Perfectly Secure Steganography Using Minimum Entropy Coupling

Anonymous Author(s)
Affiliation
Address
email

Abstract

Steganography is the practice of encoding a *plaintext* message into another piece of content, called a *stegotext*, in such a way that an adversary would not realize that hidden communication is occurring. This problem setting possesses two (competing) objectives: 1) To make the stegotext as similar as possible to the "normally" occuring content (known as *covertext*); 2) To encode as much information as possible about the content of the plaintext into the stegotext. Our first contribution is showing that any coupling procedure can be used to achieve perfectly secure steganography (assuming shared private keys for the communicating parties) against computationally unbounded passive adversaries. Our second contribution is proving that, among steganography procedures with perfect security, the one induced by minimum entropy coupling maximizes the amount of information transmitted over the channel. We combine this perspective with recent insights for approximate and iterative minimum entropy coupling to yield the first steganography procedure that can be scaled to arbitrary covertext distributions, without sacrificing security guarantees. Finally, as our third contribution, we empirically demonstrate that this procedure is able to encode plaintext messages into language and audio model covertext distributions with greater efficiency than alternative scalable approaches, despite having stricter security constraints.

1 Introduction

Modern applications, such as mobile-to-mobile communication or app-to-server communication, often require communicating sensitive information over insecure channels. Such applications motivate the development of methodologies for communicating over these channels while concealing sensitive content from adversarial third parties. Cryptographic procedures are one class of methods designed for this use case [Katz and Lindell, 2007, Chamberlain, 2017]. However, cryptographic procedures possess a drawback—they reveal to adversaries that sensitive information is being communicated, by virtue of the fact that encrypted messages (which appear as random content) are being sent over the channel. If an adversary controls the channel, it may simply block attempts to send encrypted messages, making cryptographic procedures inapplicable. Even if the adversary does not control the channel, it may engage in other undesirable activities, such as cyber-attacks.

A complementary approach to communicating sensitive information over insecure channels is steganography [Blum and Hopper, 2004, Cachin, 2004]. In steganography, the goal, informally speaking, is to encode a *plaintext* message in a manner that appears similar enough to innocuous communication (called *covertext*) that an adversary would not realize that hidden communication is occurring in the first place. Because steganographic procedures hide the existence of sensitive communication from adversaries altogether, they provide a complementary kind of security to that of cryptographic methods.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

In this work, we cast the problem of steganography as that of minimum entropy coupling (MEC). Given two marginal distributions for two random variables, the minimum entropy coupling is the joint distribution over these two random variables that has minimal joint entropy, subject to the constraint that it marginalizes correctly [Kovačević et al., 2015]. Our theoretical contribution proves that minimal entropy coupling between the covertext distribution and the ciphertext distribution (an encoded form of the plaintext that can be made to look uniformly random) yields a steganographic procedure that communicates the maximal possible amount of information about the plaintext message, subject to perfect steganographic security.

While minimum entropy coupling is an NP-hard problem, there exist $O(N \log N)$ approximation algorithms [Kocaoglu et al., 2017, Cicalese et al., 2019, Rossi, 2019] that are suboptimal (in terms of joint entropy) by no more than one bit, while retaining exact marginalization guarantees. Furthermore, Sokota et al. [2022] introduced an iterative minimum entropy coupling approach (iMEC) that iteratively applies these approximation procedures to construct couplings between one uniform distribution and one autoregressively specified distribution, both having arbitrarily large supports, while still retaining marginalization guarantees. Because ciphertext can be made to look uniformly random, and any distribution of covertext can be specified autoregressively, we can leverage iMEC to perform steganography for arbitrary covertext distributions and plaintext messages. *Excitingly, this yields the first instance of a steganography algorithm with perfect security guarantees that scales to arbitrary distributions of covertext.*

In our experiments, we evaluate iMEC using language and audio models—specifically GPT-2 and WaveRNN. We compare against arithmetic coding [Ziegler et al., 2019] and Meteor [Kaptchuk et al., 2021], other recent methods for performing steganography with deep generative models. To examine empirical security, we estimate the KL divergence between the stegotext and the covertext for each method. For iMEC, we find that the KL divergence is on the order of the numerical precision of float64, in agreement with our theoretical guarantees. In contrast, arithmetic coding and Meteor yield KL divergences many orders of magnitude larger, reflecting their weaker security guarantees. To examine encoding efficiency, we measure the number of bits transmitted per step. We find that iMEC yields superior efficiency results to those of arithmetic coding and Meteor, despite its stricter constraints.

2 Related Work

The concept of perfect security that we use in this work was first defined in [Cachin, 1998]. One case in which perfect security is possible is when the covertext distribution is uniform. In this case, perfect security can be achieved by embedding the message in uniform random ciphertext over the same domain as the covertext. However, constructing algorithms that both guarantee perfect security and transmit information at non-vanishing rates for more general covertext distributions has proved challenging. One notable result is that of Wang and Moulin [2008], who show that, in the case that letters of the covertext are independently identically distributed, public watermarking codes Moulin and O'Sullivan [2003], Somekh-Baruch and Merhav [2003, 2004] that preserve first order statistics can be used to construct perfectly secure steganography protocols of the same error rate. Another important line of research is that of Ryabko and Ryabko [2009], who show that perfectly secure steganography can be achieved in the case that letters of the covertext are independently identically distributed under the weaker assumption of black box access to the covertext distribution. In follow up work, Ryabko and Ryabko [2011] generalize their earlier work to a setting in which the letters of the covertext need only follow a k-order Markov distribution. Unlike these works, our approach does not make any assumptions on the structure of the distribution of covertext, though, unlike Ryabko and Ryabko, we do assume that this distribution is known.

There is also a body of related literature concerned with the combination of steganography and deep generative models. For example, Volkhonskiy et al. [2017] investigate the idea of training a generative adversarial network for steganography. In their setup, the generator is trained to be robust against both 1) a discriminator attempting to discriminate between real and generated images and 2) a discriminator attempting to discriminate between unmodified images and images for which a least significant bit matching has been used to embed a secret message. Another important example is the work of Dai and Cai [2019], who introduce an algorithm using Huffman coding to modify the covertext distribution in a manner that controls the total variation distance between the covertext distribution and the stegotext distribution. Perhaps the most closely related work is that of [Ziegler et al., 2019]. Ziegler et al. [2019] build on the work of Sallee [2003], who showed that compression

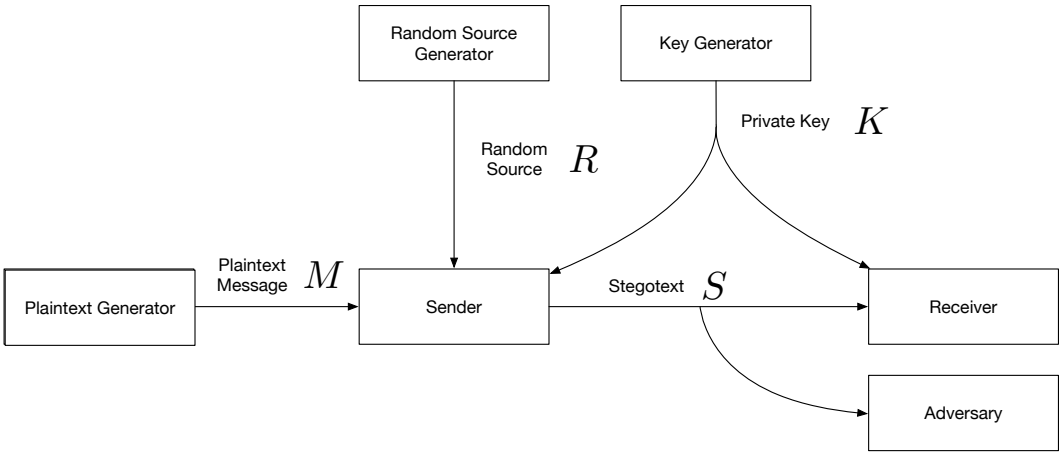

Figure 1: A graphical depiction of steganography. The sender receives a plaintext message, a source of randomness, and a private key, and outputs a stegotext. The receiver receives the same private key as the sender, along with the stegotext. The adversary also receives the stegotext.

algorithms can be used for steganography and that, in cases in which perfect compression is possible, the corresponding steganography procedure achieves perfect security. Ziegler et al. [2019] leverage this insight to perform experiments on language models, showing that arithmetic coding offers favorable trade-offs empirically compared to the Huffman coding. Most recently, Kaptchuk et al. [2021] also propose a steganography method called Meteor that resembles a modification of arithmetic coding. Kaptchuk et al. [2021] show empirical results in which Meteor outperforms arithmetic coding in terms of closeness to the covertext distribution, but is outperformed by arithmetic coding in terms of throughput. In contrast to these works, our approach possesses a provable perfect security guarantee.

3 Background

In this section, we review steganography and minimum entropy coupling, and describe the specific steganographic problem setting we consider in this work. For steganography, we provide both a high level summary of our problem setting in Section 3.1, and more precise technical descriptions in the successive sections.

3.1 Steganography

We consider a problem setting in which the distribution of normally occurring content $\mathcal{C}$, called the covertext distribution, is known to all parties (the sender, the receiver, and the adversary) and the adversary is unbounded but passive. Unbounded means that it may use arbitrarily expensive computational operations toward the end of determining whether the distribution of stegotext $\mathcal{S}$ (i.e., the distribution of text being sent by the sender), differs from the distribution of covertext; passive means that it is not allowed to modify the content sent by the sender. Our goal is to achieve so-called *perfect security* [Cachin, 1998], wherein the distribution of stegotext is exactly equal to the distribution of covertext (and, resultantly, even an unbounded passive adversary cannot distinguish between them), while simultaneously communicating as much information as possible to the receiver about the content of the plaintext message through the stegotext.

3.1.1 Problem Setting

The objects involved in steganography can be divided into two classes: those which are externally specified and those which require algorithmic specification. Each class contains three objects. The externally specified objects include the distribution over plaintext messages $\mathcal{M}$, the distribution over covertext $\mathcal{C}$, and the random source generator $\mathcal{R}$.

- The distribution over plaintext messages may be known by the adversary, but is not known by the sender or the receiver. However, the sender and receiver are aware of the domain $\mathbb{M}$ over which

$\mathcal{M}$ ranges. The realized plaintext message is explicitly made known to the sender, but not to the receiver or the adversary.

- The covertext distribution $\mathcal{C}$ is assumed to be known by the sender, the receiver, and the adversary.
- The random source generator $\mathcal{R}$ provides the sender with a mechanism to take random samples from distributions. This random source is known to the sender but not to the receiver or the adversary. As a result, randomness involved in the sender's encoding process cannot be exactly reproduced by the receiver or the adversary.

The objects requiring algorithmic specification, which are collectively referred to as a stegosystem, are the key generator $\mathcal{K}$, the encoder $\mathcal{E}$, and the decoder $\mathcal{D}$.

- The key generator $\mathcal{K}$ produces a private key K, whose realization is an element of $\{0,1\}^\lambda$ for some positive integer λ. This private key is shared between the sender and receiver over a secure channel prior to the start of the stegoprocess and can be used to coordinate communication. The key generation process $\mathcal{K}$ may be known to the adversary, but the realization of the key K is not.
- The encoder $\mathcal{E}$ takes a private key K, a plaintext message M, and a source of randomness R as input and produces a stegotext S in the space of covertexts $\mathbb{C}$.
- The decoder $\mathcal{D}$ takes a private key K and a stegotext S as input and returns an estimated plaintext message $\hat{M}$.

Many of the objects described above are depicted graphically in Figure 1.

3.1.2 Security

There are multiple ways to quantify the security level of a steganographic procedure. In this work, we are concerned with perfectly secure steganography.

Definition 3.1. *[Cachin, 1998] Given covertext distribution $\mathcal{C}$ and plaintext message space $\mathbb{M}$, a stegosystem $\langle \mathcal{K}, \mathcal{E}, \mathcal{D} \rangle$ is ϵ-secure against passive adversaries if the KL divergence between the distribution of covertext $\mathcal{C}$ and the distribution of stegotext $\mathcal{S}$ less than ϵ; i.e., KL$(\mathcal{C}, \mathcal{S}) < \epsilon$. It is perfectly secure if the KL divergence is zero; i.e., KL$(\mathcal{C}, \mathcal{S}) < 0$.*

In other words, a steganographic system is perfectly secure if the distribution of stegotext $\mathcal{S}$ communicated by the sender is exactly the same as the distribution of covertext $\mathcal{C}$.

3.1.3 Methodological Outline

One class of steganographic solution methods follows the following outline:

1. The sender and receiver use their shared private key K to inject the plaintext message space $\mathbb{M}$ into a space of binary sequences $\mathbb{X} = \{0,1\}^\ell$ called ciphertext. By using a random key, this injection can be done in such a way that the distribution over $\{0,1\}^\ell$ is uniformly random, regardless of the distribution of $\mathcal{M}$. (For example, one could generate K, uniformly at random, convert each m to binary $x = \text{bin}(m)$, and use the mapping $m \mapsto \text{bin}(m)$ XOR K.)
2. The sender uses an encoder $\{0,1\}^\ell \to \mathbb{C}$ to map the ciphertext X into stegotext (which exists in the space of covertexts).
3. The sender sends the stegotext S over the channel.
4. The receiver decodes the stegotext back into binary ciphertext.
5. The receiver decodes the binary back to the plaintext message space. (For the example above, the receiver can recover the binarized message $\text{bin}(m) = (\text{bin}(m) \text{ XOR } K) \text{ XOR } K$ using the shared private key, and invert the binary to recover the plaintext m.)

In the outline above, steps 1, 3, and 5 can be accomplished using standard operations in steganography literature and, thus, are left implicit in much of the remainder of the paper. Our methodological contribution specifically concerns steps 2 and 4.

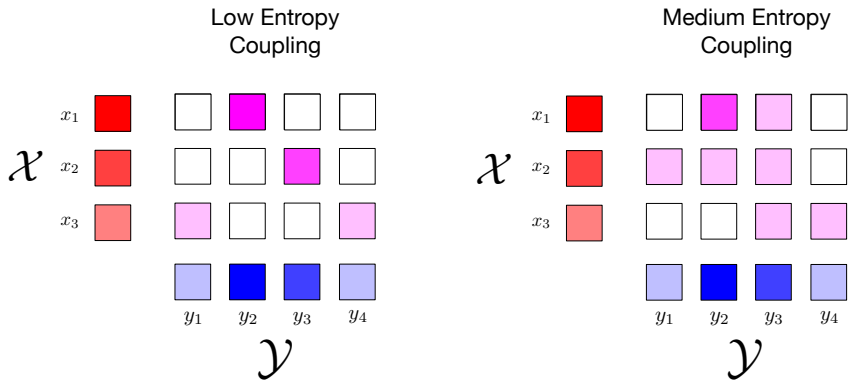

Figure 2: Two example couplings, shown in magenta, between $\mathcal{X}$ (red) and $\mathcal{Y}$ (blue). The left coupling has lower entropy than the right coupling (shading reflects the probability mass).

3.2 Minimum Entropy Coupling

Let $\mathcal{X}$ and $\mathcal{Y}$ be probability distributions over finite sets $\mathbb{X}$ and $\mathbb{Y}$. A coupling γ of $\mathcal{X}$ and $\mathcal{Y}$ is a joint distribution over $\mathbb{X} \times \mathbb{Y}$ such that, for all $x \in \mathbb{X}, \sum_{y'} \gamma(x, y') = \mathcal{X}(x)$ and such that, for all $y \in \mathbb{Y}, \sum_{x'} \gamma(x', y) = \mathcal{Y}(y)$. In other words, a coupling is a joint distribution over $\mathbb{X}$ and $\mathbb{Y}$ that marginalizes to $\mathcal{X}$ and $\mathcal{Y}$, respectively. In general, there may be many possible couplings for distributions $\mathcal{X}$ and $\mathcal{Y}$. (As an example, Figure 2 visually depicts two possible couplings for the same marginal distributions.) Let $\Gamma(\mathcal{X}, \mathcal{Y})$ denote the set of all couplings. The goal of minimum entropy coupling (MEC) is to find the element of $\Gamma(\mathcal{X}, \mathcal{Y})$ with minimal entropy. In other words, to find $\gamma \in \Gamma(\mathcal{X}, \mathcal{Y})$ such that the entropy $\mathcal{H}(\gamma) = -\sum_{x,y} \gamma(x, y) \log \gamma(x, y)$ is no larger than that of any other coupling in $\Gamma(\mathcal{X}, \mathcal{Y})$.

In general, computing the MEC is an NP-hard problem. That said, there has been substantial recent progress in approximating MECs. Cicalese et al. [2019], Rossi [2019] recently showed that it is possible to approximate MECs in $N \log N$ time with a solution guaranteed to be suboptimal by no more than one bit. Even more recently, Sokota et al. [2022] introduced an iterative minimum entropy coupling approach (iMEC) that uses an approximate MEC algorithm as a subroutine to couple distributions with arbitrarily large supports, so long as one of the distributions is uniform and the other can be specified autoregressively. This approach provably produces couplings, meaning that exact marginalization to the inputs is guaranteed, regardless of the input distributions. Because this approach, which we call iterative minimum entropy coupling (iMEC), is central to our experiments, we describe it in further detail below.

3.2.1 Iterative Minimum Entropy Coupling

Assume that $\mathcal{X}$ is a uniform distribution and let $\mathbb{X}_1 \times \cdots \times \mathbb{X}_n = \mathbb{X}$ and $\mathbb{Y}_1 \times \cdots \times \mathbb{Y}_m = \mathbb{Y}$ be factorizations over the spaces that $\mathcal{X}$ and $\mathcal{Y}$ range. iMEC implicitly defines a coupling γ between $\mathcal{X}$ and $\mathcal{Y}$ using procedures that iteratively call an approximate MEC as a subroutine [Sokota et al., 2022]. These procedures can sample $\gamma(Y \mid x)$ and query $\gamma(X \mid y)$ for a given x and y respectively. To align with steganography terminology, we will call these operations encoding and decoding. Because these procedures share a similar structure, we describe them as a unified operation as follows:

1. Initialize a uniform distribution μ_i over $\mathbb{X}_i$ for each $i = 1, \ldots, m$.
2. Iterate $j = 1, \ldots, m$.
   (a) Select $i^* = \max_i \mathcal{H}(\mu_i)$ to be the index of block whose distribution has maximal entropy.
   (b) Call the approximate MEC subroutine between μ_{i^*} and $\mathcal{Q}(Y_j \mid y_{1:j-1})$. Denote this coupling as ν. If performing encoding, set $y_j = Y_j \sim \nu(Y_j \mid x_{i^*})$; if performing decoding, y_j is known and no sampling is required. Update μ_{i^*} to be equal to $\nu(X_{i^*} \mid y_j)$.
3. If performing encoding, return y; if performing decoding, return $\gamma(X \mid y) \colon x \mapsto \prod_i \mu_i(x_i)$.

4 Steganography as Minimum Entropy Coupling

Having introduced steganography and minimum entropy coupling, we are ready to explain our contribution. We describe how steganography can be handled as a coupling problem as follows:

Steganography as Coupling

1. Let $\gamma \in \Gamma(\mathcal{X}, \mathcal{C})$ be a coupling over ciphertext distribution $\mathcal{X}$ and covertext distribution $\mathcal{C}$.
2. Given ciphertext x, let the sender communicate stegotext $S \sim \gamma(C \mid X = x)$.
3. Given stegotext S, let the receiver estimate ciphertext $\arg\max_{x'} \gamma(X = x' \mid C = S)$.

Viewing steganography as a coupling problem has value because of its strong security guarantees.

Proposition 1. *Steganography as coupling has perfect steganographic security.*

Proof. Consider that the distribution of stegotext $\mathcal{S}$ is dictated by $\gamma(C \mid X = x)$ for a given ciphertext x. Thus the marginal distribution of stegotext is given by $\mathbb{E}_{X \sim \mathcal{X}} \gamma(C \mid X) = \gamma(C)$. By definition of a coupling, we have $\gamma(C) = \mathcal{C}$. Therefore, we have $\mathcal{S} = \mathcal{C}$ and $\mathrm{KL}(\mathcal{C}, \mathcal{S}) = 0$. □

Viewing steganography, in particular, as a minimum entropy coupling problem, has value because of its implications on information throughput.

Proposition 2. *Performing steganography as coupling with a minimum entropy coupling procedure maximizes the mutual information $\mathcal{I}(M; S)$ between the plaintext message and the stegotext, subject to the constraint of perfect steganographic security.*

Proof. Consider that $\mathcal{I}(M; S) = \mathcal{H}(M) + \mathcal{H}(S) - \mathcal{H}(M, S)$. Now, note that the distribution of M is externally specified and, therefore, $\mathcal{H}(M)$ cannot be optimized. Next, recall that perfect steganographic security implies $\mathcal{S} = \mathcal{C}$. Thus, the distribution of S is externally specified, implying $\mathcal{H}(S)$ cannot be optimized. Finally, recall that our ciphertext encoding is injective and ranges over a discrete distribution. Therefore, $\mathcal{H}(M, S) = \mathcal{H}(X, S)$. Noting that $\mathcal{H}(X, S)$ is exactly the quantity being minimized by minimum entropy coupling yields our result. □

We also observe that, given the sender's procedure, the receiver's behavior is rational.

Remark 1. *Given the sender's encoding procedure, the receiver's decoding procedure minimizes its error rate.*

Proof. Because the sender's encoding process is dictated by the joint distribution γ, the posterior over ciphertexts is $\gamma(X = x \mid C = S)$. Therefore, the error rate is $1 - \gamma(X = \hat{x} \mid C = S)$, which is minimized by $\hat{x} = \arg\max_{x'} \gamma(X = x' \mid C = S)$. □

In light of of these results, we suggest that viewing steganography as a minimum entropy coupling problem is a natural and fundamental perspective. Indeed, we have proven that, among steganography procedures with perfect security, the one induced by minimum entropy provably maximizes the amount of information transmitted by the sender. Furthermore, our insight is easily extended beyond our theoretical results. Because, as discussed in the background, it is always possible to make ciphertext look uniformly random, iMEC [Sokota et al., 2022] can immediately be plugged into the steganography as coupling framework for arbitrary covertext distributions. And while iMEC does not possess proven approximation guarantees as a minimum entropy coupling algorithm, it yields performant couplings in large scale settings, as we will see in the experiments.

5 Experiments

We empirically compare iMEC against arithmetic coding [Ziegler et al., 2019] and Meteor [Kaptchuk et al., 2021] on four different covertext types. We also include a variant of Meteor that employs bin-sorted probabilities [Kaptchuk et al., 2021, Meteor:reorder].

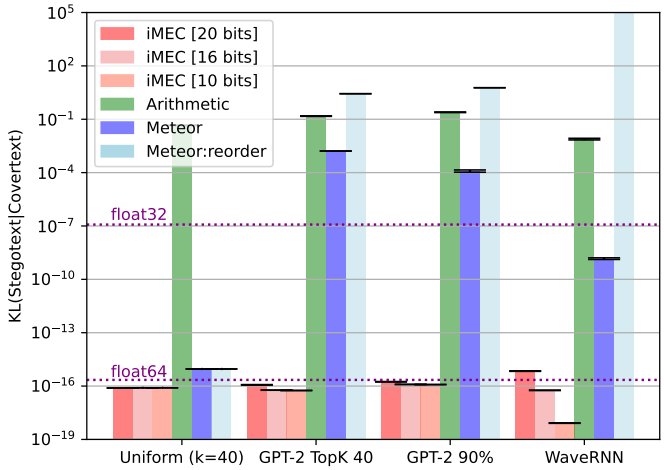

Figure 3: Kullback-Leibler divergences between the stegotext distribution and covertext distribution for each method. Errorbars indicate estimates of 95% confidence intervals computed from the central limit theorem over 1000 runs.

5.1 Experiment Setup

Our first covertext distribution consists of uniformly random noise (*UNIF*) of dimension 40 and a mean channel entropy of $\overline{\mathcal{H}}_C = 5.32$ bits. The second and third covertext distributions are variants of GPT-2 [Radford et al., 2019] with 12 attention modules [Wolf et al., 2020] conditioned on 1024-character strings from the *Wikitext-103* dataset [Merity et al., 2016]. The second covertext distribution performs *top-k* sampling from a re-normalised categorical distribution over the 40 highest-probability outputs. The third covertext distribution instead performs *nucleus sampling* [Holtzman et al., 2020] from the highest-probability outputs that together comprise 90% of the raw channel entropy [Radford et al., 2019]. The fourth covertext distributions consists of a *text-to-speech (TTS) pipeline* [Yang et al., 2022] based on Tacotron-2 [Shen et al., 2018] mel-spectogram encodings, followed by a WaveRNN generative model [Kalchbrenner et al., 2018] pretrained on the LJSpeech dataset [Ito and Johnson, 2017] with an average channel entropy of $\overline{\mathcal{H}}_C = 0.35$ bits. We condition the WaveRNN model using text generated by GPT-2 that we convert into audio data using mel-spectogram encodings.

Our implementation of iMEC makes use of the approximative minimum entropy coupling heuristic suggested by [Kocaoglu et al., 2017, Algorithm 1] as its subprocedure. We also integrated the original implementations of arithmetic coding Ziegler et al. [2019] and Meteor/Meteor:reorder Kaptchuk et al. [2021] into our codebase.

All experiments were performed on a AMD Ryzen Threadripper PRO 3955WX with 16 physical cores and 2x NVIDIA GeForce RTX 3090 GPUs. Note that, apart from neural model forwards, iMEC encoding and decoding occupies just a single CPU core, while both algorithmic coding and Meteor make use of multiple CPUs, and Meteor:reorder also makes use of GPUs during encoding and decoding.

All ciphertexts are 80-bit bitstrings sampled uniformly at random. We measure the encoding efficiency of each method by measuring the amount of covertext required to transmit these bitstrings. We tuned the hyper-parameters of each method to yield the best performance on this task. For iMEC, we stop transmission when the entropy of every posterior block is less than 0.1, i.e., $\forall i, \mathcal{H}(\mu_i) < 0.1$. We found the error rate for this threshold to be lower than 1 in 10^6 bits (see Appendix for further discussion). For arithmetic coding and Meteor/Meteor:reorder, we performed a grid search over precision values of $8, 16, 32$ and 48 bits. We find that both arithmetic coding and Meteor get stuck in a (seemingly) infinite loop at a frequency of about 3% (see Appendix for further discussion). For simplicity, we simply discard these runs from our investigation. This may result in slight overestimation of these algorithms' encoding efficiencies.

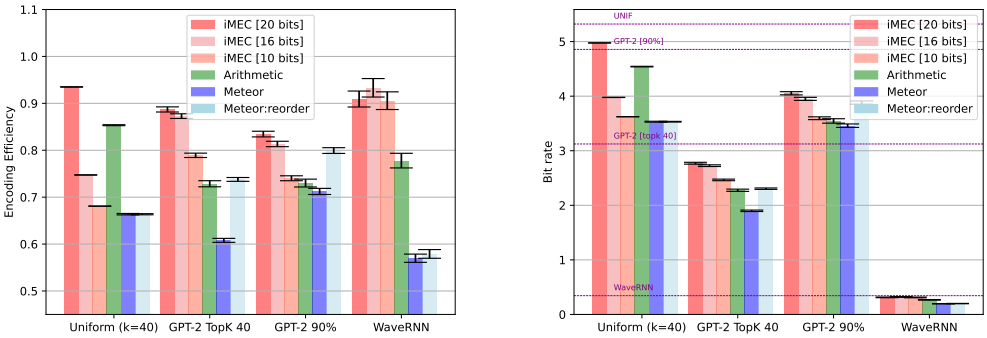

Figure 4: Comparative encoding efficiencies (left) and bit rates (right) of our method and baselines. Each method is evaluated at tuned hyperparameters. Errorbars indicate estimates of 95% confidence intervals computed using the central limit theorem from 1000 runs. Horizontal lines on the right plot correspond to the mean entropy of the channel.

5.2 Results

Empirical Security We first investigate the empirical security of each method. As estimating $\mathrm{KL}(\mathcal{C}, \mathcal{S})$ is difficult, we instead show $\mathrm{KL}(\mathcal{S}, \mathcal{C})$, which can easily be estimated from samples. We present the results in Figure 3. As expected from theory, we find that iMEC's security is on order of the precision of the numerical data type employed in its implementation—rendering it ϵ-*secure* for arbitrarily small ϵ in practice. On the other hand, arithmetic coding's divergence is consistently many orders of magnitude above the precision of the data type. For Meteor and Meteor:reorder, we observe good security for a uniform distribution. But, for more realistic distributions, they yield divergences many orders of magnitude above the precision of the data type. That said, consistent with Kaptchuk et al.'s findings, we observe that Meteor does result in better security than arithmetic coding. On the other hand, for Meteor:reorder, we see poorer security than arithmetic coding for realistic distributions, and even observe a divergent KL divergence for WaveRNN.

Encoding Efficiency We next compare encoding efficiencies and bit rates. We say bit rate to mean the number of bits encoded per token. We say encoding efficiency to mean the bit rate normalized to the average channel entropy. We show the results for both in Figure 4. We present results for iMEC with varying block sizes $\{|\mathbb{X}_i|\}_i$ over the ciphertext. These block sizes are hyperparameters for iMEC and can be made larger or smaller. We find that iMEC with a block size of 20 bits yields superior encoding efficiency to the baselines across all three of our realistic distributions. We also observe that iMEC with a block size of 16 bits generally yields competitive or superior performance to the other methods, and that 10 bit block sizes also performs competitively. Impressively, at a standard encoding frequency of 24kHz, iMEC's performance for WaveRNN would allow it to encode a private message of 225 kilobytes in just 30 seconds of TTS voicemail—sufficient for sending compressed images. Among the baselines, we observe that arithmetic coding tends to produce higher efficiency than Meteor, again consistent with Kaptchuk et al.'s findings.

Speed Lastly, we examine the speed of each algorithm. We show results in Figure 5. While, in the previous section, we observed that increasing iMEC's block size generally improves encoding efficiency, we see here that this improved efficiency does not come without cost. While 10 bit blocks require an order of magnitude less time than model inference, 16 bit blocks require the same order of magnitude of time as model inference, and 20 bit blocks require an order of magnitude more time than that. The wall-clock time of arithmetic coding and Meteor are generally comparable to that of the 10 bit blocks (though, as noted in the experimental setup, they also use more computational resources than iMEC), while the wall-clock of Meteor:reorder varies somewhat significantly depending on the task. We believe it is possible (perhaps even likely) that innovations in approximate minimum entropy coupling will allow some of the cost of coupling to be distributed across multiple cores, making the block sizes we experiment with here much cheaper.

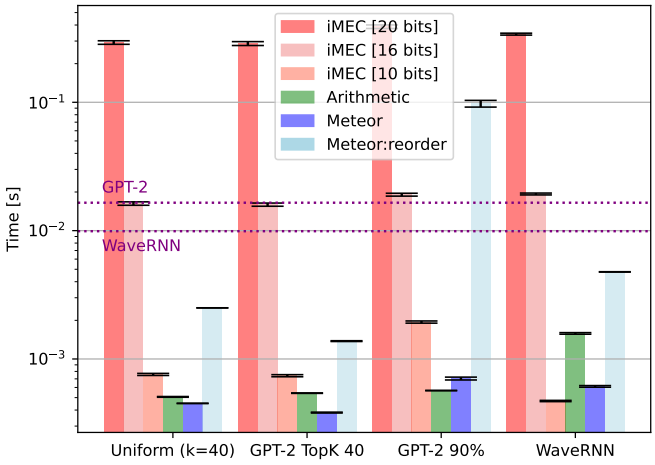

Figure 5: Comparative speed evaluation from the cover distribution of our method and baselines. Each method is evaluated at optimal hyperparameters. Errorbars indicate estimates of 95% confidence intervals computed using the central limit theorem from 1000 runs. Horizontal lines indicate the amount of time required for model inference for GPT-2 and WaveRNN.

6 Conclusion and Future Work

In this work, we showed how steganography can be approached as a minimum entropy coupling problem. First, we proved that coupling algorithms yield steganography procedures with perfect security; second, we proved that, among steganography algorithms with perfect security, the one induced by minimum entropy coupling maximizes information throughput. In aggregate, we believe that these findings suggest that the steganography problem setting we consider may be viewed most naturally through the lens of minimum entropy coupling. Furthermore, we show that this insight is also practical. Using recent innovations in approximate and iterative minimum entropy coupling [Kocaoglu et al., 2017, Sokota et al., 2022], we showed how this perspective can be used to perform steganography with deep generative model covertext distributions. In empirical evaluations, we show that iterative minimum entropy coupling is perfectly secure in practice, up to numerical precision, and exhibits superior efficiency compared to existing methods.

The most significant limitations of our work arise from the problem setting we consider. First, we assume that the adversary is passive, and that the stegotext arrives to the receiver unperturbed. While this assumption is standard among related work and may hold for many digital transmission channels, it is unrealistic in other settings. One direction for future work is to extend the minimum entropy coupling perspective of steganography to settings in which the channel medium is noisy. We believe that this may be possible by taking inspiration from ideas in error control literature. Second, we assume that white box access to the covertext distribution is available (also a standard assumption among related work). Unfortunately, even modern deep generative models struggle to exactly capture complex distributions (though it is expected that this issue will be somewhat ameliorate over time, due to ongoing research efforts in the deep learning community). Furthermore, in realistic scenarios, the distribution of "normally" occurring content may shift over time and depend on other external context, making it difficult to capture. Dropping this second assumption appears more challenging the first one, suggesting that a minimum entropy coupling perspective on steganography may be better suited to settings in which the covertext distribution can be modeled accurately.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Conclusion and Future Work.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

References

M. Blum and N. Hopper. Toward a theory of steganography. 2004.

C. Cachin. An information-theoretic model for steganography. In D. Aucsmith, editor, *Information Hiding*, pages 306–318, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-49380-8.

C. Cachin. Digital steganography, 2004.

A. Chamberlain. Applications of cryptography, Mar 2017. URL `https://blogs.ucl.ac.uk/infosec/2017/03/12/applications-of-cryptography/`.

F. Cicalese, L. Gargano, and U. Vaccaro. Minimum-entropy couplings and their applications. *IEEE Transactions on Information Theory*, 65:3436–3451, 2019.

F. Dai and Z. Cai. Towards near-imperceptible steganographic text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4303–4308, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1422. URL https://aclanthology.org/P19-1422.

A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

K. Ito and L. Johnson. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/, 2017.

N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu. Efficient Neural Audio Synthesis. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2410–2419. PMLR, July 2018. URL https://proceedings.mlr.press/v80/kalchbrenner18a.html. ISSN: 2640-3498.

G. Kaptchuk, T. M. Jois, M. Green, and A. D. Rubin. Meteor: Cryptographically secure steganography for realistic distributions. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 1529–1548, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384544. doi: 10.1145/3460120.3484550. URL https://doi.org/10.1145/3460120.3484550.

J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007. ISBN 978-1-58488-551-1.

M. Kocaoglu, A. Dimakis, S. Vishwanath, and B. Hassibi. Entropic Causal Inference. In *AAAI*, 2017.

M. Kovačević, I. Stanojević, and V. Šenk. On the entropy of couplings. *Information and Computation*, 242:369–382, 2015. ISSN 0890-5401. doi: https://doi.org/10.1016/j.ic.2015.04.003. URL https://www.sciencedirect.com/science/article/pii/S0890540115000450.

S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer Sentinel Mixture Models. *CoRR*, abs/1609.07843, 2016. URL http://arxiv.org/abs/1609.07843. arXiv: 1609.07843.

P. Moulin and J. O'Sullivan. Information-theoretic analysis of information hiding. *IEEE Transactions on Information Theory*, 49(3):563–593, 2003. doi: 10.1109/TIT.2002.808134.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language Models are Unsupervised Multitask Learners. *undefined*, 2019. URL https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe.

M. Rossi. Greedy additive approximation algorithms for minimum-entropy coupling problem. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 1127–1131, 2019. doi: 10.1109/ISIT.2019.8849717.

B. Ryabko and D. Ryabko. Asymptotically optimal perfect steganographic systems. *Problems of Information Transmission*, 45:184–190, 06 2009. doi: 10.1134/S0032946009020094.

B. Ryabko and D. Ryabko. Constructing perfect steganographic systems. *Information and Computation*, 209(9):1223–1230, 2011. ISSN 0890-5401. doi: https://doi.org/10.1016/j.ic.2011.06.004. URL https://www.sciencedirect.com/science/article/pii/S0890540111001064.

P. Sallee. Model-based steganography. volume 2939, pages 154–167, 10 2003. ISBN 978-3-540-21061-0. doi: 10.1007/978-3-540-24624-4_12.

J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu. Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783, Apr. 2018. doi: 10.1109/ICASSP.2018.8461368. ISSN: 2379-190X.

S. Sokota, C. S. de Witt, M. Igl, L. M. Zintgraf, P. Torr, M. Strohmeier, J. Z. Kolter, S. Whiteson, and J. N. Foerster. Communicating via markov decision processes. In *Proceedings of the 39th International Conference on Machine Learning*, ICML'22. JMLR.org, 2022.

A. Somekh-Baruch and N. Merhav. On the error exponent and capacity games of private watermarking systems. *IEEE Transactions on Information Theory*, 49(3):537–562, 2003. doi: 10.1109/TIT.2002.808132.

A. Somekh-Baruch and N. Merhav. On the capacity game of public watermarking systems. *IEEE Transactions on Information Theory*, 50(3):511–524, 2004. doi: 10.1109/TIT.2004.824920.

D. Volkhonskiy, I. Nazarov, B. Borisenko, and E. Burnaev. Steganographic generative adversarial networks. *Proceedings of NIPS 2016 Workshop on Adversarial Training*, 03 2017.

Y. Wang and P. Moulin. Perfectly secure steganography: Capacity, error exponents, and code constructions. *IEEE Transactions on Information Theory*, 54(6):2706–2722, 2008. doi: 10.1109/TIT.2008.921684.

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. HuggingFace's Transformers: State-of-the-art Natural Language Processing. Technical Report arXiv:1910.03771, arXiv, July 2020. URL `http://arxiv.org/abs/1910.03771`. arXiv:1910.03771 [cs] type: article.

Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi. TorchAudio: Building Blocks for Audio and Speech Processing. Technical Report arXiv:2110.15018, arXiv, Feb. 2022. URL `http://arxiv.org/abs/2110.15018`. arXiv:2110.15018 [cs, eess] type: article.

Z. Ziegler, Y. Deng, and A. Rush. Neural linguistic steganography. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1210–1215, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1115. URL `https://aclanthology.org/D19-1115`.

A Appendix

In the original submission, we made a clerical error in the caption of Figure 3—for WaveRNN, for iMEC with 20 bits, the confidence interval was estimated from 100 runs, not 1000. We also made a plotting error in Figure 4—the error bars were wider than they should have been. A corrected version is shown below in Figure 6.

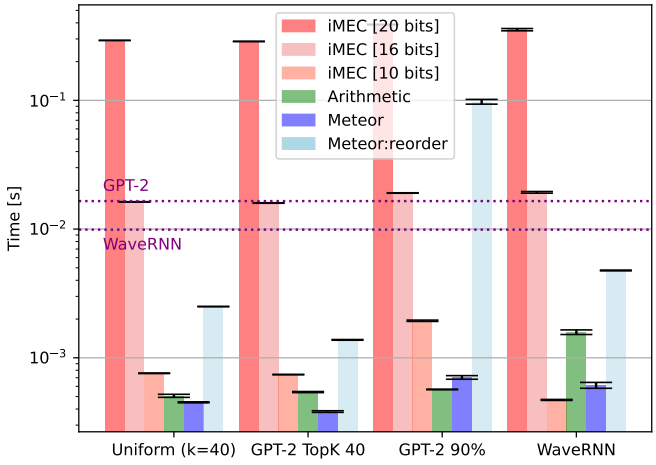

Figure 6: Comparative speed evaluation from the cover distribution of our method and baselines. Each method is evaluated at optimal hyperparameters. Errorbars indicate estimates of 95% confidence intervals computed using the central limit theorem from 1000 runs. Horizontal lines indicate the amount of time required for model inference for GPT-2 and WaveRNN.

A.1 iMEC Error Rate

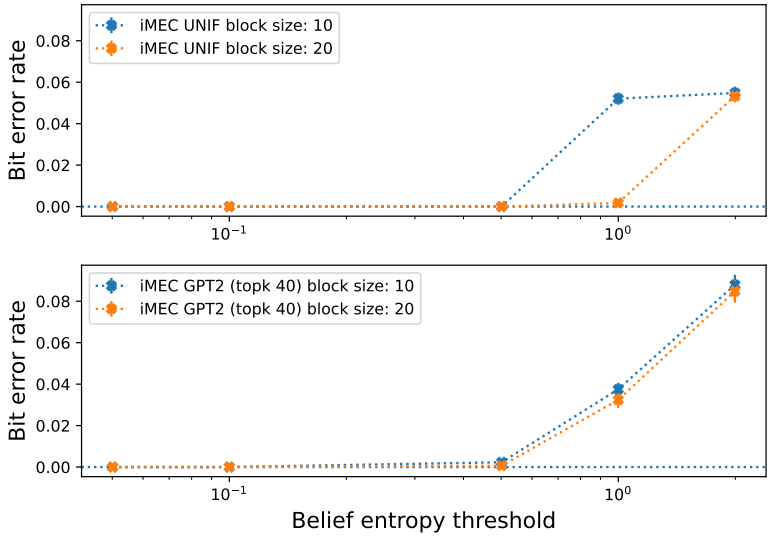

Figure 7: Bit error rate as a function of threshold size. The error bars shown are the standard deviation of the mean over 100 trajectories.

We show error rate as a function of the belief entropy threshold in Figure 7. As is suggested by the figure, the error rate can be made arbitrarily small by selecting a sufficiently small treshold value.

A.2 Non-Termination Frequency for Arithmetic Coding

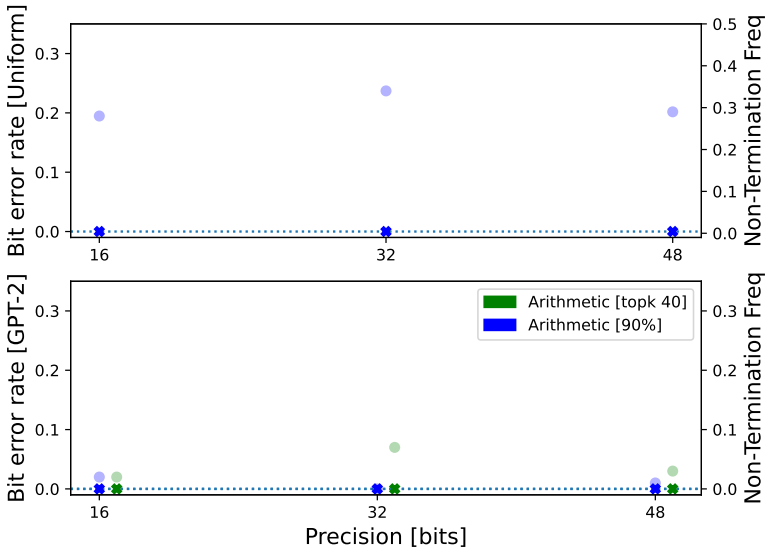

Figure 8: Bit error rate (X) and non-termination frequencies (circles) for arithmetic coding vs. precision hyperparameter. We show estimates over 100 trajectories.

A plot of the error rate and non-termination frequency is shown in Figure 8. While we did not observe errors, non-termination occurred with non-negligible probability.

A.3 Stegotext samples

To illustrate the effect of bias, we reproduce a sample stegotext for both iMEC (block size 10), as well as Meteor:reorder (precision 32) and the private message length is 20 bytes. Both examples have been mildly post-processed for readability, including by removing special characters and whitespaces.

Context:

> Heck horses are dun or grullo (a dun variant) in color, with no white markings. The breed has primitive markings, including a dorsal stripe and horizontal striping on the legs. Heck horses generally stand between 12.2 and 13.2 hands (50 and 54 inches, 127 and 137 cm) tall. The head is large, the withers low, and the legs and hindquarters

iMEC produces the following stegotext:

> are short. The neck is wide and thick, a characteristic that can be inherited from the male. The face can be seen as a broad head, with pointed toes. The head and neck are often used as a tool for hunting, though their appearance often depends on their social organization. The legs are

Meteor:reorder produces the following stegotext:

> have a narrow and angular shape. The fore and hind legs are longer than the head. The tail is broad and short in a shape similar to the neck or neckbone. The front legs have a sharp protrusion that leads from the head to the head but not from the tail. The hind legs have long pangs (2) and lower

Note how Meteor:reorder's high bias seemingly lowers the content quality of the output text.

A.4 Broader Impact

Our work makes a fundamental algorithmic contribution to shared private key steganography. We do not feel that our contribution raises specific negative societal concerns beyond those regarding shared private key steganography as a field.