StegoZip: A Plug-and-Play Framework for Increasing Steganographic Payload Capacity with Large Language Model

Anonymous ACL submission

Abstract

Generative steganography is a current research hotspot, yet its secret message payload capacity is often limited by low entropy during generation. The low capacity necessitates long stego texts or numerous transmissions, increasing the risk of detection by third parties. Prior studies have primarily enhanced payload capacity by making more effective use of available entropy while largely overlooking the equally critical step of secret message preprocessing. In this paper, we propose *StegoZip*, the first plug-and-play framework that employs large language model-driven dynamic semantic redundancy pruning combined with index compression coding to optimize secret message preprocessing and further increase payload capacity. In combination with advanced steganography, the experimental results demonstrate that *StegoZip* can increase the payload capacity by 2–3× while reducing the time per unit message by approximately 50%. Furthermore, *StegoZip* operates independently of the steganography embedding process, ensuring that it does not impact the security of the original method.

1 Introduction

As an information-hiding technique, steganography aims to achieve covert communication by imperceptibly modifying cover media (e.g., images, audio, text) while avoiding detection by adversaries (Kahn, 1996; Provos and Honeyman, 2003; Channalli and Jadhav, 2009; Zhang et al., 2025). Unlike cryptography, which protects content through encryption, steganography ensures security by eliminating physical or statistical traces of hidden information in cover data (Johnson and Jajodia, 1998; Cachin, 1998; Hopper et al., 2002).

Linguistic steganography, which exploits text as the most prevalent communication medium, as shown in Fig. 1, typically follows two core phases during message encoding (Rani and Chaudhary, 2013; Krishnan et al., 2017; Majeed et al., 2021):

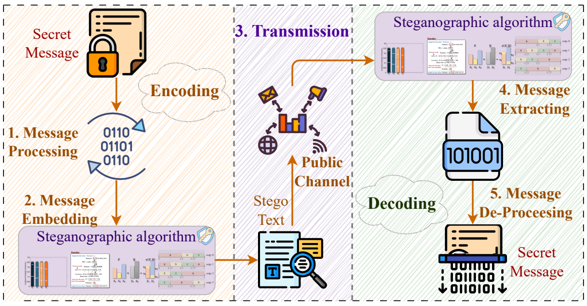

Figure 1: General process of linguistic steganography.

1) **Message Processing:** preprocessing secret messages through compression (e.g., Huffman coding), encryption (e.g., AES), and format conversion (e.g., ASCII-to-binary mapping). 2) **Message Embedding:** most of these methods adopt channel coding methods to embed messages while balancing imperceptibility and capacity, exemplified by Syndrome-Trellis Codes (STC) (Filler et al., 2011) and Steganographic Polar Codes (SPC) (Li et al., 2020). During message decoding, authorized receivers reconstruct the secret message through inverse transformations via shared keys. However, conventional methods face two principal limitations: restricted payload capacity (the ratio of secret message length to stego text length) and detectable statistical deviations between cover texts and stego texts (Wu et al., 2023).

With breakthroughs in generative large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023), a paradigm shift has emerged in steganography. The explicit output of token probability distributions by LLMs enables provably secure steganography under the security constraint of maintaining unchanged sampling distributions (Chen et al., 2018; Yang et al., 2018; Chen et al., 2021). ADG (Zhang et al., 2021) partitions vocabulary into clusters of similar probabilities through adaptive dynamic grouping and randomly selects tokens within clusters for information hiding; Meteor (Kaptchuk et al., 2021) proposes range reversible sampling that encodes messages as sampling interval offsets while compressing code length via shared prefixes; iMEC (de Witt et al., 2022) implements near-theoretical-limit embedding efficiency through iterative optimization of message encoding paths on the basis of minimum entropy coupling theory; Discop (Ding et al., 2023) decomposes high-dimensional token selection into multi-round binary decisions through Huffman tree construction of distribution copies, significantly reducing computational complexity.

However, although existing methods enhance payload capacity through iterative embedding (Yang et al., 2018; de Witt et al., 2022; Ding et al., 2023) and probability reordering (Kaptchuk et al., 2021), their optimizations focus solely on the message embedding phase, i.e., how to utilize the statistical characteristics of cover texts to embed secret messages more efficiently. This singular focus overlooks critical opportunities in message preprocessing optimization by LLMs, particularly regarding redundancy elimination and semantic compression of secret messages before embedding operations. Importantly, the low payload capacity necessitates long stego texts or multiple transmissions to maintain the integrity of the secret message; however, these behaviors increase the risk of detection by adversaries.

Thus, we propose *StegoZip*, the first plug-and-play framework designed to address the limitations in payload capacity via LLM-driven secret message preprocessing. The framework comprises two key components: information-driven dynamic semantic redundancy pruning (DSRP) and probability-driven index compression coding (ICC). By identifying high semantic redundancy in conventional message texts (Chen et al., 2024), DSRP leverages the semantic comprehension of LLMs to eliminate low-information elements dynamically to produce compressed content. For the receiver, a fine-tuned private restorer trained on public datasets reconstructs the original, semantically rich messages from their compressed forms. Moreover, building on Shannon's information theory (Shannon, 1951) and extending prior work that harnessed the predictive power of LLMs for compression (Valmeekam et al., 2023), we pioneer the adaptation of this theoretical foundation for steganographic message compression through the ICC. Since *StegoZip* only optimizes secret message preprocessing without altering the underlying steganographic algorithms, it preserves their inherent security. This architecture not only paves the way for more efficient message embedding but also ensures the preservation of semantic integrity.

Our main contributions are as follows:

- We identify communication risks arising from the low payload capacity in existing steganography, highlighting that their emphasis on embedding often neglects the crucial phase of secret message preprocessing optimization.
- We propose *StegoZip*, the first plug-and-play secret message preprocessing method designed to enhance the payload capacity independently of advanced steganography without compromising their security.
- We integrate *StegoZip* with current advanced linguistic steganography. The experimental results reveal that it increases the capacity by 2–3× and reduces the processing time per unit message by up to 50%.

2 Related Work

2.1 Generative Linguistic Steganography.

Linguistic steganography conceals secret messages within a text carrier. Traditional methods, e.g., Syndrome-Trellis Codes (STC) (Filler et al., 2011), and Steganographic Polar Codes (SPC) (Li et al., 2020), achieve this by modifying components of the cover text, often inducing statistical deviations from the natural distribution, rendering the stego text susceptible to detection by adversaries. In contrast, generative language modeling has revolutionized the field by offering novel avenues for embedding secret messages into generative data (Chen et al., 2018). These models are designed not only to learn underlying distributions but also to act as precise sampling mechanisms, producing content that is increasingly statistically indistinguishable from naturally occurring text, which provides a robust foundation for secure steganography.

Autoregressive language models, which dominate the field of text generation, operate by processing an initial prompt and iteratively sampling from an explicit probability distribution over tokens to generate text. Within this framework, secret messages can be incorporated into the token generation process without perturbing the intrinsic statistical properties of the output (Yang et al., 2018; Zhou et al., 2022). This embedding mechanism leverages the random sampling process, where non-overlapping subintervals of the unit interval are used to govern token selection, facilitating the en-

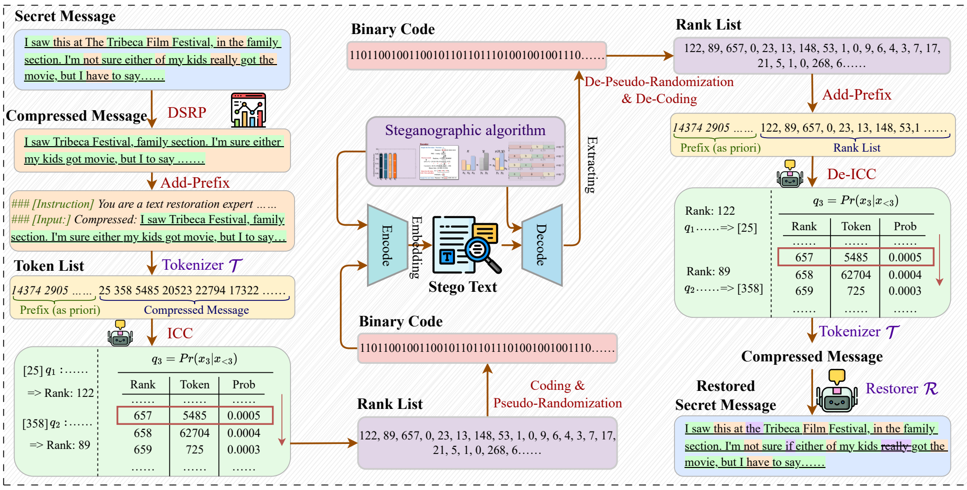

Figure 2: The framework of *StegoZip* comprises two core components: information-driven semantic pruning (DSRP) and probability-driven token-rank mapping (ICC). The extraction process mirrors their inverse operations.

coding of secret messages in a manner that preserves the overall distribution of the generated text.

Recent advances in provably secure linguistic steganography have capitalized on these principles. For example, ADG (Zhang et al., 2021) partitions the explicit probability distributions of generative models into groups of equal probability masses and encodes secret messages through group selection. Meteor (Kaptchuk et al., 2021) utilizes range reversible sampling to represent secret data within the shared prefixes of the sampled intervals. Discop (Ding et al., 2023) generates multiple "distribution copies" from a given probability distribution and uses the index values of these copies to denote the secret messages.

Despite these advancements, the inherent low entropy in the probability of text generation limits payload capacity. However, the emergence of large-scale models introduces significant opportunities not only during the embedding stage but also in the processing of secret messages. In light of this, we propose *StegoZip* to increase capacity via LLM-driven message processing and compression.

3 Method

3.1 Overview

As illustrated in Fig. 2, *StegoZip* comprises two LLM-driven components: Dynamic Semantic Redundancy Pruning (DSRP) and Index Compressed Coding (ICC). Initially, the framework leverages LLMs to systematically remove redundant elements via information-driven semantic pruning. Subsequently, the same LLM facilitates probability-driven index compression to generate rank sequences. These sequences are then subjected to binary encoding and cryptographic-grade pseudo-randomization, generating provably secure bit streams compatible with current steganographic systems. Finally, these bit streams are embedded into cover texts via a steganographic algorithm for secure transmission over public channels.

The authorized receiver, possessing prior knowledge of steganography, binary encoding schema, and cryptographic parameters, along with architecturally congruent LLM configurations, executes inverse transformation to decode the compressed indices. Following successful extraction, a shared semantic restorer $\mathcal{R}$ fine-tuned on public datasets reconstructs the rich semantic representation via context-aware. To ensure synchronization between the sender and receiver, the sender side also utilizes $\mathcal{R}$ as the LLM for DSRP and ICC aforementioned.

Throughout the steganographic process, the payload capacity optimization of *StegoZip* is independent of the steganographic embedding process and thus does not affect its security. The details of each module are as follows.

3.2 Private Restorer in StegoZip Framework.

First, we introduce the private restorer $\mathcal{R}$, a core component throughout the *StegoZip* framework obtained by fine-tuning a base language model as illustrated in Fig. 3. In everyday communication, rich semantics help the receiver fully understand the sender's point of view. However, such semantic redundancy can significantly increase the payload burden in public channel transmission and is impractical for scenarios with limited communication resources. Therefore, we leverage the powerful language comprehension capability of LLM to prune the semantics of secret messages and retain only the most critical parts for transmission. However, on the basis of human language perception alone, it may be difficult for the receiver to understand the semantic pruned secret messages, or ambiguity may arise. For this reason, we once again utilize the context-aware capability of the LLM to restore the rich semantics of the original message from the compressed version. Therefore, we fine-tune a private restorer $\mathcal{R}$ shared by both parties to accomplish these tasks. The implementation involves three key steps: Self-Information Calculation, Semantic Pruning, and Instruction Fine-tuning.

Self-Information Calculation. Initially, we must assemble the dataset for fine-tuning. The objective is to train the model to handle the task of semantic restoration effectively. To achieve this, the input should consist of text characterized by low semantic content, whereas the output should feature text with rich semantics. Considering a public text dataset $\mathcal{D}_p$, for each sample $x_p \in \mathcal{D}_p$ designated as output, we must quantize and compress its semantics to form the corresponding input. We employ the concept of self-information from information theory to quantify the information content of each lexical unit (the entity resulting from word tokenization, e.g., English sentences segmented by spaces) in x_p. This metric is facilitated by the base language model, and the self-information for a lexical unit u_j in the text sample x_p is defined as:

$$I_{\text{lex}}(u_j) = \sum_{i=1}^{k} \mathcal{I}\left(w_j^{(i)}\right) \tag{1}$$

where u_j represents the j-th lexical unit consisting of k tokens $\{w_j^{(1)}, ..., w_j^{(k)}\}$. Each lexical unit can be broken down into multiple tokens for processing by an LLM. For the t-th token w_t in the sequence, its self-information is defined as:

$$\mathcal{I}(w_t) = -\log P(w_t) = -\log p(w_t|w_{<t}) \tag{2}$$

where $p(w_t|w_{<t})$ represents the conditional probability given by the LLM. The more unlikely a token is to be sampled, the greater its self-information.

Semantic Pruning. After the self-information of all lexical units in x_p obtained, we remove the units with low information through α-ratio pruning:

$$\mathcal{D}_c = \{x_p \odot \mathbb{I}(I_{\text{lex}}(u_j) > \tau_\alpha) \mid \forall x_p \in \mathcal{D}_p\} \tag{3}$$

where $\odot$ denotes element-wise multiplication, $\mathbb{I}(\cdot)$ is the indicator function, and τ_α represents the α-quantile threshold satisfying:

$$P(I_{\text{lex}}(u_j) \le \tau_\alpha) = \alpha \tag{4}$$

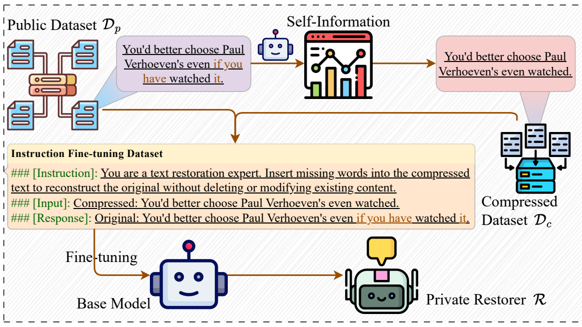

Figure 3: Process of the instruction fine-tuning.

Instruction Fine-tuning. After completing Semantic Pruning, we obtain the semantic compressed dataset $\mathcal{D}_c$. Then, we construct the instruction fine-tuning dataset $\mathcal{D}_{ft}$ via template-based pairing, as shown in Fig. 3:

$$\mathcal{D}_{\text{ft}} = \left\{ \left(\underbrace{x_{ins} \| x_c}_{\text{Input}}, \underbrace{x_p}_{\text{Output}} \right) \middle| \, x_p \in \mathcal{D}_p, x_c \in \mathcal{D}_c \right\} \tag{5}$$

where x_{ins} is the instruction and $\|$ denotes string concatenation. For the model to fully understand the semantic restoration task, we require it to restore the original rich semantics only by inserting words without deleting or modifying the existing content in the compressed text. Then, we fine-tune the base language model to obtain a private restorer $\mathcal{R}$ shared by both the sender and the receiver.

3.3 Dynamic Semantic Redundancy Pruning.

After establishing the shared private restorer $\mathcal{R}$, the *StegoZip* module can be integrated into current steganography. The dynamic pruning mechanism operates through two coordinated phases aligned with the restorer fine-tuning process: Self-Information Calculation and Semantic Pruning.

For the secret message m, we generate its compressed representation m_c via similar self-information processing by using $\mathcal{R}$:

$$m_c = m \odot \mathbb{I}\left(I_{\text{lex}}(u_j) > \tau'_\alpha\right) \tag{6}$$

To prevent excessive pruning in short texts with high entropy, we dynamically adjust the pruning threshold on the basis of the average self-information of m and the empirical value on the instruction fine-tuning dataset $\mathcal{D}_{\text{ft}}$:

$$\tau'_\alpha = \tau_\alpha \cdot \left(1 - \eta \cdot \frac{\bar{\mathcal{I}}(m) - \bar{\mathcal{I}}(\mathcal{D}_p)}{\bar{\mathcal{I}}(\mathcal{D}_p)}\right) \tag{7}$$

Algorithm 1: Index Compressed Coding.

Input: Compressed Message m_c, Tokenizer $\mathcal{T}$, Restorer $\mathcal{R}$, Huffman Codebook $\mathcal{C}$, Pseudo-Random Binary Key $\mathbf{K}$.
Output: Pseudo-Random Bit Stream $\mathbf{S}$.

$\mathbf{W}_c \leftarrow \mathcal{T}(m_c)$;
$\mathbf{B} \leftarrow \emptyset$;
foreach *token* $w_j \in \mathbf{W}_c = \{w_1, ..., w_k\}$ **do**
  $p\left(w_j^{|\mathcal{V}|} | w_{<j}\right) \leftarrow \mathcal{R}(\{w_1, ..., w_{<j}\})$;
  $r(w_j) \leftarrow \text{rank}\left(w_j \mid p\left(w_j^{|\mathcal{V}|} | w_{<j}\right)\right)$;
  $b_j \leftarrow \text{HuffmanEncode}(r(w_j), \mathcal{C})$;
  $\mathbf{B} \leftarrow \mathbf{B} \cup \{b_j\}$;
end
$\mathbf{S} \leftarrow \mathbf{B} \oplus \mathbf{K}$;
return $\mathbf{S}$;

Algorithm 2: Secret Message Restoration.

Input: Pseudo-Random Bit Stream $\mathbf{S}$, Tokenizer $\mathcal{T}$, Restorer $\mathcal{R}$, Huffman Codebook $\mathcal{C}$, Pseudo-Random Binary Key $\mathbf{K}$.
Output: Restored Secret message $\hat{m}$.

$\mathbf{B} \leftarrow \mathbf{S} \oplus \mathbf{K}$;
$\{r_1, \ldots, r_k\} \leftarrow \text{HuffmanDecode}(\mathbf{B}, \mathcal{C})$;
$\mathbf{W}_c \leftarrow \emptyset$;
foreach *rank* $r_j \in \{r_1, ..., r_k\}$ **do**
  $p\left(w_j^{|\mathcal{V}|} | w_{<j}\right) \leftarrow \mathcal{R}(\mathbf{W}_c)$;
  $w_j \leftarrow \text{de-rank}\left(r(w_j) \mid p\left(w_j | w_{<j}^{|\mathcal{V}|}\right)\right)$;
  $\mathbf{W}_c \leftarrow \mathbf{W}_c \cup \{w_j\}$;
end
$m_c \leftarrow \mathcal{T}(\mathbf{W}_c)$;
$\hat{m} \leftarrow \mathcal{R}(m_c)$;
return $\hat{m}$;

where τ_α is the predefined threshold from Eq. (4). $\bar{\mathcal{I}}(m)$ is the average self-information of secret message m and $\bar{\mathcal{I}}(\mathcal{D}_p)$ is the average self-information of $\bar{\mathcal{I}}(x_p)$ from the public dataset $\mathcal{D}_p$:

$$\bar{\mathcal{I}}(m) = \frac{1}{T}\sum_{t=1}^{T} \bar{\mathcal{I}}\left(w_t^{(m)}\right) = -\frac{1}{T}\sum_{t=1}^{T}\left(\log(w_t^{(m)} | w_{<t}^{(m)})\right) \tag{8}$$

$$\bar{\mathcal{I}}(\mathcal{D}_p) = \frac{1}{|\mathcal{D}_p|}\sum_{i=1}^{|\mathcal{D}_p|} \bar{\mathcal{I}}\left(x_p^{(i)}\right) \tag{9}$$

The values of self-information approaching infinity are not considered in the calculations. The ratio of units to be removed is dynamically determined based on the information of the secret message.

3.4 Index Compressed Coding

Following dynamic semantic pruning, we convert the compressed message m_c into binary codes through probability-driven index encoding. Inspired by (Valmeekam et al., 2023), our method leverages the token prediction prior of the LLM to achieve high compression ratios.

Let $\mathbf{W}_c = \{w_1, ..., w_k\}$ where each w_j represents a token in the tokenizerd sentence. We rank tokens by their conditional probabilities:

$$r(w_j) = \text{rank}\left(w_j | p\left(w_j^{|\mathcal{V}|} | w_{<j}\right)\right) \in \{1, ..., |\mathcal{V}|\} \tag{10}$$

where $|\mathcal{V}|$ is the vocabulary size of the LLM, and $p(w_j^{|\mathcal{V}|} | w_{<j})$ indicates the sampling probability when generating the j-th token. The probability-driven token-rank mapping enables efficient compression coding, where a higher sampling probability results in a lower rank. Given that our fine-tuned restorer $\mathcal{R}$ has been exposed to numerous instances of semantic pruning text, we employ it to deduce the ranks with the same template prefix, thereby achieving a higher compression rate. Furthermore, since the rank list is numerical, we convert these numbers into bit format $\mathbf{B}$ via Huffman encoding. Then, to align the provably secure steganography, we pseudo-randomize the $\mathbf{B}$ via XOR with pseudo-random binary key $\mathbf{K}$ generated by a secure stream encryption algorithm such as ChaCha20 (Bernstein et al., 2008). Finally, we can embed the resulting pseudo-random bit stream $\mathbf{S}$ into the cover text by secure steganographic embedding function. The whole process of the ICC is shown in Algo. 1.

3.5 Secret Message Restoration

The decoding framework enables message extraction and reconstruction through invertible transformations of the encoding pipeline, as formalized in Algo. 2. Given stego text x_s, the receiver first extracts the embedded bit stream $\mathbf{S}$ via the negotiated steganographic extraction function. Subsequently, the original numerical rank sequences can be obtained through de-pseudo-randomization with cryptographically synchronized parameters and Huffman decoding with the same codebook. Furthermore, the restorer $\mathcal{R}$ replicates the index compression coding generation process via inverse rank-token mapping, converting each rank into its corresponding token to reconstruct the semantic pruning message.

As the compressed representation may be insufficient for complete semantic comprehension through human perception, the shared restorer $\mathcal{R}$ performs instruction-guided semantic expansion. Crucially, the information-driven pruning method preserves high-information lexical units while discarding redundant elements lower than threshold

Table 1: The efficiency of *StegoZip*. As a plug-and-play module, *StegoZip* significantly improves the payload capacity while reducing the whole steganography processing time.

Base Model	Dataset	Algo	Payload (%) $\uparrow$	Encoding Time (s) $\downarrow$	Decoding Time (s) $\downarrow$
Qwen2.5-7B	IMDb	Meteor	9.13	115.93	117.46
		+*StegoZip*	29.73($\uparrow$ 20.60)	64.75($\downarrow$ 51.18)	92.00($\downarrow$ 25.46)
		Discop	15.14	100.73	100.39
		+*StegoZip*	49.52($\uparrow$ 34.38)	34.58($\downarrow$ 66.15)	60.82($\downarrow$ 39.57)
	AGNews	Meteor	8.59	32.02	33.24
		+*StegoZip*	33.24($\uparrow$ 24.65)	11.15($\downarrow$ 20.87)	14.44($\downarrow$ 18.80)
		Discop	14.29	16.97	15.64
		+*StegoZip*	55.17($\uparrow$ 30.88)	5.97($\downarrow$ 11.00)	9.86($\downarrow$ 5.78)
Vicuna-7B-v1.5	IMDb	Meteor	9.12	115.75	119.60
		+*StegoZip*	27.69($\uparrow$ 18.57)	69.54($\downarrow$ 46.21)	88.81($\downarrow$ 30.79)
		Discop	15.12	100.46	103.70
		+*StegoZip*	45.95($\uparrow$ 30.83)	37.04($\downarrow$ 63.42)	52.98($\downarrow$ 50.72)
	AGNews	Meteor	8.63	31.96	34.99
		+*StegoZip*	27.22($\uparrow$ 18.49)	13.58($\downarrow$ 18.38)	21.74($\downarrow$ 13.25)
		Discop	14.31	20.35	20.54
		+*StegoZip*	45.28($\uparrow$ 30.97)	7.26($\downarrow$ 13.09)	14.21($\downarrow$ 6.33)

τ_α, enabling the restorer to reconstruct original semantic content through maximum likelihood estimation. The larger the α is, the less information is retained, and the greater the error between the restored secret message and the original message.

4 Experiments

4.1 Implementation Details

LLMs. In this paper, we select two mainstream open-source LLMs, Qwen2.5-7B (Team, 2024) and Vicuna-7B-v1.5 (Touvron et al., 2023). For generation, random sampling is employed with a temperature setting of 0.7, without using top-p or top-k.

Datasets. Text datasets are used for fine-tuning the restorer and generating stego text. For fine-tuning, the IMDb (Maas et al., 2011) and AGNews (Maas et al., 2011) datasets are used. The IMDb dataset, with an average sample length of 1300 characters, is divided into a training set with 25,000 texts and a test set with 25,000 texts, but only 2,000 texts are randomly sampled for testing in each evaluation. Only the "business" category of the AGNews dataset with an average sample length of 241 characters is selected and divided into a training set containing 30,000 texts and a test set containing 1,900 texts. We use LoRA (Hu et al., 2021) to fine-tune the base LLMs for 2 epochs. To generate the stego text, the WikiText-2-v1 (Merity et al., 2016) dataset is used for the text generation task. In each instance, a text is randomly sampled from the dataset, and the first two sentences are extracted to serve as the prompt for guiding the generation.

Baselines. In the main experiment, the parameters for the proposed Dynamic Semantic Redundancy Pruning method are set to $\alpha = 0.3$ and $\eta = 1.0$. To the best of our knowledge, current linguistic steganography do not specifically consider message processing; thus, we adopt a common setup, using Huffman compression with UTF-8 encoding as the baseline message processing method. We consider mainstream methods for the underlying generative steganography: Meteor and Discop.

Evaluation metrics. We evaluate *StegoZip* from both efficiency and text quality perspectives:

1. Efficiency: We divide the efficiency into payload capacity and processing time. The payload capacity refers to the ratio of the secret message length to the stego text length, which is the most important metric for assessing StegoZip's compression capability. The processing time encompasses the average encoding time, which spans from processing the secret message to the completion of generating the stego text, and the average decoding time, which involves extracting the bit stream from the stego text and restoring the secret message.

2. Text Quality: We evaluate restored message quality at the word, sentence, and paragraph levels via metrics such as Rouge-1, Rouge-2, Rouge-ℓ (Lin, 2004), and Pairwise Similarity Percentages (P-SP). Higher values of these metrics are better. Rouge-1 and Rouge-2 calculate the proportion of single words (1-gram) and word pairs (2-grams) from the original secret message that appears in

Table 2: The efficiency of Restorer $\mathcal{R}$. Using the original secret message $\mathcal{D}_o$ as the reference text, the restored message $\mathcal{D}_r$ is much better than the compressed message $\mathcal{D}_c$ in word, sentence, and paragraph levels.

Model	Dataset	Ori-Gen	Rouge-1 (%) ↑	Rouge-2 (%) ↑	Rouge-ℓ (%) ↑	P-SP (%) ↑
Qwen2.5-7B	IMDb	$\mathcal{D}_o - \mathcal{D}_c$	74.17	53.48	74.17	93.34
		$\mathcal{D}_o - \mathcal{D}_r$	89.27(↑ 15.10)	74.54(↑ 21.06)	86.59(↑ 12.42)	95.79(↑ 2.45)
	AGNews	$\mathcal{D}_o - \mathcal{D}_c$	75.15	56.56	75.15	92.20
		$\mathcal{D}_o - \mathcal{D}_r$	89.96(↑ 14.81)	78.38(↑ 21.82)	88.98(↑ 13.83)	93.96(↑ 1.76)
Vicuna-7B-v1.5	IMDb	$\mathcal{D}_o - \mathcal{D}_c$	72.00	51.48	72.00	93.34
		$\mathcal{D}_o - \mathcal{D}_r$	93.55(↑ 21.55)	78.34(↑ 26.86)	87.13(↑ 15.13)	94.78(↑ 1.44)
	AGNews	$\mathcal{D}_o - \mathcal{D}_c$	72.54	52.69	72.54	82.78
		$\mathcal{D}_o - \mathcal{D}_r$	90.83(↑ 18.29)	76.59(↑ 23.90)	86.82(↑ 14.28)	90.93(↑ 8.15)

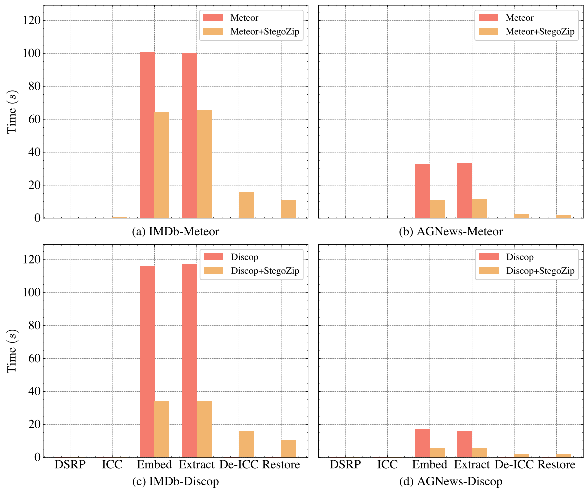

Figure 4: Time consumption of stego process.

the restored secret message, which better captures word order information. Rouge-ℓ calculates the proportion of the longest common subsequence in the original secret message that appears in the restored message, measuring semantic coherence. P-SP, which is based on a paraphraser model (Wieting et al., 2021), quantifies the semantic similarity between the original secret message and the restored message at the paragraph level.

All the experiments are run on a single RTX A6000 GPU. More detailed experimental settings are shown in the Appendix B.

4.2 Main Performance of StegoZip

Efficiency of StegoZip. The experimental results, shown in Tab. 1, demonstrate that the proposed plug-and-play *StegZip* significantly enhances the original steganography by achieving a **2–3× improvement in the secret message payload capacity**. This performance gain stems from semantic pruning and probability-driven index compression, which efficiently compresses lexical units in covert messages. Despite introducing additional preprocessing steps that are time consuming, as shown in Tab. 2 and Fig. 4, the optimized payload efficiency **reduces the steganographic embedding time and extraction time by approximately 50%**, as fewer binary codes are required to be embedded into cover text per unit message. Among the additional steps, rank decompression and semantic restoration operations are more time-consuming because of the necessity of implementing the generation-like process of new tokens through multiple forward propagation steps. In contrast, the semantic pruning and rank compression processes require only one forward propagation without generating a new token. Furthermore, the elevated payload capacity inherently strengthens security by minimizing the volume of stego text needed for transmission, thereby reducing adversaries' suspicion under equivalent communication requirements compared with those of the original methods. These advancements position *StegZip* as a practical solution for balancing capacity, time, and security in linguistic steganography systems.

Efficiency of the Restorer. We further assess the efficacy of the restoration model $\mathcal{R}$; the results, as depicted in Tab. 2, confirm the substantial restoration capabilities of $\mathcal{R}$ across various evaluation metrics. Both LLMs show significant enhancements in Rouge scores when comparing restored messages $\mathcal{D}_r$ with compressed messages $\mathcal{D}_c$, reflecting improved unigram overlap and paragraph-level coherence. While the P-SP index only increases marginally, it consistently exceeds 90%. This indicates that the DSRP, which employs low self-information pruning, effectively preserves the essential semantics of the original message and mitigates the pressure of redundancy, thereby reducing the burden on public channel transmission.

Generalization of Restorer. To assess the cross-domain generalization capabilities of *StegoZip*, we conduct tests on the fine-tuned Qwen-restorer $\mathcal{R}$

Table 3: The generalization of the Restorer $\mathcal{R}$.

Fine-Tuning Set	IMDb		AGNews	
Test Dataset	Payload↑	P-SP↑	Payload↑	P-SP↑
IMDb	**49.52**	**95.79**	47.51	93.15
AGNews	47.11	86.90	**55.17**	**93.96**

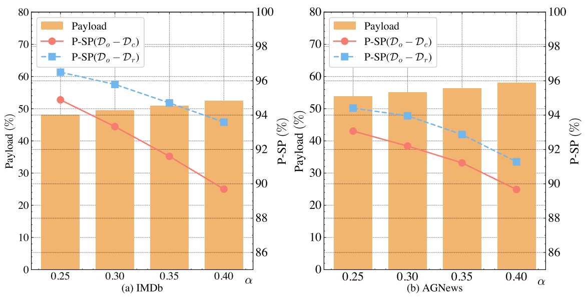

Figure 5: Impact of pruning threshold α.

using domain-shifted datasets, as detailed in Tab. 3. The results integrated on Discop indicate substantial performance drops in scenarios involving domain shifts. Specifically, a restorer trained on the IMDb dataset exhibited pronounced performance declines when tested on AGNews data, principally due to two critical distribution mismatches: 1) the stark contrast between IMDb's lengthy movie reviews and AGNews' succinct business articles in terms of textual complexity and 2) the domain-specific structural patterns prevalent in news articles as opposed to the subjective narrative styles found in movie reviews. These cross-domain variations impede the ability of *StegoZip* to accurately predict the next token in index compression coding and maintain semantic integrity during text restoration tasks. Thus, maintaining consistency in the steganographic environment during message transmission is crucial.

Pruning Threshold α. We also explore the influence of the self-information pruning ratio α, focusing on the payload capacity and semantic preservation of Discop. Within the range $\alpha \in [0.25, 0.40]$, an increase in payload capacity is observed, coinciding with degradation in both the original-compressed similarity P-SP($\mathcal{D}_o - \mathcal{D}_c$) and the original-restored similarity P-SP($\mathcal{D}_o - \mathcal{D}_r$), following a similar trend. This degradation occurs as a higher pruning proportion results in more succinct compressed data and consequently limits the restorer's contextual awareness, leading to incomplete information reconstruction. Therefore, it is important to balance the steganographic payload capacity and the faithful representation of the original message post-transmission.

Table 4: Ablation experiment on *StegoZip*.

Method	IMDb		AGNews	
	Payload↑	P-SP↑	Payload↑	P-SP↑
Discop+StegoZip	**49.52**	95.79	55.17	93.96
w/o η in DSRP	49.42	95.41	**55.37**	92.78
w/o DSRP	35.78	**100.00**	45.54	**100.00**
w/o Prefix in ICC	49.36	95.79	53.03	93.96
w/o $\mathcal{R}$ in ICC	47.28	95.79	47.31	93.96
w/o ICC	19.29	95.79	18.22	93.96
Discop	15.14	**100.00**	14.29	**100.00**

4.3 Ablation Experiment

We further perform ablation experiments with base model Qwen to prove the effectiveness of all the components of *StegoZip* as shown in Tab. 4, where "w/o" means not adopted. The ablation results highlighted substantiate the critical contributions of the individual components of our framework. In particular, the adaptive coefficient η in the dynamic semantic redundancy pruning (DSRP) module mitigates the risk of overcompression, especially in high-entropy short samples. Such overcompression cases will hinder semantic restoration even though compression may be more efficient. Moreover, incorporating a prompt prefix template during index compression—as opposed to its omission—enables the restorer, which has been fine-tuned to anticipate subsequent compressed content, to predict the next token more accurately, thereby enhancing compression efficiency. Overall, the experimental results affirm that each advancement within our framework not only bolsters the payload capacity for steganography but also ensures superior preservation of the embedded semantic message. More experimental results are shown in the Appendix C.

5 Conclusion

In this paper, we propose *StegoZip*, a play-and-plug framework that employs large language models for dynamic semantic redundancy pruning and index compression coding. By integrating it into advanced steganography, we realize a payload capacity increase of 2–3× and an approximately 50% reduction in per unit message processing time. These improvements not only increase the efficiency of the steganographic process but also reduce the frequency of communication between two parties, thereby decreasing the risk of detection. This method paves the way for efficient and secure message embedding, highlighting the potential of advanced preprocessing techniques to augment traditional steganographic frameworks.

Limitations

Despite extensive experimental validation of *StegoZip*'s superior performance, our work still has certain limitations: First, since *StegoZip* introduces the message preprocessing based on the LLM to increase the payload capacity of text steganography algorithms, it inevitably incurs additional preprocessing time and computational resource overhead even if the overall steganography time is shortened. Second, *StegoZip* does not perfectly restore the original secret information, as shown in Tab. 2, where the restored secret message is somewhat different from the original secret message at the word, sentence, and paragraph levels. If high accuracy of the secret message is required, the dynamic semantic redundancy pruning module can be omitted, and the probability-driven index compressed coding module alone can be utilized to more than double the payload capacity, as demonstrated in Tab 4. Finally, *StegoZip* requires access to the high-precision LLM for compressing and decompressing secret information, which makes it unsuitable for scenarios with limited computational resources (Bai et al., 2024).

Ethics Statement

In this paper, we propose the *StegoZip* framework to enhance the payload capacity of steganography, specifically for scientific research and educational purposes. We strictly adhere to established scientific research regulations to ensure data privacy and security throughout the experimental process, and we rigorously avoid any violation of personal privacy or engagement in illegal activities. We are committed to responsibly advancing academic research in information security and ensuring that our contributions positively impact society.

References

Minhao Bai, Jinshuai Yang, Kaiyi Pang, Xin Xu, Zhen Yang, and Yongfeng Huang. 2024. Provably robust and secure steganography in asymmetric resource scenario. *Preprint*, arXiv:2407.13499.

Daniel J Bernstein et al. 2008. Chacha, a variant of salsa20. In *Workshop record of SASC*, volume 8, pages 3–5. Citeseer.

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

Christian Cachin. 1998. An information-theoretic model for steganography. In *International Workshop on Information Hiding*, pages 306–318. Springer.

Shashikala Channalli and Ajay Jadhav. 2009. Steganography an art of hiding data. *arXiv preprint arXiv:0912.2319*.

Kejiang Chen, Hang Zhou, Hanqing Zhao, Dongdong Chen, Weiming Zhang, and Nenghai Yu. 2018. When provably secure steganography meets generative models. *arXiv preprint arXiv:1811.03732*, 1(3):4.

Kejiang Chen, Hang Zhou, Hanqing Zhao, Dongdong Chen, Weiming Zhang, and Nenghai Yu. 2021. Distribution-preserving steganography based on text-to-speech generative models. *IEEE Transactions on Dependable and Secure Computing*, 19(5):3343–3356.

Liang Chen, Yatao Bian, Yang Deng, Deng Cai, Shuaiyi Li, Peilin Zhao, and Kam-Fai Wong. 2024. Watme: Towards lossless watermarking through lexical redundancy. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9166–9180.

Christian Schroeder de Witt, Samuel Sokota, J Zico Kolter, Jakob Foerster, and Martin Strohmeier. 2022. Perfectly secure steganography using minimum entropy coupling. *arXiv preprint arXiv:2210.14889*.

Jinyang Ding, Kejiang Chen, Yaofei Wang, Na Zhao, Weiming Zhang, and Nenghai Yu. 2023. Discop: Provably secure steganography in practice based on" distribution copies". In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2238–2255. IEEE.

Tomáš Filler, Jan Judas, and Jessica Fridrich. 2011. Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Transactions on Information Forensics and Security*, 6(3):920–935.

Nicholas J Hopper, John Langford, and Luis Von Ahn. 2002. Provably secure steganography. In *Advances in Cryptology—CRYPTO 2002: 22nd Annual International Cryptology Conference Santa Barbara, California, USA, August 18–22, 2002 Proceedings 22*, pages 77–92. Springer.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Neil F Johnson and Sushil Jajodia. 1998. Steganalysis: The investigation of hidden information. In *1998 IEEE Information Technology Conference, Information Environment for the Future (Cat. No. 98EX228)*, pages 113–116. IEEE.

David Kahn. 1996. The history of steganography. In *International workshop on information hiding*, pages 1–5. Springer.

Gabriel Kaptchuk, Tushar M Jois, Matthew Green, and Aviel D Rubin. 2021. Meteor: Cryptographically secure steganography for realistic distributions. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1529–1548.

R Bala Krishnan, Prasanth Kumar Thandra, and M Sai Baba. 2017. An overview of text steganography. In *2017 fourth international conference on signal processing, communication and networking (ICSCN)*, pages 1–6. IEEE.

Weixiang Li, Weiming Zhang, Li Li, Hang Zhou, and Nenghai Yu. 2020. Designing near-optimal steganographic codes in practice based on polar codes. *IEEE Transactions on Communications*, 68(7):3948–3962.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Mohammed Abdul Majeed, Rossilawati Sulaiman, Zarina Shukur, and Mohammad Kamrul Hasan. 2021. A review on text steganography techniques. *Mathematics*, 9(21):2829.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Niels Provos and Peter Honeyman. 2003. Hide and seek: An introduction to steganography. *IEEE security & privacy*, 1(3):32–44.

Neha Rani and Jyoti Chaudhary. 2013. Text steganography techniques: A review. *International Journal of Engineering Trends and Technology (IJETT)*, 4(7):3013–3015.

Claude E Shannon. 1951. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Chandra Shekhara Kaushik Valmeekam, Krishna Narayanan, Dileep Kalathil, Jean-Francois Chamberland, and Srinivas Shakkottai. 2023. Llmzip: Lossless text compression using large language models. *arXiv preprint arXiv:2306.04050*.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Paraphrastic representations at scale. *arXiv preprint arXiv:2104.15114*.

Hanzhou Wu, Tianyu Yang, Xiaoyan Zheng, and Yurun Fang. 2023. Linguistic steganography and linguistic steganalysis. In *Adversarial Multimedia Forensics*, pages 163–190. Springer.

Kuan Yang, Kejiang Chen, Weiming Zhang, and Nenghai Yu. 2018. Provably secure generative steganography based on autoregressive model. In *International Workshop on Digital Watermarking*, pages 55–68. Springer.

Siyu Zhang, Zhongliang Yang, Jinshuai Yang, and Yongfeng Huang. 2021. Provably secure generative linguistic steganography. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3046–3055. Association for Computational Linguistics.

Ziwei Zhang, Juan Wen, Liting Gao, Wanli Peng, and Yiming Xue. 2025. Linguistic steganalysis based on few-shot adversarial training. *IEEE Transactions on Dependable and Secure Computing*, pages 1–15.

Xuejing Zhou, Wanli Peng, Boya Yang, Juan Wen, Yiming Xue, and Ping Zhong. 2022. Linguistic steganography based on adaptive probability distribution. *IEEE Transactions on Dependable and Secure Computing*, 19(5):2982–2997.

A More Related Work

A.1 Provably Secure Steganography.

Empirically secure steganography (e.g., STC (Filler et al., 2011) and SPC (Li et al., 2020)) inevitably allows an adversary to distinguish cover text from stego text with a non-negligible advantage. In contrast, provably secure steganography strives either to eliminate this advantage (i.e., achieve information-theoretic security (Cachin, 1998)) or to reduce it to a negligible level (i.e., attain computational security (Hopper et al., 2002)).

We define the cover channel, denoted by C_h, as the conditional probability distribution of cover signals C given the history h. Assuming the availability of a perfect sampler M that precisely follows the distribution C_h, we denote by $M_b^{C_h}$ the process that samples the next segment of cover output of length b. A steganographic system (or stegosystem) is defined as a triple of algorithms $(\mathsf{KGen}, \mathsf{Embed}, \mathsf{Extract})$, corresponding to key generation, embedding, and extraction, respectively. The embedding process takes as input a key k produced by:

$$k \leftarrow \mathsf{KGen}(\alpha), \tag{11}$$

a message m, and history x, and then uses the sampler M to produce an output sequence:

$$s_1 \,|\, s_2 \,|\, \cdots \,|\, s_s \leftarrow \mathsf{Embed}^M(k, m, x) \tag{12}$$

of length s. Similarly, the extraction process uses the same key k and history x to extract the secret message $\widetilde{m}$ from the stego sample:

$$\widetilde{m} \leftarrow \mathsf{Extract}^M\Big(k,\ \mathsf{Embed}^M(k, m, x),\ x\Big). \tag{13}$$

To formalize computational security, we consider a distinguishing game in which an adversary W attempts to differentiate between the cover distribution C and the stego distribution S. The adversary is challenged to distinguish between (i) samples produced by the secret-message-driven embedding Embed and (ii) samples generated by a normal random sampling procedure O that follows the cover distribution. The adversary's advantage is defined as:

$$\mathsf{Adv}^{\mathsf{SS}}_{C,S}(W) = \Bigg| \Pr_{k,\,M,\,\mathsf{Embed}}\Big[W^{M,\,\mathsf{Embed}(k,\cdot,\cdot)} = 1\Big] - \Pr_{M,O}\Big[W^{M,\,O(\cdot,\cdot)} = 1\Big] \Bigg|, \tag{14}$$

where the probability is taken over the randomness in k, M, Embed, and O. A stegosystem is considered computationally secure if, for every probabilistic polynomial-time adversary, this advantage is negligible in the security parameter α:

$$\mathsf{Adv}^{\mathsf{SS}}_{C,S}(W) < \mathsf{negl}(\alpha). \tag{15}$$

Accordingly, to ensure that the bitstream processed by *StegoZip* can be securely embedded into cover text using established provably secure steganography, the bitstream must first be pseudo-randomized. This is typically achieved by performing an XOR operation with a pseudo-random binary keystream generated by a secure stream encryption algorithm such as CHACHA20 (Bernstein et al., 2008).

B More Experiment Settings

B.1 Fine-tune

The fine-tuning experiment was configured with a random seed of 42, a micro-batch size of 2, and an overall batch size of 32, resulting in gradient accumulation steps computed as the batch size divided by the micro-batch size. The training ran for 2 epochs with a learning rate of 3e-4. The sequence length was capped at 1024 for the IMDb dataset and 512 for the AGNews dataset. The LoRA parameters were set to LORA_R = 8, LORA_ALPHA = 32, and a dropout rate of 0.05, and the targeted modules included {q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj}. Furthermore, the model was loaded in int8 precision and fine-tuned via FP16, with the training and validation split set to a ratio of 4:1.

Furthermore, to accurately identify the end position of the restorer's response, an "[END]" flag is appended to the conclusion of responses in the training set, and this "[END]" marker is also utilized as the termination signal during inference.

B.2 Model Inference

During the model inference phase, the model is loaded in the FP32 format because of the high-precision probabilistic sort of index compression coding. When the restorer is tasked with recovering high semantic information, random sampling is employed with a temperature parameter set to 0.7 without using top-p and top-k. The maximum number of newly generated tokens corresponds to the values used during training, with the IMDb dataset for long text capped at 1024 tokens and the AGNews dataset at 512 tokens.

B.3 Steganography

Meteor. We strictly followed the official open-source repository of Meteor[1], adopting the version with probability reordering to increase payload capacity, and integrated it into our codebase.

Discop. We strictly followed the official open-source repository of Discop[2], adopted the version with Huffman Tree to increase payload capacity, and integrated it into our codebase.

LLMs. To avoid the security impact of the fine-tuned LLM $\mathcal{R}$ on the steganography process, we do

[1] The repository of Meteor can be found at: https://gist.github.com/tusharjois/ec8603b711ff61e09167d8fef37c9b86

[2] The repository of Discop can be found at: https://github.com/comydream/Discop

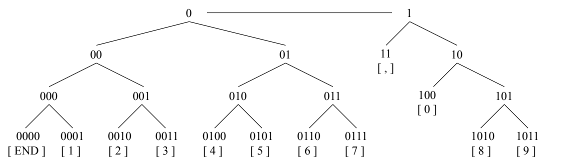

Figure 6: Huffman codebook.

not use it to generate stego text in our experiments but instead use the LLaMA2-7B (Touvron et al., 2023) model.

B.4 Huffman Codebook

Following the application of a probability-driven token-rank mapping, we obtain a sequence of numerical values, e.g., [0, 2, 1, 6, . . .]. This sequence is then required to be transformed into a binary format. Given the variable frequency of data and symbol occurrences during the interaction between two entities, we employ Huffman coding to convert this numerical array into a binary sequence, as depicted in Fig. 6. In this example, as the separator, the symbol " , " emerges as the most frequently occurring and is consequently assigned the shortest code length of 2 bits. The number "0" follows, receiving a code length of 3 bits due to being the second most frequent. The remaining digits display a similar frequency and are thus encoded with a uniform code length of 4 bits each.

It is typical for large language models not to terminate the generation process immediately after the complete embedding of covert messages, i.e., outputting a complete passage. To avoid drawing suspicion from external observers, the sender usually prolongs the generation until the text reaches a natural conclusion. To facilitate this, we introduce the additional "0000" encoding to denote the end of the secret message. This strategy ensures that the covert communication is seamlessly integrated within the overall message, thereby preserving the integrity of the cover.

This strategy enables the encoding of secret messages via an average code length that is more concise, and it allows both parties to establish a fixed codebook, thereby facilitating a more convenient and streamlined communication process.

For the Huffman compression algorithm utilized in the comparison method, we have employed the conventional technique of constructing Huffman trees and generating codebooks based on the character frequency. This method aims to enhance compression efficiency, thereby optimizing the payload capacity of the underlying steganography.

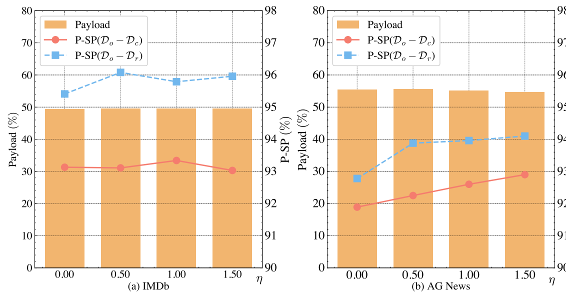

Figure 7: Impact of the adaptive coefficient η.

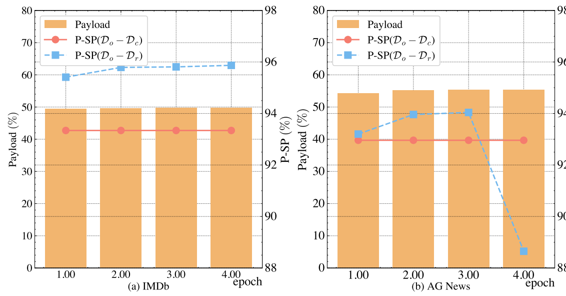

Figure 8: Impact of the fine-tuning epoch.

C More Experiment Results

C.1 Dynamic Adaptive Coefficient η.

The experimental results, as shown in Fig. 7, reveal the significant impact of the adaptive coefficient η on the *StegoZip* performance in the Dynamic Semantic Redundancy Pruning (DSRP) module. As the value of η increases, the model seeks a balance between compression efficiency and semantic resilience. Lower η values tend to favor higher compression efficiency but may lead to over-compression, which increases the risk of losing important semantic information and thus affects the restoration of semantic information. In contrast, higher η values are more conservative in compression and help protect semantic information but may sacrifice some compression efficiency. In addition, owing to the uneven entropy of the test samples, the model needs to adaptively adjust the η value to cope with the compression demand of different samples. This adaptive adjustment helps the model to flexibly balance the compression efficiency and semantic restoration ability when facing samples of different complexities and information densities, thus improving the performance and robustness of the *StegoZip* in general.

C.2 Epoches of Fine-tuning.

We further investigated the impact of the number of fine-tuning epochs for the restorer $\mathcal{R}$ on our experimental outcomes, as depicted in Fig.8. After a

P-SP	Original Message:	Compressed Message:	Restored Message:
100.0%	Mosaic Merger to Take Effect Today Polk County will retain its position at the heart of the US phosphate industry, at least through the end of this decade, following the merger of IMC Global Inc.	Mosaic Merger to Effect Today Polk will retain its position at heart of US phosphate industry, at least through end of this decade, following IMC Global Inc.	Mosaic Merger to Take Effect Today Polk County will retain its position at the heart of the US phosphate industry, at least through the end of this decade, following the merger of IMC Global Inc.
100.0%	Arthritis Drug Vioxx Pulled Off Market Sept. 30, 2004 -- Long-term use of the painkiller Vioxx doubles a person #39;s risk of heart attack and stroke, a huge clinical trial shows.	Arthritis Vioxx Pulled Off Sept. 30, Long-term use painkiller Vioxx doubles person #39;s risk of attack stroke, a huge clinical trial shows.	Arthritis Drug Vioxx Pulled Off Market Sept. 30, 2004 -- Long-term use of the painkiller Vioxx doubles a person #39;s risk of heart attack and stroke, a huge clinical trial shows.
95.86%	Time Warner settles with DOJ, SEC for \\$510 mil Time Warner Inc. on Wednesday settled criminal securities fraud charges the government leveled on its America Online unit, agreeing to pay \\$210 million to end the Justice Department #39;s probe.	settles with DOJ, SEC for \\$510 mil Warner Inc. on Wednesday criminal securities fraud charges leveled its America unit, agreeing pay \\$210 end Justice Department #39;s probe.	Time Warner settles with DOJ, SEC for \\$510 mil Time Warner Inc. on Wednesday settled criminal and securities fraud charges the government leveled at its America ~~Online~~ unit, agreeing to pay \\$210 million to end the Justice Department #39;s probe.
95.64%	Fed lifts rates a further quarter point By Andrew Balls in Washington and Jennifer Hughes in New York. The US Federal Reserve on Tuesday raised interest rates by a quarter point to 2.25 per cent and signalled there had been no change in its assessment of economic conditions.	Fed lifts rates a further quarter Andrew Balls Jennifer Hughes in New York. US Federal Reserve Tuesday raised interest rates by a quarter cent signalled had been no change its assessment of economic conditions.	Fed lifts rates a further quarter point By Andrew Balls ~~in Washington~~ and Jennifer Hughes in New York. The US Federal Reserve on Tuesday raised interest rates by a quarter percentage point to 2.25 per cent and signalled that there had been no change in its assessment of economic conditions.
90.68%	Cheap airfares help BAA profits Britain #39;s biggest airport operator BAA posted a 16 percent jump in first-half earnings on Tuesday, meeting expectations as cheap airfares and stronger economies drove up passenger numbers.	Cheap airfares help BAA profits #39;s biggest operator BAA posted percent jump first-half earnings Tuesday, meeting expectations as airfares and stronger economies drove up numbers.	Cheap airfares help BAA profits Britain #39;s biggest airport operator BAA PLC posted a 16 percent jump in first-half earnings ~~on~~ Tuesday, meeting expectations as ~~cheap~~ airfares and ~~strong~~ economies drove up passengers' numbers.
80.57%	Auto Parts Sector Falls on Delphi News Investors sold off shares of auto parts makers Friday after Delphi Corp. issued a profit warning and said it would cut nearly 5 percent of its work force next year.	Falls on Delphi News Investors sold shares of makers Friday after Delphi Corp. issued a profit warning said it would cut nearly 5 work force next year.	Blue Chips Fall on Delphi News Investors sold ~~off~~ shares of 194 stock makers Friday after Delphi Corp. issued a profit warning and said it would cut nearly 5 percent of its work force next year.

Figure 9: Some examples of secret message restoration results. In this case, the background color of green is the part that is preserved, yellow is the part that is pruned, and red is where the restored secret message differs from the original message. P-SP measures the similarity between the original secret message and the restored secret message.

single round of fine-tuning, the model essentially grasps the restoration task and can largely fulfill the requirements of the restoration process. Two rounds of fine-tuning are better for this task, effectively balancing training time and performance. Notably, an excessive number of fine-tuning rounds yields diminishing returns for the AGNews dataset, likely owing to the brevity of its sample texts. With limited training data, such datasets are more susceptible to overfitting, which can undermine the fine-tuning process.

C.3 Case Studies.

We present a comparative analysis between pruned secret messages and their reconstructed counterparts, accompanied by quantitative comparisons of semantic similarity (Pairwise Similarity Percentages, P-SP). Our case studies reveal that the semantic pruning algorithm predominantly targets and removes non-essential grammatical elements such as articles and prepositions. However, it inadvertently also eliminates vital context-dependent information, especially temporal, spatial, and quantitative references. This loss necessitates inference based on linguistic context or the parametric knowledge embedded within large language models (LLMs).

As evidenced by our experimental results shown in Fig. 9, the reconstruction fidelity exhibits a strong positive correlation with the P-SP. When this ratio exceeds 95%, the reconstructed messages maintain semantic equivalence with the original content. However, sub-optimal ratios below this threshold lead to progressive semantic degradation, primarily manifesting as irretrievable loss of specific named entities and numerical descriptors that lack sufficient contextual cues for LLM-based inference.

To address these limitations, we propose two complementary mitigation strategies:

1. **Operational protocol enhancement:** This method involves requiring human operators to manually reintroduce critical metadata tags during the compression phase. This step ensures that essential information, which might be overlooked by automated processes, is preserved.

2. **Algorithmic improvement:** We propose the development of context-aware lexical saliency metrics and use more powerful language models. These metrics are designed to more accurately capture the inferential dependencies of information elements, thus preventing the premature pruning of content that is semantically crucial.

Nonetheless, the restorer is instrumental in aiding the enhancement and refinement of the semantic content within the covert message.