PACER: PROVABLY SECURE STEGANOGRAPHY UNDER ASYMMETRIC RECEIVER ACCESS WITH CERTIFIED RECOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Many high-capacity provably secure steganography (PSS) schemes for large language models require the receiver to reproduce the sender's token distribution using the same model and prompt. In practice, privacy or resource constraints may prevent the receiver from accessing the sender's prompt or model, creating asymmetric receiver access. Existing asymmetric baselines use token-only recovery to avoid this dependence, but often have low embedding rates or limited recovery reliability and do not benefit from receiver-side resources that may remain available. We present PACER, a provably secure linguistic steganography framework for asymmetric receiver access with certified recovery. PACER separates the sender's token distribution used during generation from the receiver-side information used for recovery. When a receiver-side distribution is available, PACER combines maximal coupling with token-class coupling to extract recovery evidence while preserving the sender's token distribution. Otherwise, token-class coupling alone enables token-only recovery. An e-process accumulates this evidence and certifies payload blocks under a prescribed error bound. We prove PACER's computational security and show that the probability of incorrect recovery in any certification process is at most this bound. Experiments show that PACER achieves reliable recovery across different receiver access modes and that receiver-side distributions increase the embedding rate over token-only recovery. With receiver-side information, PACER achieves more than 42 times the embedding rate of the reliable asymmetric baseline. In token-only mode, PACER achieves more than 16 times that rate, with faster encoding and decoding.

1 INTRODUCTION

Generative models enable secret messages to guide text generation, with the resulting text directly serving as the stego (Yang et al., 2018; Chen et al., 2019). Recent provably secure steganography (PSS) schemes preserve the model's conditional distribution at each generation step, so that stegotext follows the same distribution as ordinary model sampling (Kaptchuk et al., 2021; Ding et al., 2023; Liao et al., 2025). Many high-capacity PSS schemes, however, require the receiver to reproduce the sender's token distribution using the same model and prompt. A mismatch in either can cause recovery failure (Bai et al., 2025a; Wang et al., 2025; Yan & Murawaki, 2026; Wang et al., 2026).

In practice, generated text may leave its original generation environment, while privacy or resource constraints may prevent the receiver from accessing the sender's prompt or running its model (Jiang et al., 2025b; Yan & Ding, 2025). The receiver may instead rely on a smaller model or partial context, which cannot reproduce the sender's token distribution exactly but may still provide useful information for recovery. We call this setting asymmetric receiver access.

Existing approaches leave a practical gap between two extremes. ADS (Pang & Bai, 2026a) and weakly asymmetric steganography (Liao et al., 2026) relax receiver-side requirements but still rely on shared-model access or exact partial information about the sender's token distribution. At the other extreme, USLM (Zamir, 2024), PRSSA (Bai et al., 2025b), and Disreo (Jiang et al., 2026) use only observed tokens for recovery. Although this eliminates dependence on the sender's model and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

prompt, these methods do not use available receiver-side resources and have low embedding rates or limited recovery reliability.

We present PACER, a framework for **P**rovably secure steganography under **A**symmetric receiver access with **Ce**rtified **R**ecovery. PACER follows the observation that the receiver need not reproduce the sender's token distribution exactly as long as its available information provides useful evidence for payload recovery. PACER embeds and recovers the payload block by block. When receiver-side information is available, PACER combines maximal coupling with token-class coupling to extract recovery evidence while preserving the sender's token distribution. Maximal coupling maximizes the probability that a proposal reconstructed by the receiver matches the generated token (Vaserstein, 1969; Wang et al., 2021), while token-class coupling extracts additional evidence from the remaining probability mass after proposal rejection. As receiver-side information becomes more limited, token-class evidence becomes more important. When no receiver-side distribution is available, token-class coupling alone enables token-only recovery.

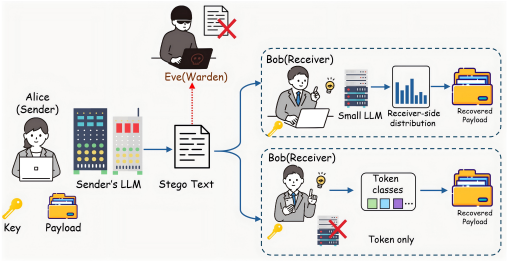


Figure 1: PACER preserves the sender's generation distribution while adapting recovery to the model and context available to the receiver.

During recovery, PACER considers all possible payload values for each block and derives evidence for each candidate from the observed tokens. An e-process (Ville, 1939; Shafer et al., 2011; Ramdas et al., 2023) accumulates this evidence over successive tokens until exactly one candidate crosses the certification threshold, at which point PACER certifies and recovers the block. We prove that the probability of incorrect recovery during any certification process does not exceed the prescribed error bound. The sender reproduces the receiver's certification decisions, allowing both endpoints to advance through payload blocks without an additional synchronization channel.

We evaluate PACER across four datasets and three model families under different receiver access conditions. With receiver-side information, PACER achieves an embedding rate of up to 630.8 bits/10^3 tokens and exceeds the reliable asymmetric baseline PRSSA (Bai et al., 2025b) by more than $42\times$. Even in token-only mode, PACER achieves more than $16\times$ its embedding rate, with faster encoding and decoding. Ablations show that token-class coupling provides larger gains as the sender's and receiver-side distributions overlap less. Neural steganalysis accuracy remains between 49.12% and 51.74%, close to random guessing. We further evaluate receiver-side retokenization and prove that the certification error bound still holds.

This paper makes four contributions. **Problem formulation.** We formulate provably secure steganography under asymmetric receiver access, where the receiver may lack the sender's prompt or model and cannot reconstruct the sender's token distribution. **Embedding construction.** We develop a distribution-preserving construction that combines maximal coupling with token-class coupling to use available receiver-side information while preserving the sender's token distribution. PACER also supports token-only recovery when this information is unavailable, and we prove its computational security. **Certified recovery.** We introduce a certified decoder that accumulates evidence for payload candidates using an e-process and prove a prescribed bound on incorrect block recovery, including under receiver-side retokenization. **Evaluation.** We evaluate PACER across four datasets, three model families, and different receiver access conditions, demonstrating reliable recovery and substantial improvements in embedding rate and encoding and decoding efficiency over existing asymmetric baselines.

2 Problem Setting

PACER embeds a binary payload into language-model-generated text for recovery by a receiver that may lack the sender's model, prompt, or both. We next formalize the system and its receiver-access conditions.

2.1 System Model

A sender uses model $\mathcal{M}$ and prompt x to embed a binary payload bitstream B while generating text. If the payload length is not divisible by b, B is padded according to a fixed rule. PACER divides B into complete b-bit blocks from $\mathcal{C} = \{0,1\}^b$ and embeds them sequentially. The resulting token sequence is $Y_{1:T}$, with generated prefix $Y_{<t}$ and next-token distribution $P_t = P_{\mathcal{M}}(\cdot \mid x, Y_{<t})$. The endpoints share only a secret key K and a tokenizer with vocabulary $\mathcal{V}$, while all other parameters are public.

We consider only a passive probabilistic polynomial-time (PPT) warden $\mathcal{A}$ that observes the transmitted text and attempts to detect hidden communication, without insertion, deletion, substitution, or rewriting. It knows PACER, $\mathcal{M}$, x, the receiver mode and resources, and all public parameters, but not K. Appendix B gives the detailed threat model.

Following most existing PSS schemes, our core formulation operates on the token IDs produced during generation. In actual communication, however, the receiver observes the transmitted text and must retokenize it before recovery. Even when both endpoints use the same tokenizer, retokenization may not reproduce the sender's generated token sequence. This phenomenon is known as tokenization ambiguity (Nozaki & Murawaki, 2022; Qi et al., 2024; Wang et al., 2025). Corollary 3.1 gives the certification error bound under tokenization ambiguity, and Table 4 evaluates PACER's recovery performance in this setting.

2.2 Receiver Access Modes

We focus on asymmetric receiver access, where the receiver cannot reproduce the sender's token distribution P_t because it lacks the sender's prompt or model. We represent these conditions using four receiver access modes $r \in \{\mathsf{PD}, \mathsf{MD}, \mathsf{DD}, \mathsf{TO}\}$, where PD (prompt-degraded) uses the sender's model without the prompt, MD (model-degraded) uses a smaller model with the prompt, DD (dual-degraded) uses a smaller model without the prompt, and TO (token-only) uses only observed tokens.

Let $\widetilde{\mathcal{M}}$ denote the smaller model available to the receiver in MD and DD. We instantiate it with a smaller model from the same family as $\mathcal{M}$ to retain tokenizer compatibility. In the first three modes, both endpoints construct the same receiver-side distribution Q_t from the available resources and generated prefix $Y_{<t}$. In TO, the receiver uses only the observed tokens and constructs no Q_t, but none of the four modes requires the receiver to reproduce P_t.

2.3 Objectives

Steganographic security. Steganographic security requires the generated text to be computationally indistinguishable from ordinary model sampling under the same model and prompt.

Certified recovery. Certified recovery requires a prescribed bound on the probability of recovery error.

3 PACER Method

PACER separates the sender's token distribution used for generation from the information used for recovery. Accordingly, distribution-preserving embedding preserves P_t, while certified recovery bounds the probability of incorrect block output. PACER embeds and recovers the payload block by block. Only the choices associated with the true block value affect token generation, while the receiver evaluates all payload candidates from the observed tokens. The receiver converts these observations into candidate paths and accumulates their evidence until the current block is certified.

3.1 Distribution-Preserving Embedding

For the current b-bit block, let $c^* \in \mathcal{C}$ denote its true value. PACER generates one token at a time to embed c^*, while every output token follows the sender's token distribution P_t. Because only the choices associated with c^*, rather than those for incorrect candidates, affect the generation of Y_t, the receiver can accumulate path evidence that distinguishes the true value over successive tokens.

PACER uses a domain-separated pseudorandom function (PRF) to derive the pseudorandom values for each candidate. We use inverse cumulative distribution function (inverse-CDF) sampling throughout the construction. Given a discrete distribution D and a uniform value $u \in [0,1)$, inverse-CDF

sampling returns the token whose cumulative probability interval contains u, thereby producing a sample from D.

Maximal coupling When Q_t is available, PACER uses maximal coupling (Vaserstein, 1969; Wang et al., 2021) and inverse-CDF sampling on Q_t for each payload candidate c to obtain a proposal $z_t(c)$. The receiver repeats this computation for every candidate, whereas the sender uses only $z = z_t(c^*)$ to generate the current token.

During recovery, the receiver compares $z_t(c)$ with the observed token Y_t. For the true candidate, maximal coupling maximizes the probability of $Y_t = z$ under the required marginals $z \sim Q_t$ and $Y_t \sim P_t$. An incorrect candidate's proposal does not participate in the generation of Y_t.

Specifically, the sender accepts z with probability $\min\{1, P_t(z)/Q_t(z)\}$.

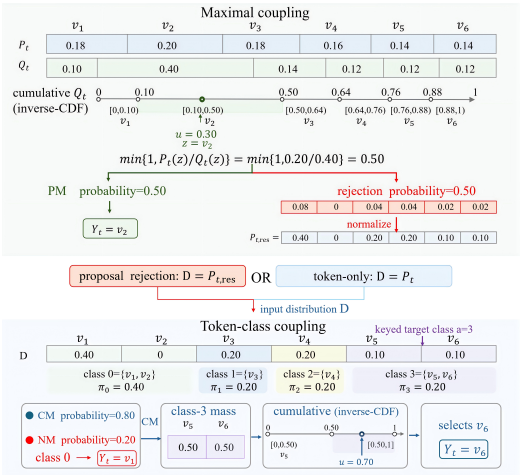


Figure 2: Distribution-preserving embedding.

If accepted, it outputs $Y_t = z$. For any token v, the accepted branch contributes

$$Q_t(v) \min\left\{1, \frac{P_t(v)}{Q_t(v)}\right\} = \min\{P_t(v), Q_t(v)\}$$

of probability mass. Its total probability is $\omega(P_t, Q_t) = \sum_{v \in \mathcal{V}} \min\{P_t(v), Q_t(v)\}$.

If the proposal is rejected, the remaining mass of P_t defines

$$P_{t,\mathrm{res}}(v) = \frac{P_t(v) - \min\{P_t(v), Q_t(v)\}}{1 - \omega(P_t, Q_t)}$$

when $\omega(P_t, Q_t) < 1$. Sampling the residual branch with marginal $P_{t,\mathrm{res}}$ gives, for every token v,

$$\Pr[Y_t = v] = \min\{P_t(v), Q_t(v)\} + P_t(v) - \min\{P_t(v), Q_t(v)\} = P_t(v).$$

Moreover, $P_{t,\mathrm{res}}$ assigns zero mass to the rejected proposal, so $\Pr[Y_t = z] = \omega(P_t, Q_t)$, the maximal equality probability for marginals P_t and Q_t.

Token-class coupling After proposal rejection, the final token cannot equal the rejected proposal, but it can still be used to distinguish payload candidates. PACER therefore assigns each candidate a token partition and a target class. During recovery, the receiver checks whether the observed token belongs to that candidate's target class.

This mechanism does not require Q_t. When Q_t is available, token-class coupling acts on the residual distribution after proposal rejection. When Q_t is unavailable, it acts directly on P_t, thereby supporting token-only recovery. We write

$$D = \begin{cases} P_{t,\mathrm{res}}, & \text{after proposal rejection,} \\ P_t, & \text{in token-only mode.} \end{cases}$$

For the true block value c^*, let $G : \mathcal{V} \to \{0, \ldots, M-1\}$ denote its token partition, and let a denote its target class, chosen uniformly from $\{0, \ldots, M-1\}$. Define $\pi_j = \sum_{v:G(v)=j} D(v)$, for $j = 0, \ldots, M-1$, as the probability mass of class j under D.

During recovery, the receiver checks whether $G(Y_t) = a$. For the true candidate, PACER maximizes the probability of this event under the uniform target class and the required marginal $Y_t \sim D$. For an incorrect candidate, its target class does not participate in token generation.

PACER accepts the target class a with probability $\min\{1, M\pi_a\}$. If a is accepted, PACER draws Y_t from D conditioned on class a. If a is rejected, it selects a class in proportion to the residual masses $\pi_j - \min\{\pi_j, 1/M\}$ and then draws Y_t from D conditioned on the selected class.

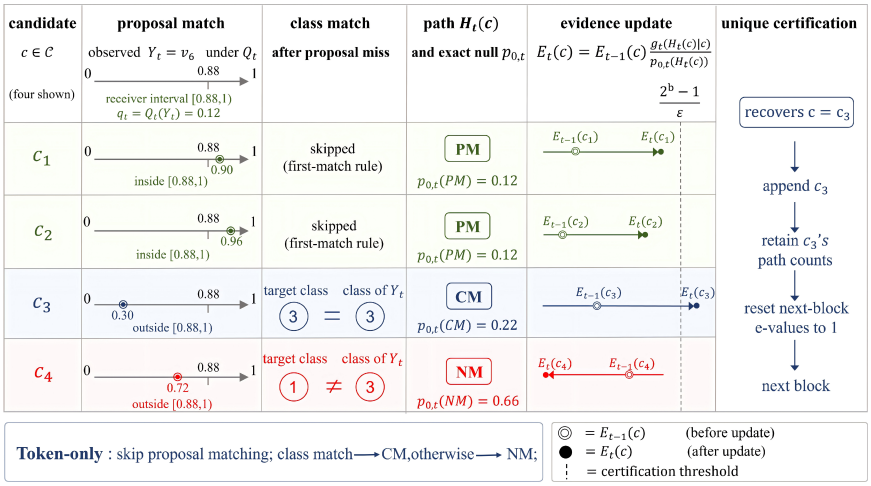


Figure 3: Certified recovery. Each observed token determines one path and one e-value update for every payload candidate. A unique threshold crossing certifies the corresponding block.

For every class j, the accepted and residual branches together contribute $\min\{1/M, \pi_j\} + [\pi_j - \min\{1/M, \pi_j\}] = \pi_j$. The output token therefore retains marginal distribution D. This construction is a maximal coupling between the uniform target class and the class induced by a token drawn from D, and thus maximizes the probability that the generated token belongs to the target class.

Combining the two couplings preserves P_t when Q_t is available, while token-class coupling alone preserves it in token-only mode. Hence, $Y_t \sim P_t$ in every receiver mode. Figure 2 summarizes the embedding process. Complete pseudocode is provided as Algorithm 1 in Appendix H, and complete proofs are provided in Appendix C.1.

Lemma 3.1 (One-step distribution preservation)**.** *In the random-function model with exact sampling and arithmetic, PACER produces $Y_t \sim P_t$ for every realized prefix, payload block, and receiver mode.*

Proof sketch. For token-class coupling, let π_j be the mass of class j under its working distribution D. The accepted branch selects j with mass $\min\{1/M, \pi_j\}$, and the residual branch supplies $\pi_j - \min\{1/M, \pi_j\}$. Their sum is π_j, so conditional sampling within the selected class preserves D. Thus TO preserves P_t directly. When Q_t is available, proposal acceptance contributes $\min\{P_t(v), Q_t(v)\}$ to token v, while the residual branch contributes $P_t(v) - \min\{P_t(v), Q_t(v)\}$. The two branches therefore sum to $P_t(v)$. □

Theorem 3.1 (Steganographic Security)**.** *With exact sampling and arithmetic and a secure domain-separated PRF, PACER's generated text $Y_{1:T}$ is computationally indistinguishable from ordinary model sampling $X_{1:T}$ for every PPT warden and polynomially bounded T:*

$$|\Pr[\mathcal{A}(Y_{1:T}) = 1] - \Pr[\mathcal{A}(X_{1:T}) = 1]| \leq \mathrm{negl}(\lambda).$$

Here $X_{1:T}$ uses the same model, prompt, and stopping rule, $\mathcal{A}$ is any PPT warden, and λ is the security parameter.

Proof sketch. Hybrid H_0 is the real construction, and H_1 replaces the domain-separated PRF by a random function. Any distinguisher between these hybrids breaks PRF security, while Lemma 3.1 shows that under H_1, $\Pr[Y_t = v \mid Y_{<t} = y_{<t}] = P_{\mathcal{M}}(v \mid x, y_{<t})$ for every prefix and payload. The chain rule makes $Y_{1:T}$ identically distributed to ordinary model sampling, so the remaining hybrid gap is zero. □

3.2 Certified Recovery

For every observed token Y_t, the receiver evaluates each payload candidate using proposal equality when Q_t is available and target-class membership in all modes. PACER maps these outcomes to candidate paths, compares them with their wrong-candidate baseline, and accumulates evidence across tokens until the current block is certified. Figure 3 illustrates this process.

Candidate paths For each candidate c, PACER checks the two possible events in the same order in which they arise during embedding and assigns the first one that holds. We call this the *first-match rule*. When Q_t is available, the receiver first compares the candidate's proposal $z_t(c)$ with Y_t. If they are equal, it assigns $H_t(c) = \mathsf{PM}$, where PM denotes proposal match, and performs no further check for that candidate. Only if the proposal does not match does the receiver check whether Y_t belongs to the candidate's target class. If it does, it assigns $H_t(c) = \mathsf{CM}$, where CM denotes class match. Otherwise, PACER assigns $H_t(c) = \mathsf{NM}$, where NM denotes no match.

Thus, when Q_t is available, $\mathcal{H}^Q = \{\mathsf{PM}, \mathsf{CM}, \mathsf{NM}\}$. Each candidate receives exactly one path at every step. For the true candidate, this path matches the embedding branch because $P_{t,\mathrm{res}}$ excludes a rejected proposal, and rejecting target class a leaves zero residual mass in that class. In token-only mode there is no proposal, so $\mathcal{H}^{\mathrm{TO}} = \{\mathsf{CM}, \mathsf{NM}\}$.

PACER next needs a reference for how often each path would occur for an incorrect candidate. For candidate c, consider the null hypothesis $c \neq c^*$, and let $p_{0,t}$ denote its conditional path distribution. When Q_t is available, define $q_t = Q_t(Y_t)$. In the ideal random-function model, an incorrect candidate's choices do not affect the generation of Y_t. Its proposal equals Y_t with probability q_t. By the first-match rule, the class check is reached only when the proposal does not match, and a uniform target class contains Y_t with probability $1/M$. Hence,

$$p_{0,t}(\mathsf{PM}) = q_t, \qquad p_{0,t}(\mathsf{CM}) = (1 - q_t)\frac{1}{M}, \qquad p_{0,t}(\mathsf{NM}) = (1 - q_t)\left(1 - \frac{1}{M}\right).$$

In token-only mode, $p_{0,t}(\mathsf{CM}) = 1/M$ and $p_{0,t}(\mathsf{NM}) = 1 - 1/M$.

Evidence accumulation and certification Because a single token rarely identifies an entire payload block, PACER accumulates each candidate's path evidence over successive tokens. Let $E_t(c)$ denote the e-value of candidate c after processing Y_t. Before observing the current path, PACER assigns a predictive distribution $g_t(\cdot \mid c)$ over the path space and updates

$$E_t(c) = E_{t-1}(c)\frac{g_t(H_t(c) \mid c)}{p_{0,t}(H_t(c))}, \qquad E_0(c) = 1.$$

The denominator is the wrong-candidate probability of the observed path, while the numerator is candidate c's predictive probability. Consistent excess over the baseline makes $E_t(c)$ grow.

Ideally, g_t would match the true candidate's conditional path distribution, but because that distribution depends on the unavailable P_t, PACER instead uses the Krichevsky–Trofimov (KT) predictor (Krichevsky & Trofimov, 1981; Watanabe & Roos, 2015) to adapt the prediction to the candidate's preceding path counts. Let $n_{t-1,h}(c)$ denote the number of occurrences of path h available for candidate c before step t. Then

$$g_t(h \mid c) = \frac{n_{t-1,h}(c) + 1/2}{\sum_{h' \in \mathcal{H}} n_{t-1,h'}(c) + |\mathcal{H}|/2}, \qquad h \in \mathcal{H}.$$

The validity argument only requires g_t to be fixed before the current path is observed and normalized over the path space. The predictor choice therefore affects how quickly evidence accumulates, but not the validity of the e-process or the certification error bound.

Let $\varepsilon \in (0, 1)$ be the prescribed certification error bound. PACER certifies candidate c when $E_t(c) \geq (2^b - 1)/\varepsilon$. The factor $2^b - 1$ accounts for the incorrect candidates in the current block. If no candidate reaches the threshold, PACER continues with the next token. If exactly one candidate reaches it, PACER recovers the corresponding b-bit block. After every generated token, the sender performs the same candidate-path and e-value updates as the receiver. A unique certification therefore advances both endpoints to the next block at the same token.

The first block starts with zero path counts. After a block is uniquely certified, the certified candidate's cumulative path counts become the starting counts for every candidate in the next block of the same text, and all e-values are reset to one.

If multiple candidates reach the threshold on the same token, PACER does not recover the current block and terminates recovery from the current text. If the text ends before a unique certification, the current block remains unrecovered and can be continued in a later generation.

A certification process begins when the current block's e-values are initialized to one and ends at a unique threshold crossing, a simultaneous multiple crossing, or the end of the current text. Complete pseudocode for certified recovery is provided as Algorithm 2 in Appendix H, and complete proofs are provided in Appendix C.2.

Lemma 3.2 (Incorrect-candidate evidence)**.** *For each $c \neq c^*$, $\{E_t(c)\}$ is a nonnegative supermartingale during a certification process, initialized at one.*

Proof sketch. Domain separation makes an incorrect candidate's proposal, partition, and target class independent of the generated token. Its path therefore follows the baseline $p_{0,t}$. Because g_t is fixed before observing the current path and sums to one,

$$\mathbb{E}\left[\frac{g_t(H_t(c) \mid c)}{p_{0,t}(H_t(c))} \,\middle|\, \mathcal{F}_{t-1}, Y_t\right] = \sum_{h:p_{0,t}(h)>0} g_t(h \mid c) \leq 1.$$

Multiplying by $E_{t-1}(c)$ and applying the tower property proves the claim. □

Theorem 3.2 (Certified Recovery)**.** *In the random-function model with exact sampling and arithmetic, for a b-bit block with true value c^*, PACER satisfies $\Pr[$the process outputs some $c \neq c^*] \leq \varepsilon$ during each certification process.*

Proof sketch. Ville's inequality and Lemma 3.2 bound the probability that one incorrect candidate reaches $(2^b - 1)/\varepsilon$ by $\varepsilon/(2^b - 1)$. A union bound over the $2^b - 1$ incorrect candidates gives ε. The bound is conditional on every process-start history and therefore also holds unconditionally. □

Corollary 3.1 (Receiver-side retokenization)**.** *The same certification bound holds after any deterministic receiver-side retokenization.*

Proof sketch. Candidate-specific random-function values remain independent of token generation after deterministic retokenization. Conditioning on the retokenized history therefore preserves the wrong-candidate baselines and the supermartingale argument. □

4 Experiments

Setup We generate 300-token continuations for 1,000 prompts per condition from C4, IMDB, Essays, and CNN–DailyMail (Raffel et al., 2020; Maas et al., 2011; IvyPanda, 2024; See et al., 2017). Sender/receiver model pairs are OPT-6.7B/1.3B, Llama-3.1-8B/3.2-1B, and Qwen3-8B/1.7B (Zhang et al., 2022; AI@Meta, 2024; Yang et al., 2025). Unless stated otherwise, PACER uses $b = 16$, $M = 16$, and $\varepsilon = 10^{-6}$, with temperature 1 and top-$k = 50$. We compare with Disreo, PRSSA, and USLM under asymmetric access. Symmetric Discop and Meteor serve as references, and their no-prompt variants test asymmetric mismatch. ADS (Pang & Bai, 2026a) and weakly asymmetric steganography (Liao et al., 2026) assume different receiver access and are not directly comparable. Full configurations, including hardware, are in Appendices D and F. Random Sampling denotes generation from $\mathcal{M}$ without payload embedding.

Metrics and protocol For a continuation of T tokens with L recovered bits, the embedding rate is $R_{\text{emb}} = 10^3 L/T$. BER is the fraction of the L recovered bits that differ from the payload, excluding unrecovered suffixes from its denominator. Encoding and decoding speeds divide L by their respective running times, while generation speed uses tokens per second. Perplexity is computed with a fixed external Mistral-7B evaluator (Jiang et al., 2023). Detectability is measured with FCN (Yang et al., 2019b), CNN (Yang et al., 2020b), R-BiLSTM-C (RBC) (Niu et al., 2019), and BiLSTM-Dense (BLD) (Yang et al., 2020a). An accuracy of 50% is random guessing.

Each reported condition averages sample-level metrics over 1,000 continuations, using generated token IDs in the main evaluation and receiver-retokenized text in a separate experiment.

Main results Table 1 shows that PACER-PD reaches 630.8 bits/10^3 tokens and exceeds the reliable asymmetric baseline by more than $42\times$. PACER-MD and PACER-DD remain above PACER-TO, confirming that even mismatched or incomplete receiver-side resources are useful, while PACER-TO reaches 213.7 bits/10^3 tokens and exceeds the baseline by more than $16\times$ without a receiver-side model or prompt. These gains persist across datasets and model families (Figure 6 in the appendix). PACER-PD improves encoding speed by $4.0\times$–$8.4\times$ despite receiver-side inference, giving it the

Table 1: Main results on C4 across three model families. Bold numbers mark the best valid values among asymmetric methods within each model family, while red BER values indicate clear recovery failure.

Family	Receiver access	Method	Receiver resources: Model	Receiver resources: Prompt	$R_{\text{emb}}\uparrow$ (bit/10^3 tok.)	$S_{\text{gen}}\uparrow$ (tok./s)	$S_{\text{enc}}\uparrow$ (bit/s)	$S_{\text{dec}}\uparrow$ (bit/s)	PPL $\downarrow$	BER $\downarrow$ (%)
OPT	Asymmetric	PACER-PD	$\mathcal{M}$	No	**630.8**	25.41	**16.03**	26.57	8.60	0.00
		PACER-MD	$\widetilde{\mathcal{M}}$	Yes	524.1	30.41	15.94	32.34	8.58	0.00
		PACER-DD	$\widetilde{\mathcal{M}}$	No	444.9	29.90	13.30	27.68	8.51	0.00
		PACER-TO	None	No	211.9	31.16	6.60	**86.71**	8.64	0.00
		Disreo	None	No	57.2	33.22	1.90	49.45	8.56	2.15
		PRSSA	None	No	13.0	32.31	0.42	40.45	8.54	0.00
		USLM	None	No	39.9	**37.54**	1.50	23.79	8.57	8.41
		Discop w/o prompt	$\mathcal{M}$	No	2355.3	37.48	88.27	1.85	8.76	44.44
		Meteor w/o prompt	$\mathcal{M}$	No	1851.3	31.21	57.78	62.72	8.32	50.21
	Symmetric reference	Discop	$\mathcal{M}$	Yes	2355.3	37.48	88.27	88.34	8.76	0.00
		Meteor	$\mathcal{M}$	Yes	1851.3	31.21	57.78	57.31	8.32	0.00
	–	Random Sampling	–	–	–	38.02	–	–	8.57	–
Llama3	Asymmetric	PACER-PD	$\mathcal{M}$	No	**578.5**	19.51	**11.29**	17.30	7.11	0.00
		PACER-MD	$\widetilde{\mathcal{M}}$	Yes	457.5	23.99	10.98	21.61	7.18	0.00
		PACER-DD	$\widetilde{\mathcal{M}}$	No	357.9	23.57	8.44	16.91	7.03	0.00
		PACER-TO	None	No	213.7	27.44	5.86	**89.02**	7.12	0.00
		Disreo	None	No	61.9	25.85	1.60	21.11	7.57	3.23
		PRSSA	None	No	13.2	25.76	0.34	38.84	7.22	0.00
		USLM	None	No	37.1	**30.01**	1.11	19.79	7.44	9.30
		Discop w/o prompt	$\mathcal{M}$	No	2347.7	29.18	68.51	7.15	7.23	52.00
		Meteor w/o prompt	$\mathcal{M}$	No	1779.5	24.65	43.86	48.84	7.28	49.55
	Symmetric reference	Discop	$\mathcal{M}$	Yes	2347.7	29.18	68.51	68.53	7.23	0.00
		Meteor	$\mathcal{M}$	Yes	1779.5	24.65	43.86	42.65	7.28	0.00
	–	Random Sampling	–	–	–	31.11	–	–	7.09	–
Qwen3	Asymmetric	PACER-PD	$\mathcal{M}$	No	**279.6**	14.57	**4.07**	6.49	5.97	0.00
		PACER-MD	$\widetilde{\mathcal{M}}$	Yes	217.8	16.15	3.52	6.13	5.91	0.00
		PACER-DD	$\widetilde{\mathcal{M}}$	No	149.2	15.89	2.37	4.28	5.79	0.00
		PACER-TO	None	No	122.1	22.31	2.72	**52.68**	5.88	0.00
		Disreo	None	No	41.1	24.82	1.02	11.74	6.11	5.21
		PRSSA	None	No	6.6	24.24	0.16	24.00	5.95	0.00
		USLM	None	No	23.5	**26.81**	0.63	12.23	5.94	12.77
		Discop w/o prompt	$\mathcal{M}$	No	1646.8	28.01	46.13	23.12	5.83	51.13
		Meteor w/o prompt	$\mathcal{M}$	No	1288.3	24.11	31.06	28.69	5.73	50.64
	Symmetric reference	Discop	$\mathcal{M}$	Yes	1646.8	28.01	46.13	46.15	5.83	0.00
		Meteor	$\mathcal{M}$	Yes	1288.3	24.11	31.06	30.40	5.73	0.00
	–	Random Sampling	–	–	–	27.02	–	–	5.99	–

highest valid asymmetric encoding speed, while PACER-TO requires no such inference and improves decoding speed by $1.8\times$–$2.3\times$ to achieve the highest decoding speed.

PACER has no observed bit errors in any receiver mode. PRSSA also has zero observed BER but at a much lower rate, whereas Disreo and USLM have both lower rates and nonzero BER. Discop and Meteor recover reliably with shared prompts, but without the original prompt their BER approaches 50%.

Table 2: Steganalysis accuracy (%) across four PACER modes.

Mode	FCN	CNN	RBC	BLD
PACER-PD	51.31	49.86	51.74	51.18
PACER-MD	49.12	50.47	50.11	49.68
PACER-DD	49.63	51.29	49.77	50.52
PACER-TO	50.58	50.83	51.36	51.45

Text quality and detectability PACER's PPL closely matches Random Sampling within each model family, while accuracy across the four steganalyzers in Table 2 remains between 49.12% and 51.74%, close to random guessing.

Table 3: Residual token-class coupling ablation with OPT models.

Receiver view	Model size	$\bar{\omega}$	R_{emb} (bit/10^3 tok.) w/o res.	R_{emb} (bit/10^3 tok.) w/ res.	Gain (%)
Prompt	125M	0.669	258.9	348.4	34.57
	350M	0.718	329.3	394.3	19.74
	1.3B	0.802	478.5	524.1	9.53
	2.7B	0.828	539.6	574.1	6.39
No prompt	125M	0.619	226.0	325.1	43.85
	350M	0.661	257.8	342.1	32.70
	1.3B	0.733	379.8	444.9	17.14
	2.7B	0.756	411.8	474.8	15.30

Residual token-class coupling Here, $\bar{\omega}$ averages the per-token overlap $\omega(P_t, Q_t)$ from Section 3.1, and smaller values indicate less overlap. Table 3 shows that residual token-class coupling raises embedding rate by 6.39%–43.85%, with larger gains for smaller receiver models and missing prompts. Maximal coupling uses shared mass for proposal evidence, whereas residual coupling uses the sender's remaining mass and matters more as overlap decreases.

Parameter sensitivity Relaxing ε lowers the certification threshold and increases embedding rate in all four modes (Figure 4). Nonzero BER appears only after ε is increased by several orders of magnitude. The default $\varepsilon = 10^{-6}$ is conservative while still maintaining a high embedding rate. Figure 5 shows that embedding rate increases with b because each certification recovers more bits, while the candidate space grows as 2^b and reduces processing speed at larger b. Beyond the default $b = 16$, the additional embedding-rate gain is limited while the processing-speed loss becomes substantial. Sensitivity to the number of token classes M is reported in Appendix G.

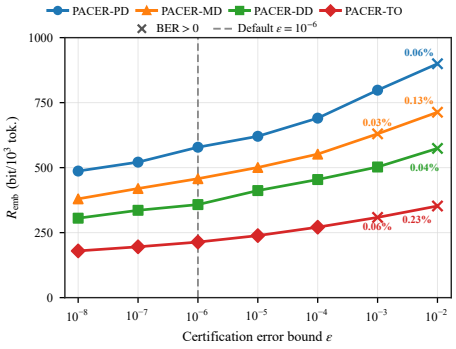


Figure 4: Embedding rate versus error bound ε.

Tokenization ambiguity Ambiguity rate denotes the fraction of samples with changed token IDs, rate drop the relative decrease in R_{emb}, and sender sync. the time required to reproduce recovery decisions.

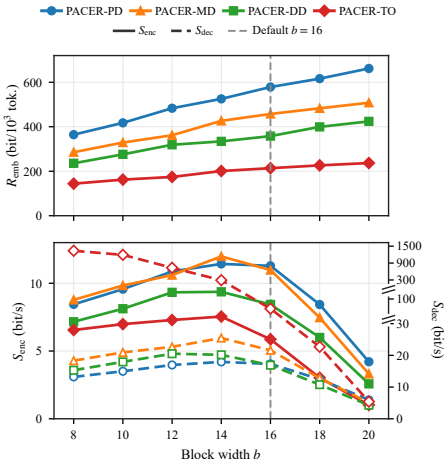


Figure 5: Rate and speed versus block width b.

Table 4 reports 6.5%–6.8% ambiguity and a 3.13%–4.15% rate drop, with no observed bit errors. Retokenization can alter candidate paths and reduce certified payload, but Corollary 3.1 preserves the certification bound. Sender synchronization takes 0.03–0.43 seconds per sample.

Multi-bit watermarking comparison MPAC (Yoo et al., 2024) and BiMark (Feng et al., 2025) target short payloads for detection and attribution and need not ensure complete recovery. Table 5 reports full recovery, the fraction of samples whose entire payload is recovered correctly. At 64 bits, both baselines reach 213.3 bits/10^3 tokens, but full recovery falls to 18.1% and 1.3%. PACER-TO reaches 213.7 with 100% recovery and zero observed BER, while PACER-PD reaches 578.5. BiMark has PACER-like PPL, whereas MPAC is substantially higher.

Table 4: PACER under tokenization ambiguity.

Mode	Ambiguity rate (%)	R_{emb} (bit/10^3 tok.) w/o retok.	R_{emb} (bit/10^3 tok.) w/ retok.	Rate drop (%)	Sender sync. (s)	BER (%)
PD	6.8	578.5	554.5	4.15	0.43	0.00
MD	6.5	457.5	443.0	3.17	0.20	0.00
DD	6.6	357.9	346.7	3.13	0.27	0.00
TO	6.8	213.7	206.3	3.46	0.03	0.00

Table 5: Multi-bit watermarking comparison.

Method	R_{emb} ↑ (bit/10^3 tok.)	BER ↓ (%)	Full recovery ↑ (%)	PPL ↓
MPAC (4-bit)	13.3	0.30	98.9	14.57
BiMark (4-bit)	13.3	0.50	98.2	7.18
MPAC (16-bit)	53.3	0.39	96.3	14.16
BiMark (16-bit)	53.3	0.91	93.0	7.26
MPAC (64-bit)	213.3	4.43	18.1	14.65
BiMark (64-bit)	213.3	7.98	1.3	7.11
PACER-PD	**578.5**	**0.00**	**100.0**	7.11
PACER-TO	213.7	**0.00**	**100.0**	7.12

5 Conclusion

PACER provides provably secure steganography under asymmetric receiver access while preserving the sender's distribution. Across four access modes, it bounds recovery errors after retokenization and improves embedding rate and efficiency on three model families and four datasets. Reliable recovery therefore does not require reconstructing the sender's distribution.

Reproducibility statement

We provide the code and accompanying instructions in the supplementary material.

Ethics statement

Steganography is dual-use. It can protect communication privacy but may also be used to conceal malicious activities. PACER improves payload recovery when receiver access is limited and may therefore make covert communication applicable to a broader range of settings. At the same time, accurately characterizing these capabilities and their applicable conditions can support future research on detection, defense, and security governance. We state PACER's security goals, assumptions, and scope, and release implementation and evaluation artifacts to support reproducibility and independent scrutiny. All experiments use public datasets and publicly released models in a controlled offline environment and involve no real users, human subjects, or personal data.

AI use statement

Generative AI tools were used for overall language polishing and manuscript compression, related-work research, and assistance in writing LaTeX and experimental code.

References

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Minhao Bai, Kaiyi Pang, Guorui Liao, Jinshuai Yang, and Yongfeng Huang. Shimmer: a provably secure steganography based on entropy collecting mechanism. In *Proceedings of the 34th USENIX Conference on Security Symposium*, SEC '25, USA, 2025a. USENIX Association. ISBN 978-1-939133-52-6.

Minhao Bai, Jinshuai Yang, Kaiyi Pang, Xin Xu, Zhen Yang, and Yongfeng Huang. Provably robust and secure steganography in asymmetric resource scenario. In *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 1438–1456, 2025b. doi: 10.1109/SP61157.2025.00155.

Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, 2019. doi: 10.1109/TIFS.2018.2871749.

Kejiang Chen, Hang Zhou, Hanqing Zhao, Dongdong Chen, Weiming Zhang, and Nenghai Yu. When provably secure steganography meets generative models, 2019. URL https://arxiv.org/abs/1811.03732v2.

Falcon Dai and Zheng Cai. Towards near-imperceptible steganographic text. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4303–4308, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1422. URL https://aclanthology.org/P19-1422/.

Jinyang Ding, Kejiang Chen, Yaofei Wang, Na Zhao, Weiming Zhang, and Neng H. Yu. Discop: Provably secure steganography in practice based on "distribution copies". *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 2238–2255, 2023. URL https://api.semanticscholar.org/CorpusID:260002610.

Tina Fang, Martin Jaggi, and Katerina Argyraki. Generating steganographic text with LSTMs. In Allyson Ettinger, Spandana Gella, Matthieu Labeau, Cecilia Ovesdotter Alm, Marine Carpuat, and Mark Dredze (eds.), *Proceedings of ACL 2017, Student Research Workshop*, pp. 100–106, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL https://aclanthology.org/P17-3017/.

Xiaoyan Feng, He Zhang, Yanjun Zhang, Leo Yu Zhang, and Shirui Pan. BiMark: Unbiased multilayer watermarking for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu

(eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 17049–17067. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/feng25u.html.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Nicholas J. Hopper, John Langford, and Luis von Ahn. Provably secure steganography. In Moti Yung (ed.), *Advances in Cryptology — CRYPTO 2002*, pp. 77–92, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45708-4.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 45408–45436, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/c5b00c5bdcc6fe35907dbcca03d27652-Paper-Conference.pdf.

IvyPanda. IvyPanda. https://ivypanda.com/essays/, 2024. Accessed: January 19, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Jun Jiang, Kejiang Chen, Na Zhao, Yuang Qi, Xin Zhang, Weiming Zhang, and Nenghai Yu. Disreo: Provably secure no-box-extraction linguistic steganography based on distribution reorganization. *IEEE Transactions on Multimedia*, 28:2970–2983, 2026. doi: 10.1109/TMM.2026.3651038.

Ya Jiang, Chuxiong Wu, Massieh Kordi Boroujeny, Brian Mark, and Kai Zeng. StealthInk: A multi-bit and stealthy watermark for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27685–27709. PMLR, 13–19 Jul 2025a. URL https://proceedings.mlr.press/v267/jiang25j.html.

Zhifeng Jiang, Zhihua Jin, and Guoliang He. PromptKeeper: Safeguarding system prompts for LLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 2712–2728, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.147. URL https://aclanthology.org/2025.findings-emnlp.147/.

Gabriel Kaptchuk, Tushar M. Jois, Matthew Green, and Aviel D. Rubin. Meteor: Cryptographically Secure Steganography for Realistic Distributions. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1529–1548, Virtual Event Republic of Korea, November 2021. ACM. ISBN 978-1-4503-8454-4. doi: 10.1145/3460120.3484550.

Andrew D Ker. Improved detection of lsb steganography in grayscale images. In *International workshop on information hiding*, pp. 97–115. Springer, 2004.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Raphail E. Krichevsky and Victor K. Trofimov. The performance of universal encoding. *IEEE Transactions on Information Theory*, 27:199–207, 1981. ISSN 0018-9448. doi: 10.1109/tit.1981.1056331.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=FpaCL1MO2C.

Guorui Liao, Jinshuai Yang, Weizhi Shao, and Yongfeng Huang. *A framework for designing provably secure steganography*. USENIX Association, USA, 2025. ISBN 978-1-939133-52-6.

Guorui Liao, Shuyu Gan, Jinshuai Yang, Yongfeng Huang, and Zhongyi Huang. From symmetry toward weak asymmetry: Secure steganography under the receiver's partial channel knowledge. In *35th USENIX Security Symposium (USENIX Security 26)*, Baltimore, MD, August 2026. USENIX Association. URL https://www.usenix.org/conference/usenixsecurity26/presentation/liao.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Yan Niu, Juan Wen, Ping Zhong, and Yiming Xue. A hybrid r-bilstm-c neural network based text steganalysis. *IEEE Signal Processing Letters*, 26(12):1907–1911, 2019. doi: 10.1109/LSP.2019.2953953.

Jumon Nozaki and Yugo Murawaki. Addressing segmentation ambiguity in neural linguistic steganography. In Yulan He, Heng Ji, Sujian Li, Yang Liu, and Chua-Hui Chang (eds.), *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 109–116, Online only, November 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.aacl-short.15. URL https://aclanthology.org/2022.aacl-short.15/.

Kaiyi Pang and Minhao Bai. Provable secure steganography based on adaptive dynamic sampling. In *35th USENIX Security Symposium (USENIX Security 26)*, Baltimore, MD, August 2026a. USENIX Association. URL https://www.usenix.org/conference/usenixsecurity26/presentation/pang-kaiyi.

Kaiyi Pang and Minhao Bai. Provably secure steganography based on list decoding, 2026b. URL https://arxiv.org/abs/2604.21394.

Yuang Qi, Kejiang Chen, Kai Zeng, Weiming Zhang, and Nenghai Yu. Provably secure disambiguating neural linguistic steganography. *IEEE Transactions on Dependable and Secure Computing*, pp. 1–14, 2024. doi: 10.1109/TDSC.2024.3519322.

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. Provably robust multi-bit watermarking for ai-generated text. In *Proceedings of the 34th USENIX Conference on Security Symposium*, SEC '25, USA, 2025. USENIX Association. ISBN 978-1-939133-52-6.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38(4):576–597, 2023. ISSN 0883-4237. doi: 10.1214/23-STS894. Publisher Copyright: © (2023) All Rights Reserved.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://www.aclweb.org/anthology/P17-1099.

Glenn Shafer, Alexander Shen, Nikolai Vereshchagin, and Vladimir Vovk. Test martingales, bayes factors and p-values. *Statistical Science*, 26(1):84–101, 2011. ISSN 0883-4237. doi: 10.1214/10-STS347.

L. N. Vaserstein. Markov processes over denumerable products of spaces, describing large systems of automata. *Probl. Peredachi Inf., 1969,*, pp. 64–72, 1969.

Jean Ville. *Étude critique de la notion de collectif*. PhD thesis, 1939. URL https://numdam.org/item/THESE_1939__218__1_0/.

Guanyang Wang, John O'Leary, and Pierre Jacob. Maximal couplings of the metropolis-hastings algorithm. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1225–1233. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/wang21d.html.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. Towards codable watermarking for injecting multi-bits information to llms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=JYu5Flqm9D.

Yaofei Wang, Gang Pei, Kejiang Chen, Jinyang Ding, Chao Pan, Weilong Pang, Donghui Hu, and Weiming Zhang. SparSamp: Efficient provably secure steganography based on sparse sampling. In *34th USENIX Security Symposium (USENIX Security '25)*. USENIX Association, August 2025.

Yaofei Wang, Weilong Pang, Kejiang Chen, Jinyang Ding, W Zhang, D Hu, and Nenghai Yu. Breaking the generative steganography trilemma: Anstega for optimal capacity, efficiency, and security. In *33rd Annual Network and Distributed System Security Symposium (NDSS'26), The Internet Society*, 2026.

Kazuho Watanabe and Teemu Roos. Achievability of asymptotic minimax regret by horizon-dependent and horizon-independent strategies. *J. Mach. Learn. Res.*, 16(1):2357–2375, January 2015. ISSN 1532-4435.

Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. A resilient and accessible distribution-preserving watermark for large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53443–53470. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wu24h.html.

Yangxinyu Xie, Xiang Li, Tanwi Mallick, Weijie Su, and Ruixun Zhang. Debiasing watermarks for large language models via maximal coupling. *Journal of the American Statistical Association*, 120 (551):1424–1436, 2025. doi: 10.1080/01621459.2025.2520455. URL https://doi.org/10.1080/01621459.2025.2520455.

Ruiyi Yan and Yugo Murawaki. Efficient provably secure linguistic steganography via range coding. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 890–907, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.39. URL https://aclanthology.org/2026.acl-long.39/.

Xiao Yan and Yi Ding. Are we there yet? a measurement study of efficiency for llm applications on mobile devices. In *Proceedings of the 2nd International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things*, FMSys, pp. 19–24, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400716089. doi: 10.1145/3722565.3727192. URL https://doi.org/10.1145/3722565.3727192.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger

Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Hao Yang, YongJian Bao, Zhongliang Yang, Sheng Liu, Yongfeng Huang, and Saimei Jiao. Linguistic steganalysis via densely connected LSTM with feature pyramid. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '20, pp. 5–10, New York, NY, USA, 2020a. Association for Computing Machinery. ISBN 9781450370509. doi: 10.1145/3369412.3395067. URL https://doi.org/10.1145/3369412.3395067.

Kuan Yang, Kejiang Chen, Weiming Zhang, and Nenghai Yu. Provably secure generative steganography based on autoregressive model. In *International Workshop on Digital Watermarking*, 2018. URL https://api.semanticscholar.org/CorpusID:59524394.

Zhong-Liang Yang, Xiao-Qing Guo, Zi-Ming Chen, Yong-Feng Huang, and Yu-Jin Zhang. Rnn-stega: Linguistic steganography based on recurrent neural networks. *IEEE Transactions on Information Forensics and Security*, 14(5):1280–1295, 2019a. doi: 10.1109/TIFS.2018.2871746.

Zhongliang Yang, Yongfeng Huang, and Yu-Jin Zhang. A fast and efficient text steganalysis method. *IEEE Signal Processing Letters*, 26(4):627–631, 2019b. doi: 10.1109/LSP.2019.2902095.

Zhongliang Yang, Yongfeng Huang, and Yu-Jin Zhang. Ts-csw: text steganalysis and hidden capacity estimation based on convolutional sliding windows. *Multimedia Tools Appl.*, 79(25–26): 18293–18316, July 2020b. ISSN 1380-7501. doi: 10.1007/s11042-020-08716-w. URL https://doi.org/10.1007/s11042-020-08716-w.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. Advancing beyond identification: Multi-bit watermark for large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4031–4055, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.224. URL https://aclanthology.org/2024.naacl-long.224/.

Or Zamir. Undetectable steganography for language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=fq6aQoMSHz.

Siyu Zhang, Zhongliang Yang, Jinshuai Yang, and Yongfeng Huang. Provably secure generative linguistic steganography. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3046–3055, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.268. URL https://aclanthology.org/2021.findings-acl.268.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Representation Learning*, volume 2024, pp. 43738–43772, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/beae9ed5316bcc48e616754c06c11875-Paper-Conference.pdf.

Zachary Ziegler, Yuntian Deng, and Alexander M 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)*, pp. 1210–1215, 2019.

A Background and Related Work

A.1 Provably Secure Steganography

Traditional steganography hides secret messages by modifying existing covers, but such modifications may leave detectable statistical artifacts (Ker, 2004; Boroumand et al., 2019). Generative language models provide a natural alternative carrier for linguistic steganography: secret messages can guide token selection during generation, while the generated text itself serves as the carrier (Fang et al., 2017; Yang et al., 2019a; Ziegler et al., 2019; Zhang et al., 2021).

Early neural methods explored vocabulary partitioning, fixed- or variable-length coding, Huffman-style coding, and arithmetic coding (Fang et al., 2017; Yang et al., 2019a; Dai & Cai, 2019; Ziegler et al., 2019). These methods improved the practicality and capacity of linguistic steganography, but generally lacked formal security guarantees. Provably secure steganography (PSS) addresses this limitation by preserving the model's conditional next-token distribution during generation, so that stegotext follows the same distribution as ordinary model sampling (Kaptchuk et al., 2021; Ding et al., 2023; Bai et al., 2025a).

Recent high-capacity PSS schemes have made substantial progress in embedding rate, recovery reliability, and efficiency, including Meteor (Kaptchuk et al., 2021), Discop (Ding et al., 2023), Shimmer (Bai et al., 2025a), FDPSS (Liao et al., 2025), SparSamp (Wang et al., 2025), ANStega (Wang et al., 2026), List-decoding PSS (Pang & Bai, 2026b), and RRC (Yan & Murawaki, 2026). Despite these advances, reliable recovery in high-capacity PSS generally still depends on reconstructing the sender's token distribution. Accurately reconstructing this distribution requires the receiver to use the same model and prompt as the sender.

A.2 Relaxing Receiver-Side Access

Recent work has relaxed receiver-side access along several dimensions, but the resulting methods still rely on either information closely aligned with the sender's generation process or token-only recovery. ADS (Pang & Bai, 2026a) removes the need for explicit token probabilities by allowing the receiver to replay the sender's sampling process. Such replay, however, still requires both parties to access the same seed-controllable model interface and maintain synchronized contexts. Weakly asymmetric steganography (Liao et al., 2026) further reduces receiver-side requirements but assumes exact knowledge of selected properties of the sender's next-token distribution, such as token rankings. These properties may be unavailable when the receiver uses a different model or has only partial context.

In a separate token-only line of work, USLM (Zamir, 2024) recovers symbols of a dynamic error-correcting code from observed tokens using sampling scores, while PRSSA (Bai et al., 2025b) uses shared randomness and statistical testing and Disreo (Jiang et al., 2026) uses synchronized vocabulary permutations and distribution reorganization to recover information from token positions. These methods eliminate the receiver's dependence on the sender's model and prompt. However, they do not use model or context resources that may still be available to the receiver, and existing methods have low embedding rates or limited recovery reliability.

A.3 LLM Watermarking

LLM watermarking embeds signals into the generated text primarily to detect or attribute model-generated content. Early zero-bit methods bias the selection of tokens using vocabulary partitions (Kirchenbauer et al., 2023; Zhao et al., 2024). Later, unbiased or distribution-preserving designs seek to reduce distributional distortion, with some token marginals preserved under specified randomness assumptions (Hu et al., 2024; Wu et al., 2024; Kuditipudi et al., 2024; Xie et al., 2025). These methods determine whether a watermark is present but do not encode multi-bit payloads specified by users.

Multi-bit watermarking further encodes short identifiers or provenance information for robust attribution (Wang et al., 2024; Yoo et al., 2024; Qu et al., 2025; Feng et al., 2025; Jiang et al., 2025a). Its payloads are typically much smaller than those targeted by linguistic steganography, and exact recovery of the entire payload is not always required for effective detection and attribution. More

fundamentally, the two fields differ in purpose, with LLM watermarking supporting detection and attribution of generated text while linguistic steganography targets covert transmission of sender-specified secret payloads.

B Detailed Threat Model

Section 2 defines the communication system, receiver access modes, and security and recovery objectives. Here we specify the warden's observation and knowledge used by the security analysis.

Passive warden We consider a passive probabilistic polynomial-time (PPT) warden $\mathcal{A}$ whose goal is steganographic detection (Hopper et al., 2002; Kaptchuk et al., 2021). The warden observes each complete text transmitted over the public channel. Let $X_{1:T}$ denote the cover sequence, generated by $\mathcal{M}$ under the same prompt x but without payload embedding. The warden seeks to distinguish the PACER stegotext $Y_{1:T}$ from the cover sequence $X_{1:T}$. The public channel delivers the generated text without insertion, deletion, substitution, or rewriting.

Warden knowledge Following Kerckhoffs's principle, $\mathcal{A}$ knows the full design and implementation of PACER, the sender-side model $\mathcal{M}$, the tokenizer and vocabulary, the sampling configuration, receiver mode r, and the receiver-side resources associated with that mode. We further grant it the sender-side prompt or context x and the complete generated history, allowing it to reconstruct P_t and, when available, Q_t. The warden may run the sender's model, apply any efficient statistical test, or train a steganalyzer. The warden's resources do not vary with receiver mode r. The shared key K and its derived pseudorandom values remain secret.

Security criterion Theorem 3.1 states the warden's distinguishing advantage, whose complete proof appears in Section C.1 under the knowledge and channel model above.

C Detailed Proofs

C.1 Provable Security

We prove the distribution-preservation lemma and computational-security theorem stated in Section 3. The proof uses the detailed warden model in Section B.

Proof of Lemma 3.1. Fix a receiver mode, a realized prefix, and the current payload block with true value c^*. PACER uses domain-separated PRF inputs for different payload blocks, token steps, candidates, and pseudorandom choices. After replacing the PRF by a random function, values at distinct inputs are independent and uniform.

First consider token-class coupling applied to a distribution D, and condition on the realized partition G. Since the target class is uniform, the accepted branch contributes

$$\frac{1}{M}\min\{1, M\pi_j\} = \min\{1/M, \pi_j\}$$

to class j. The total remaining mass is

$$\sum_j \big(\pi_j - \min\{1/M, \pi_j\}\big).$$

When this mass is positive, the residual draw selects class j in proportion to $\pi_j - \min\{1/M, \pi_j\}$, so its unconditional contribution is $\pi_j - \min\{1/M, \pi_j\}$; when it is zero, the residual branch is absent. Hence, class j is selected with probability

$$\Pr[j \text{ is selected}] = \min\{1/M, \pi_j\} + \pi_j - \min\{1/M, \pi_j\} = \pi_j.$$

Sampling from D conditioned on the selected class therefore has marginal D. Zero-mass classes are never selected.

In token-only mode, $D = P_t$, so the output follows the sender's token distribution. When receiver-side information is available, an accepted proposal from the receiver-side distribution contributes

$\min\{P_t(v), Q_t(v)\}$ to token v. Upon rejection, token-class coupling samples $P_{t,\mathrm{res}}$ marginally. Thus, for $\omega(P_t, Q_t) < 1$,

$$\Pr[Y_t = v] = \min\{P_t(v), Q_t(v)\} + (1 - \omega(P_t, Q_t))P_{t,\mathrm{res}}(v) = P_t(v).$$

If $\omega(P_t, Q_t) = 1$, then $P_t = Q_t$ and the proposal is accepted almost surely. Hence every receiver mode preserves P_t. □

Proof of Theorem 3.1. In H_0, PACER uses the real PRF to derive all pseudorandom values. In H_1, the domain-separated PRF is replaced by a random function. Any non-negligible distinction between these hybrids yields a distinguisher against the PRF. Thus, for polynomial T, H_0 and H_1 are computationally indistinguishable.

Under H_1, fix a realized prefix $y_{<t}$. Lemma 3.1 gives $Y_t \sim P_t$ for every possible current payload block. Because P_t depends only on $(x, y_{<t})$, conditioning only on the prefix gives

$$\Pr[Y_t = v \mid Y_{<t} = y_{<t}] = P_{\mathcal{M}}(v \mid x, y_{<t}).$$

Applying the chain rule over $t = 1, \ldots, T$ shows that H_1 is identically distributed to the cover sequence $X_{1:T}$. Therefore,

$$\begin{aligned}
&|\Pr[\mathcal{A}(\mathsf{H}_0) = 1] - \Pr[\mathcal{A}(X_{1:T}) = 1]| \\
&\quad \le |\Pr[\mathcal{A}(\mathsf{H}_0) = 1] - \Pr[\mathcal{A}(\mathsf{H}_1) = 1]| \\
&\qquad + |\Pr[\mathcal{A}(\mathsf{H}_1) = 1] - \Pr[\mathcal{A}(X_{1:T}) = 1]| .
\end{aligned}$$

The first term is negligible by PRF security, and the second is zero by the chain-rule argument above. □

C.2 Certified Recovery Guarantees

Under the ideal random-function model of Section C.1 with exact sampling and arithmetic, we first show that the wrong-candidate baselines defined in Section 3.2 are the exact conditional path distributions of incorrect candidates. This yields a valid e-process for each incorrect candidate and a certification error bound for every certification process. We then show that the same bound holds after deterministic retokenization.

Consider a certification process for a block with true value c^*, and let τ denote the token index just before the process begins. Let $\mathcal{F}_t$ contain the generated prefix $Y_{1:t}$, the current block index and retained path counts, and all candidate paths and random-function values revealed through step t. Here $H_t(c)$ and $p_{0,t}$ are the candidate path and wrong-candidate baseline from Section 3.2.

Proof of Lemma 3.2. Fix $\mathcal{F}_{t-1}$ and the observed token Y_t. The sender generates Y_t using the random-function values indexed by the true candidate c^*. For any $c \neq c^*$, the domain separation above makes the current candidate-specific proposal, partition, and target-class values independent and uniform conditional on $\mathcal{F}_{t-1}$ and Y_t.

When Q_t is available, the inverse-CDF interval of Y_t under Q_t has length $q_t = Q_t(Y_t)$. The proposal test for c thus succeeds with probability q_t. Conditional on its failure, the independent class test succeeds with probability $1/M$. The first-match rule consequently gives

$$\begin{aligned}
\Pr[H_t(c) = \mathsf{PM} \mid \mathcal{F}_{t-1}, Y_t] &= q_t, \\
\Pr[H_t(c) = \mathsf{CM} \mid \mathcal{F}_{t-1}, Y_t] &= (1 - q_t)/M, \\
\Pr[H_t(c) = \mathsf{NM} \mid \mathcal{F}_{t-1}, Y_t] &= (1 - q_t)(1 - 1/M).
\end{aligned}$$

In token-only mode, the proposal test is absent, so paths CM and NM have probabilities $1/M$ and $1 - 1/M$, respectively. These are exactly the incorrect-candidate path distributions defined in Section 3.2. Thus, for every incorrect candidate $c \neq c^*$ and every $h \in \mathcal{H}$,

$$\Pr[H_t(c) = h \mid \mathcal{F}_{t-1}, Y_t] = p_{0,t}(h).$$

The KT predictor $g_t(\cdot \mid c)$ uses the path counts available before step t. It is therefore $\mathcal{F}_{t-1}$-measurable, is fixed before $H_t(c)$ is observed, and sums to one over $\mathcal{H}$. The realized path lies in the support of

$p_{0,t}$, so the update is well-defined. Therefore,

$$\mathbb{E}\left[\frac{g_t(H_t(c) \mid c)}{p_{0,t}(H_t(c))} \;\middle|\; \mathcal{F}_{t-1}, Y_t\right] = \sum_{h:p_{0,t}(h)>0} g_t(h \mid c) \le 1.$$

Taking conditional expectation over Y_t and using the tower property gives

$$\begin{aligned}&\mathbb{E}\left[\frac{g_t(H_t(c) \mid c)}{p_{0,t}(H_t(c))} \;\middle|\; \mathcal{F}_{t-1}\right] \\ &\quad = \mathbb{E}\left[\sum_{h:p_{0,t}(h)>0} g_t(h \mid c) \;\middle|\; \mathcal{F}_{t-1}\right] \le 1.\end{aligned}$$

Substituting the evidence update and pulling the $\mathcal{F}_{t-1}$-measurable factor $E_{t-1}(c)$ outside the conditional expectation gives

$$\begin{aligned}\mathbb{E}[E_t(c) \mid \mathcal{F}_{t-1}] &= E_{t-1}(c)\,\mathbb{E}\left[\frac{g_t(H_t(c) \mid c)}{p_{0,t}(H_t(c))} \;\middle|\; \mathcal{F}_{t-1}\right] \\ &\le E_{t-1}(c).\end{aligned}$$

□

Lemma C.1 (Ville's inequality)**.** *Let $\{Z_s\}_{s\ge 0}$ be a nonnegative supermartingale with $Z_0 = 1$. Ville (1939) established that, for any $\gamma > 0$,*

$$\Pr\left[\sup_{s\ge 0} Z_s \ge \gamma\right] \le \frac{1}{\gamma}.$$

Proof of Theorem 3.2. Condition on the realized history $\mathcal{F}_\tau$ at the start of the certification process. By Lemma 3.2, the evidence sequence of each incorrect candidate is a nonnegative supermartingale with $E_\tau(c) = 1$. Applying Lemma C.1 to the evidence sequence of an incorrect candidate, stopped at the end of the current certification process, and setting

$$\gamma = \frac{2^b - 1}{\varepsilon},$$

shows that its probability of crossing the certification threshold, conditional on $\mathcal{F}_\tau$, is at most

$$\frac{\varepsilon}{2^b - 1}.$$

If PACER recovers an incorrect value, at least one of the $2^b - 1$ incorrect candidates must have crossed this threshold. A union bound therefore gives

$$\Pr[\text{PACER recovers some } c \ne c^* \text{ during this process} \mid \mathcal{F}_\tau] \le \varepsilon.$$

Since the bound holds for every process-start history, averaging over $\mathcal{F}_\tau$ proves the claim. □

Proof of Corollary 3.1. Let $\widetilde{Y}_{1:\widetilde{T}}$ denote the receiver's retokenized sequence. Fix an incorrect candidate $c \ne c^*$ and condition on its ideal random-function outputs for the current block. By the candidate and block domain separation, the random-function inputs defining these outputs are distinct from those used for token generation. In the ideal random-function model, these fixed outputs are therefore independent of the random-function values used for token generation. Lemma 3.1 and the chain rule then imply that $Y_{1:T}$, and hence its deterministic retokenization $\widetilde{Y}_{1:\widetilde{T}}$, is independent of the fixed outputs.

Conditioned on the retokenized history, the candidate-specific random-function values remain independent and uniform. Therefore, the same path-distribution calculation used in the proof of Lemma 3.2 gives the wrong-candidate path distribution for $\widetilde{Y}_t$, with Q_t and $p_{0,t}$ computed from the retokenized history. The same e-process argument in Lemma 3.2 and Theorem 3.2 then prove the claim. □

D Additional Experimental Details

Section 4 defines the metrics, datasets, model pairs, receiver modes, baselines, and default PACER parameters. Unless otherwise stated, auxiliary experiments use C4 and the Llama3 model family. The steganalysis experiments use 10,000 labeled samples per class, split into training, validation, and test sets at a ratio of 3:1:1.

In addition to the sampling parameters reported in the main text, all methods use top-$p = 1.0$ (Holtzman et al., 2020). All methods run on one NVIDIA RTX 4090 GPU (24 GB) and an Intel Xeon Silver 4314 CPU (16 cores, 32 threads). Method-specific baseline configurations are given in Appendix F.

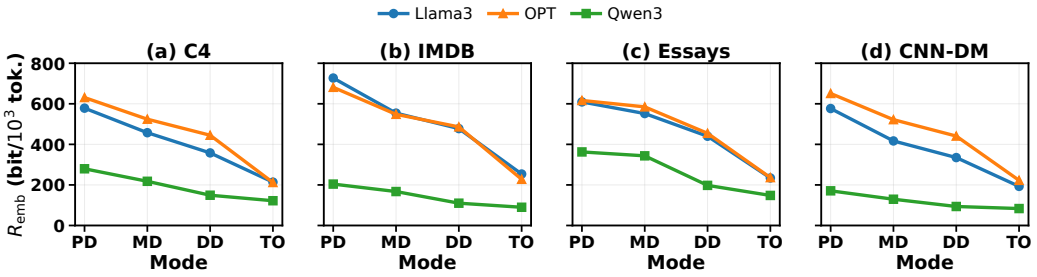


Figure 6: Embedding rate across datasets and model families.

E Qualitative Generation Examples

Table 6: Examples of stegotexts generated by Llama-3.1-8B.

Method	Generation
Prompt. The letter was hidden between the pages of an old village ledger. Clara unfolded it carefully by the window, watching the afternoon light move across the faded ink. For a moment, the room was silent except for the rain against the glass.	
Random Sampling	This was the first letter addressed to her, though there had been many sent to a Miss S. W. Bell, sister of the late William Bell of the Parish of Saint Giles. The last letter had been addressed from the Home. He was here…
PACER-PD	She wondered if this was how the letter had felt when it passed through her father's hands. She stared at the signature at the end. Clara's hand was shaking slightly as she turned back to her father's letter. There, I see the two of you…
PACER-DD	She had often tried to imagine what a letter from her mother would look like. On the one hand, it felt like a last chance to have that conversation, the one where she would finally get to be honest with her mother…
PACER-MD	Her hand trembling, Clara looked back at the words. Dearest Clara, it began. My heart is filled with joy and a sadness beyond measure. I pray you will forgive me for not waiting longer for your reply…
PACER-TO	Clara's heartbeat quickened and the paper trembled in her hand. Dear Clara, This letter will come with a bundle of letters from the dead. It is time for you to read them. They have passed one winter after the other…

F Baseline Configurations

Discop (Ding et al., 2023), Meteor (Kaptchuk et al., 2021), PRSSA (Bai et al., 2025b), USLM (Zamir, 2024), MPAC (Yoo et al., 2024), and BiMark (Feng et al., 2025) use their open-source implementations. Disreo (Jiang et al., 2026) is reproduced from its paper, and its results follow the trend reported therein. For the symmetric baselines, the sender and receiver use the same model, tokenizer, prompt, sampling configuration, and synchronized random state required by each algorithm.

Discop (symmetric reference) We use its high-capacity distribution-copy construction based on Huffman trees, with synchronized randomness for token-by-token encoding and decoding.

Meteor (symmetric reference) We use its arithmetic-coding implementation that directly embeds a bitstream, with 32-bit arithmetic precision and probability sorting enabled.

Disreo (token-only) We use its token-only implementation based on synchronized vocabulary permutation and distribution reorganization, following the default configuration in the paper. We exclude the self-checking variant that regenerates text with a modified prompt because this procedure is not covered by its provable-security analysis.

PRSSA (token-only) We use its embedding and recovery method based on distribution permutation and sequential hypothesis testing, together with the default statistic, stopping parameters, and standard model-free decoder.

USLM (token-only) We use its single-query construction based on keyed sampling scores and dynamic error-correcting codes. The decision threshold is $t = 2$, matching the paper's experimental setting.

BiMark (watermarking, token-only) We use its keyed vocabulary-partitioning construction with maximum bias strength $\delta = 1.0$ and distribution constraint $\alpha = 1.0$, which embeds message bits while preserving the token distribution in expectation.

MPAC (watermarking, token-only) We use its multi-bit construction that maps message symbols to keyed vocabulary chunks and biases generation toward the corresponding chunk, with $\gamma = 0.25$, base $= 4$, and logit bias $\delta = 3.0$.

G Token-Class Sensitivity

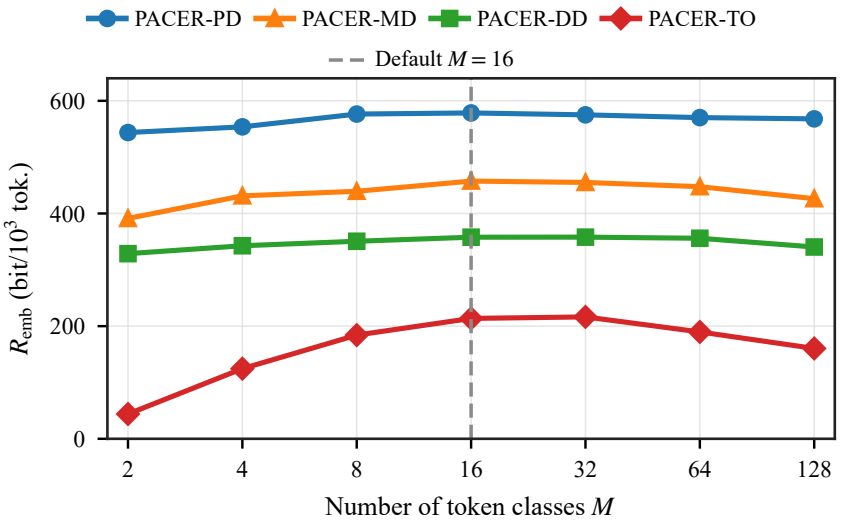


Figure 7: Embedding rate versus the number of token classes M. The dashed vertical line marks the default $M = 16$.

Figure 7 shows the effect of the number of token classes M. PACER-PD, PACER-MD, and PACER-DD exhibit only modest variation, while PACER-TO increases rapidly at smaller M and declines at larger M. PACER-PD, PACER-MD, and PACER-DD use both proposal and class paths, whereas PACER-TO relies entirely on class matches and is therefore more sensitive to M. The default $M = 16$ provides a high, stable embedding rate across all four modes while avoiding the decline at larger values.

H ALGORITHM

Algorithm 1: Distribution-preserving token embedding

Input: Sender distribution P_t, payload block c^*, key K, mode r, class count M, and Q_t if $r \neq \mathsf{TO}$
Output: Token Y_t

if $r = \mathsf{TO}$ **then**
  $D \leftarrow P_t$
else
  $z \leftarrow$ inverse-CDF sample from Q_t for c^*
  if *accept* z *with probability* $\min\{1, P_t(z)/Q_t(z)\}$ **then**
    return z
  $D \leftarrow P_{t,\mathrm{res}}$
derive token partition G and uniform target class a for c^*
$\pi_j \leftarrow \sum_{v:G(v)=j} D(v), \quad j = 0, \ldots, M-1$
if *accept* a *with probability* $\min\{1, M\pi_a\}$ **then**
  $J \leftarrow a$
else
  draw J in proportion to $\pi_j - \min\{\pi_j, 1/M\}$
return an inverse-CDF sample from $D(\cdot \mid G(\cdot) = J)$

Algorithm 2: Certified payload recovery

Input: Observed tokens, key K, mode r, and public parameters (b, M, ε)
Output: Certified payload blocks

if $r = \mathsf{TO}$ **then**
  $\mathcal{H} \leftarrow \mathcal{H}^{\mathrm{TO}}$
else
  $\mathcal{H} \leftarrow \mathcal{H}^{Q}$
initialize every candidate's path counts to zero
$E(c) \leftarrow 1$ for every $c \in \mathcal{C}$
foreach *successive observed token* Y_t **do**
  if $r \neq \mathsf{TO}$ **then**
    construct Q_t from the receiver's available resources
  compute $p_{0,t}$
  foreach $c \in \mathcal{C}$ **do**
    compute $g_t(\cdot \mid c)$ from the current path counts
    $H_t(c) \leftarrow$ the first-match path of Y_t for c
    $E(c) \leftarrow E(c)\dfrac{g_t(H_t(c) \mid c)}{p_{0,t}(H_t(c))}$
    increment the count of $H_t(c)$
  if *no* $c \in \mathcal{C}$ *satisfies* $E(c) \geq (2^b - 1)/\varepsilon$ **then**
    continue with the next token
  else if *exactly one* $c \in \mathcal{C}$ *satisfies the threshold* **then**
    append c to the certified payload blocks
    use c's cumulative path counts to initialize every candidate in the next block
    reset every candidate's e-value to one
    advance to the next block
  else
    terminate without recovering the current block
    return the certified payload blocks
return the certified payload blocks