure illustrates the basic idea of the framework of linguistic semantic steganography (left) and that based on rejection sampling strategy (right). Secret messages are encoded in the selection of semanteme α. In basic embedding algorithm (left top), stegotext is the direct output of the controllable text generation model, which may lead to errors in extraction algorithm (left bottom). To address this gap, we employed rejection sampling strategy (right). Extraction algorithm acts as a pre-set hypothesis in embedding algorithm to guarantee absolutely correct extraction at Bob's end.

where x_i denotes the i-th token in the generated sequence. In our framework, semantic space is divided into n separate zones that is able to encode $log(n)$-bit information. At Alice's end, based on controllable text generation model $p_\theta(x|\alpha)$, the generation of stegotext is conditioned on α that corresponds to the secret message m to be embedded. The steganographic generating procedure is just a simple sampling process like any other innocuous text generation. In this framework, no manipulation of the explicit conditional distribution of observed variable is required. According to the aforementioned analysis, it is capable of achieving nearly perfect imperceptibility if the prior $p(\alpha)$ remains unchanged. The condition is obviously satisfied because secret information is often thought of as being uniformly distributed, meanwhile innocent users randomly choose a semanteme to generate innocuous text.

At Bob's end, stegotext is received and inputted into the semantic classifier $p_\varphi(\alpha|x)$ to regain the semanteme α for information extraction. A diagram of the basic idea is illustrated in Fig. 2 (left). Such a process requires that the two models are infinitely accurate in fitting true probability distributions $p(x|\alpha)$ and $p(\alpha|x)$. On the one hand, it is hard for controllable text generation model to depict different semanteme precisely. Generating completely irrelevant text given orthogonal attributes is still an elusive task. On the other hand, the semantic classifier can also be defective. Absolutely correct extraction won't come true as long as one model is biased.

To get around this, we adopt rejection sampling strategy as shown in Fig. 2 (right). Rejection sampling $G(p, H_0)$ is a special sampling way that repeats basic sampling method from distribution p until the sampling value accept a pre-set hypothesis H_0 [24], [27]. Here the distribution p is set to the output of controllable text generation model $p_\theta(x|\alpha)$ and H_0 means correct extraction with semantic classifier. Once misrecognition occurs, the stegotext will be rejected. The precision with which different semantemes are distinguished will affect



Algorithm 1: Embedding Algorithm.

```
Data: set of semanteme $A_{1\times n}$, $n$-ary secret message
      $m$, controllable text generation model $p_\theta(x|\alpha)$,
      semantic classifier $p_\phi(\alpha|x)$
Result: stegotext $x$
$\alpha \leftarrow A[m]$;
while True do
    $x \sim p_\theta(x|\alpha)$;
    $\alpha' \leftarrow argmax(p_\phi(\alpha|x))$;
    if $\alpha' = \alpha$ then
        accept $x$;
        break;
    end
    else
        reject $x$;
        continue;
    end
end
```

Algorithm 2: Extraction Algorithm.

```
Data: set of semanteme $A_{1\times n}$, stegotext $x$, semantic
      classifier $p_\varphi(\alpha|x)$
Result: $n$-ary secret message $m$
$\alpha' \leftarrow argmax(p_\varphi(\alpha|x))$;
$m \leftarrow A.indexof(\alpha')$;
```

the count of loops in rejection sampling, which reflects the efficiency of steganography. Higher extraction accuracy means less regenerating and vice versa. In extreme circumstances, 100% accurate model is identical to the aforementioned situation without rejection sampling strategy while 0% accurate model is unable to generate stegotext in a limited time, which makes no sense in practice. The embedding algorithm in our framework is demonstrated in Algorithm 1.

The extraction algorithm is actually same as that in the basic idea and that in rejection sampling strategy. The pseudocode is demonstrated in Algorithm 2.

III. Methodology of Our Implementation

In order to verify the effectiveness of the proposed framework of linguistic semantic steganography, we constructed a concrete model for further assessment. We adopted large-scale Transformer-based models as controllable text generation model and semantic classifier. With the development of natural language processing, they can be replaced with more powerful models in the future.

We employed BERT [29] to construct the semantic classifier $p_\varphi(\alpha|x)$, which is an effective and efficient feature extractor that takes full advantage of the bidirectional context of tokens and shows powerful capacity for feature extraction. In terms of model structure, BERT is basically a multi-layer bidirectional Transformer encoder stack. The core function in a Transformer architecture [32] is *multi-head self-attention*. For the input feature of the i-th token in a sequence $X_i \in \mathbb{R}^{1\times d_{in}}$, result is calculated by

$$MultiHead(X_i) = Concat(h_1, ..h_h)W^O, \quad (2)$$

where

$$h_j = Attn(X_i W_j^Q, X_i W_j^K, X_i W_j^V), \quad j = 1, ..., m$$

$$Attn(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V. \quad (3)$$

$Q, K \in \mathbb{R}^{d_k\times 1}$ and $V \in \mathbb{R}^{d_v\times 1}$ are auxiliary variables to calculate self-attention. $W_j^Q, W_j^K \in \mathbb{R}^{d_{in}\times d_k}$, $W_j^V \in \mathbb{R}^{d_{in}\times d_v}$ and $W^O \in \mathbb{R}^{d_m \cdot d_v \times d_{out}}$ are trainable parameters. In our implementation, we combined the pre-trained base version with a sequence classification head to construct the semantic classifier. It is fine-tuned on the text generated by controllable text generation model with different semantemes.

We utilized the released CTRL [28] as controllable text generation model $p_\theta(x|\alpha)$, which is a conditional Transformer language model for controllable text generation. It can generate fluent text conditioned on 55 handcrafted control codes that specify certain semantemes.

In our implementation, Alice selected 1 of n control codes α to embed $log(n)$-bit secret information and generate stegotext x with the released CTRL model by rejection sampling strategy. At Bob's end, the secret information is extracted from x with the BERT-based semantic classifier, which is fine-tuned on text generated by the released CTRL to learn the high-dimensional boundaries of different semantemes. In this scenario, semanteme remains uniformly distributed and stegotext is directly generated by CTRL without any manipulation in observed symbolic space, which makes the stegotext nearly undistinguishable from innocuous covertext.

IV. Experiments and Analysis

We carried out a series of quantitative studies on the efficiency and imperceptibility of stegotext generated by the proposed method. In our experiment, the n selected control codes ($n = 2, 3, 4, 6, 8, 12, 16$) are "Movies," "Translation," "Pregnancy," "Christianity," "Politics," "Feminism," "Writing," "Netflix," "Gaming," "India," "Diet," "Legal," "Science," "Horror," "Links" and "News". For each control code, we generated 5000 text with length 50 by the released CTRL for fine-tuning the BERT-based semantic classifier. We took the parameters with the best test loss during 10-epoch training (with learning rate 0.001 and batch size 16) as the final training result. The test accuracy listed in the second column of Table I shows that the more control codes, the more difficult it is to train the classifier. The accuracy is 98.80% when $n = 2$ while it is only 69.14% when $n = 16$. Details of the statistical distribution of the generated stegotext can be found in Fig. 3.

To test the performance of efficiency, we generated 1000 stegotext and investigated the average loop count in rejection sampling. Results are listed in the third column of Table I. When there are fewer control codes, the average loop count is just a little bit more than the ideal value 1 as the semantic classifier is more accurate. The gap gets bigger when n increases but it is

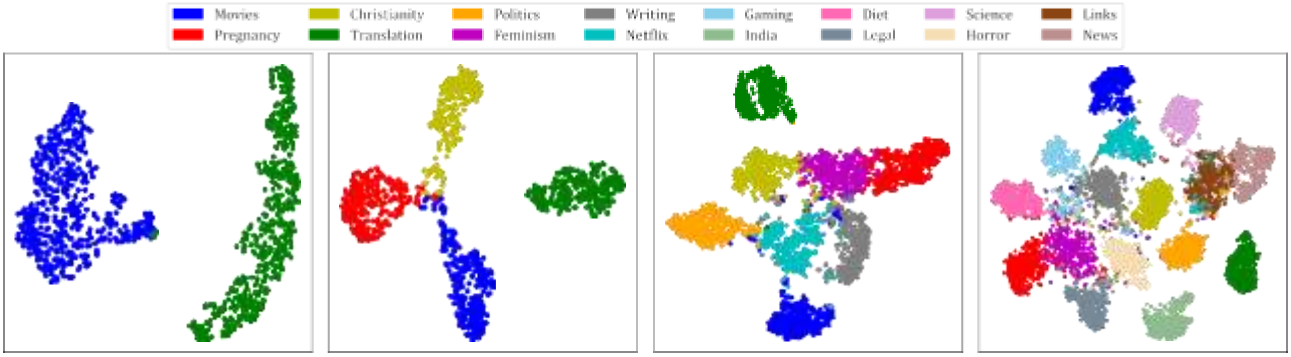


Fig. 3. This figure demonstrates the distribution of stegotext in semantic space under the framework of linguistic semantic steganography when $n = 2, 4, 8, 16$. Each node represents the semantic feature (pooler output of the fine-tuned BERT-based semantic classifier) of a certain stegotext generated by the proposed method, which is reduced to a 2-dimensional vector by t-SNE [33]. Semantic space can be divided roughly into n zones to encode $log(n)$-bit information.

TABLE I
EXPERIMENTAL RESULTS OF EFFICIENCY TEST AND QUALITY TEST

n	Test Acc	Loop Count ↓	Perplexity ↓
2	0.9880	1.0160 ± 0.1406	13.9565 ± 13.8297
3	0.9653	1.0380 ± 0.1913	14.8019 ± 09.1596
4	0.9205	1.0980 ± 0.3294	16.0039 ± 10.2108
6	0.8293	1.2330 ± 0.5736	16.5126 ± 09.4767
8	0.7650	1.3460 ± 0.7648	18.6617 ± 11.9872
12	0.7058	1.4710 ± 1.0170	19.3392 ± 12.8835
16	0.6914	1.5460 ± 1.0822	19.8346 ± 12.4944

TABLE II
EXPERIMENTAL RESULTS OF ANTI-STEGANALYSIS TEST (ACCURACY)

MODEL	[34]	[35]	BERT Classifier
n=2	0.5050	0.5195	0.5235
n=3	0.5070	0.5380	0.5195
n=4	0.5090	0.5225	0.5290
n=6	0.4945	0.4955	0.5290
n=8	0.5155	0.5230	0.5400
n=12	0.4945	0.4990	0.5260
n=16	0.4820	0.5170	0.5390
[15]	0.8870	0.8145	0.9430
[15] + {n=8}	0.8890	0.7935	0.9485
[15] + {n=16}	0.8710	0.7710	0.9260

still acceptable. Even when $n = 16$ (the accuracy of semantic classifier is only 69.14%), it only takes an average of 1.5460 loops to generate a stegotext containing 4 bits secret information, which is efficient enough. The results indicate good practicality of rejection sampling strategy.

To evaluate the performance of imperceptibility, we tested the quality of the generated stegotext and its ability to resist steganalysis. We utilized the metric *perplexity* (PPL) to measure the quality, and we obtained satisfactory results demonstrated in the last column of Table I. In fact, we noticed that the qualtiy of stegotext generated with different control codes may be varied. For example, we found the average PPL of stegotext conditioned on "Translation" and "Netflix" is 7.8292 and 30.1886, respectively. In actual use, according to the value of n, we can preferentially select control codes with smaller PPL, so as to ensure the quality and imperceptibility of the generated stegotext as much as possible.

For anti-steganalysis test, we utilized two commonly used methods [34], [35] as well as the one based on BERT to distinguish stegotext from covertext standing in the shoes of Eve. We took 1000 natural text generated by innocent users as covertext and conducted 10-fold cross validation to reduce uncertainty. We adopted the result with the lowest test loss during 50-epoch training. For [34], we set the window size to 100 and learning rate to 0.1. For [35], we constructed filters with size 3, 4, 5 and number 32. Learning rate was set to 0.01. For BERT-based steganalysis method, we used smaller learning rate 0.001. We also implemented a typical generation-based symbolic stegano- graphic method [15] with 1 bit per word embedded for compar- ison. Experimental results are shown in Table II, from which we found it is able to achieve nearly perfect imperceptibility underthe proposed framework of linguistic semantic steganography. Since the proposed approach does not explicitly manipulate the symbols in observed space, Eve is unable to detect any abnormality through statistical analysis of the observed tokens.More importantly, we found the proposed stategy of linguistic semantic steganography does not conflict with the traditional symbolic steganography, and they can be performed at the sametime, which is shown in the last two rows of Table II. It reveals that superimposing the proposed method on symbolic stegano- graphic methods can further improve its embedding capacity without damaging the anti-steganalysis ability.

V. CONCLUSION

In order to confirm the effectiveness and imperceptibility of our unique rejection sampling strategy-based framework for linguistic semantic steganography, we employed a model built with BERT and CTRL in this work. This letter can be considered a first attempt, and we hope that additional research on the concept of latent space steganography will be possible. We also discovered that it is possible to concurrently embed hidden messages in latent and visible space; this is a topic we think is worth exploring further.

References

[1] N. F. Johnson and S. Jajodia, “Exploring steganography: Seeing the unseen,” *Computer*, vol. 31, no. 2, pp. 26–34, 1998.

[2] G. J. Simmons, “The prisoners’ problem and the subliminal channel,” in *Proc. Adv. Cryptol.*, 1984, pp. 51–67.

[3] C. Cachin, “An information-theoretic model for steganography,” in *Proc. Int. Workshop Inf. Hiding.*, 1998, pp. 306–318.

[4] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. Ho, and K.-H. Jung, “Image steganography in spatial domain: A survey,” *Signal Process.: Image Commun.*, vol. 65, pp. 46–66, 2018.

[5] N. Kaur and S. Behal, “Audio steganography techniques-a survey,” in *Proc. Int. J. Eng. Res. Appl. ISSN*, 2014, pp. 2248–9622.

[6] Y. Liu, S. Liu, Y. Wang, H. Zhao, and S. Liu, “Video steganography: A review,” *Neurocomputing*, vol. 335, pp. 238–250, 2019.

[7] R. B. Krishnan, P. K. Thandra, and M. S. Baba, “An overview of text steganography,” in *Proc. 4th Int. Conf. Signal Process., Commun. Netw.*, 2017, pp. 1–6.

[8] K. Bennett, “Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text,” TR 2004-13, May 2004.

[9] I. A. Bolshakov, “A method of linguistic steganography based on collocationally-verified synonymy,” in *Proc. Int. Workshop Inf. Hiding.*, 2004, pp. 180–191.

[10] C. Y. Chang and S. Clark, “Practical linguistic steganography using contextual synonym substitution and vertex colour coding,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2010, pp. 1194–1203.

[11] W. Hao, L. Xiang, Y. Li, P. Yang, and X. Shen, “Reversible natural language watermarking using synonym substitution and arithmetic coding,” *Comput., Mater. Continua*, vol. 55, pp. 541–559, Jan. 2018, doi: 10.3970/cmc.2018.03510.

[12] M. Agarwal, “Text steganographic approaches: A comparison,” *Int. J. Netw. Secur. & Appl.*, vol. 5, no. 1, pp. 91–106, Jan. 2013.

[13] M. Grosvald and C. O. Orgun, “Free from the cover text: A human-generated natural language approach to text-based steganography,” *J. Inf. Hiding Multimedia Signal Process.*, vol. 2, no. 2, pp. 133–141, 2011.

[14] T. Fang, M. Jaggi, and K. J. Argyraki, “Generating steganographic text with LSTMs,” in *Proc. Student Res. Workshop*, Vancouver, Canada, Jul. 2017, pp. 100–106. [Online]. Available: https://www.aclweb.org/anthology/P17-3017.

[15] Z. Yang, X. Guo, Z. Chen, Y. Huang, and Y. Zhang, “RNN-Stega: Linguistic steganography based on recurrent neural networks,” *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 5, pp. 1280–1295, May 2019.

[16] Z. Yang, P. Zhang, M. Jiang, Y. Huang, and Y.-J. Zhang, “RITS: Real-time interactive text steganography based on automatic dialogue model,” in *Proc. Int. Conf. Cloud Comput. Secur.*, 2018, pp. 253–264.

[17] F. Z. Dai and Z. Cai, “Towards near-imperceptible steganographic text,” in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, Jul. 2019, pp. 4303–4308. [Online]. Available: https://www.aclweb.org/anthology/P19-1422

[18] Z. M. Ziegler, Y. Deng, and A. M. Rush, “Neural linguistic steganography,” in *Proc. Conf. Empirical Methods Natural Lang. Proc. 9th Int. Joint Conf. Natural Lang.*, Hong Kong, China, Nov. 2019, pp. 1210–1215, doi: 10.18653/v1/D19-1115.

[19] Z. Yang, N. Wei, Q. Liu, Y. Huang, and Y. Zhang, “GAN-TStega: Text steganography based on generative adversarial networks,” in *Proc. Int. Workshop Digit. Watermarking*, 2019, pp. 18–31.

[20] Z. Yang, B. Gong, Y. Li, J. Yang, Z. Hu, and Y. Huang, “Graph-Stega: Semantic controllable steganographic text generation guided by knowledge graph,” 2020, *arXiv:2006.08339*.

[21] Z. L. Yang, S. Y. Zhang, Y. T. Hu, Z. W. Hu, and Y. F. Huang, “VAE-Stega: Linguistic steganography based on variational auto-encoder,” *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 880–895, 2021.

[22] M. Liu, M. Zhang, J. Liu, Y. Zhang, and Y. Ke, “Coverless information hiding based on generative adversarial networks,” *Yingyong Kexue Xuebao/J. Appl. Sci.*, vol. 36, pp. 371–382, Mar. 2018, doi: 10.3969/j.issn.0255-8297.2018.02.015.

[23] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier GANs,” ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. Sydney, Australia: PMLR, Aug. 2017, pp. 2642–2651. [Online]. Available: http://proceedings.mlr.press/v70/odena17a.html

[24] K. Chen, H. Zhou, D. Hou, H. Zhao, W. Zhang, and N. Yu, “When provably secure steganography meets generative models,” 2018. [Online]. Available: http://arxiv.org/abs/1811.03732

[25] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2013, *arXiv:1312.6114*.

[26] D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1× 1 convolutions,” in *Proc. Adv. Neural Inf. Process. Syst. 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Red Hook, NY: Curran Associates, 2018, pp. 10 215–10 224.

[27] N. J. Hopper, J. Langford, and L. Von Ahn, “Provably secure steganography,” in *Proc. Annu. Int. Cryptol. Conf.*, 2002, pp. 77–92.

[28] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, “CTRL: A conditional transformer language model for controllable generation,” 2019, *arXiv:1909.05858*.

[29] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. Conf. North Ameri. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, Minneapolis, Minnesota, Jun. 2019, vol. 1, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[30] A. Vaswani *et al.*, “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst. 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY: Curran Associates, 2017, pp. 5998–6008.

[31] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

[32] Z. Yang, Y. Huang, and Y. Zhang, “A fast and efficient text steganalysis method,” *IEEE Signal Process. Lett.*, vol. 26, no. 4, pp. 627–631, Apr. 2019.

[33] Z. Yang, Y. Huang, and Y.-J. Zhang, “TS-CSW: Text steganalysis and hidden capacity estimation based on convolutional sliding windows,” *Multimedia Tools Appl.*, vol. 79, no. 25, pp. 18293–18316, Jul. 2020, doi: 10.1007/s11042-020-08716-w.