S-Stega: Restoring Temporal Stability in Generative Language Steganography via Salt Randomization

Anonymous ACL submission

Abstract

We propose **S-Stega**, a salt-randomized preprocessing framework for generative steganography that mitigates temporal instability arising from fixed or semi-fixed bit-consumption schemes. We show that deterministic encoders can induce reproducible, prefix-aligned regularitiesmanifesting as prefix leakage and time-structured driftthereby motivating the **Chosen-Secret-Message Adversary (CSMA)** and an **IND-CSMA** security notion. Under ideal random and pseudorandom assumptions, we prove that salting masks message bits in both statistical and computational senses, suppressing prefix-aligned leakage and attenuating temporal drift. In practice, **S-Stega** XORs a domain-separated pseudorandom salt into the bit stream prior to embedding, preserving lossless decoding and providing a codec-agnostic hardening layer for HC, AC, and ANS. Experiments across multiple LMs and four datasets show that salting drives prefix agreement near zero, weakens short-range temporal dependence, reduces trajectory separability by 40%–70%, and lowers time-sensitive steganalysis accuracy by roughly 30%, while largely preserving text quality with method-dependent effects on embedding rate.

1 Introduction

Language steganography (LS) enables covert communication by embedding secret bits into seemingly natural text, aiming to preserve both imperceptibility and security (Yang et al., 2020b). As these techniques mature, their potential misuse has drawn increasing attention, motivating extensive steganalysis research. Most existing steganalysis methods detect stego texts by measuring distributional discrepancies between stego and cover texts (Yang et al., 2021b; Xu et al., 2021; Xiang et al., 2022; Guo et al., 2022). This discrepancy is typically expressed as a statistical distance between the generation distributions of stego and cover texts. To capture this discrepancy, prior detectors instantiate it across different feature spaces and modeling paradigms, ranging from handcrafted statistical features (Nechta and Fionov, 2011) to neural discriminators based on RNN/LSTM/Transformer architectures and pretrained language model (LM) (Yang et al., 2018b; Bao et al., 2022; Luo et al., 2024), as well as analyses based on behavioral consistency and generation perturbations (Yang et al., 2021a; Yi et al., 2022).

Despite their effectiveness, most existing steganalysis methods treat generated stego texts as static objects and thus largely overlook the temporal dynamics inherent in autoregressive text generation. In practice, stego texts are produced token by token, where each generation step is influenced not only by the accumulated prefix context but also by a sliding window of control bits. Under this setting, mainstream LS methods (e.g., HC (Yang et al., 2018a), AC (Ziegler et al., 2019), SAAC (Shen et al., 2020), ADG (Zhang et al., 2021), and ANS (Liu et al., 2025)) typically adopt fixed or semi-fixed bit-wise consumption schemes, aligning secret-bit windows with generation steps over time. Such semi-fixed bit-windowto-token associations increase the sensitivity of the generation process to even minor changes in the control bits, potentially introducing time-evolving behaviors in the generation trajectory; in contrast, ADG-style methods largely avoid this issue due to randomized token selection within each bit group.

Tab. 1 presents qualitative examples showing that even tiny perturbations to suffix bits can yield coherent, directional changes in the continuation. Such deviations can propagate through autoregressive generation and accumulate into stable temporal patterns. To capture this effect, we incrementally append secret bits and project sentence-level representations to form PCA trajectories; as shown in Fig. 1, small suffix-bit changes induce

Table 1: Qualitative examples of stego texts under extended secret-bit prefixes. The secret-bit common prefix is fixed; appending 8 bits (and then another 8) yields coherent yet directional continuation shifts.

ID	$s_{\leq t-1} =$ `01...00`	$s_{\leq t} = s_{\leq t-1} \| \|$ `01110010`	$s_{\leq t+1} = s_{\leq t} \| \|$ `01100101`
1	He signed the contract without hesitation and folded the paper into his pocket. **- - -**	He signed the contract without hesitation and folded the paper into his pocket. **He believed the promise would protect his family.**	He signed the contract without hesitation and folded the paper into his pocket. **He believed the promise was a trap, yet he signed anyway to avoid suspicion.**
2	She stared at the empty street and waited for the sound of footsteps that never came. **- - -**	She stared at the empty street and waited for the sound of footsteps that never came. **The silence felt rehearsed in that narrow alley.**	She stared at the empty street and waited for the sound of footsteps that never came. **The silence felt rehearsed and deliberate, as if someone had planned the pause to unsettle her.**

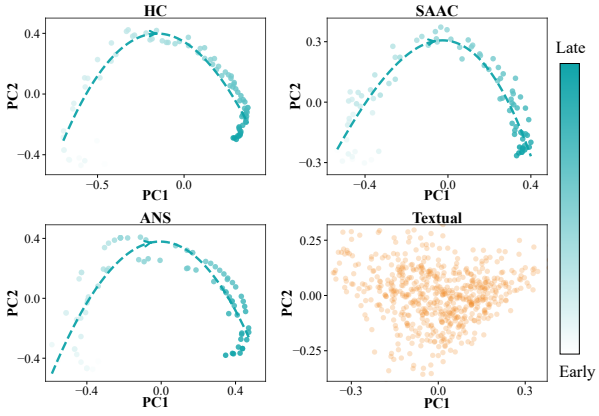

Figure 1: Sentence-level PCA trajectories of stego and cover texts under different encoding strategies.

consistent shifts in representation space, providing an intuitive view of prefix leakage.

This phenomenon reveals a previously overlooked structural risk: when the encoding process is tightly coupled with autoregressive generation dynamics, localized changes in the secret bits can propagate and amplify across generation steps, ultimately manifesting as externally observable temporal patterns in the output. This suggests that temporal consistency during generation constitutes a critical yet underexplored dimension of steganographic security.

To mitigate this structural risk, we propose **S-Stega**, a simple yet effective framework that introduces controlled randomness *prior* to embedding, thereby weakening the tight coupling between secret bits and generation trajectories. Concretely, S-Stega injects a random salt into the secret message, disrupting structured and low-entropy bit patterns and producing pseudo-randomized bit streams for embedding. Empirically, we find that salting substantially reduces prefix leakage and suppresses temporally structured deviations, while preserving competitive text quality and payload.

Contributions. i) First, we characterize *prefix-aligned temporal leakage* in mainstream generative steganography and show that fixed or semi-fixed bit-consumption schemes can induce reproducible, time-structured regularities during autoregressive generation. ii) Second, we introduce a CSMA-oriented evaluation protocol and empirically demonstrate that such temporal regularities are exploitable by both neural steganalyzers and a complementary key-aided verifier stress test. iii) Finally, we propose **S-Stega**, a lightweight, model-agnostic bit-side salting framework that breaks deterministic coupling via domain-separated pseudorandom salts, substantially reducing temporal leakage while largely preserving text quality, with method-dependent effects on embedding rate.

2 Chosen-Secret-Message Security (IND-CSMA)

We focus on a setting motivated by interactive or repeated-use deployments (e.g., APIs), where an adversary may trigger multiple embeddings and adaptively choose payloads to probe for systematic, prefix-aligned drift.

We consider an adaptive chosen-secret-message adversary that can query a generation oracle with secrets of its choice and observe the resulting outputs. This captures distinguishability risks arising from fixed or semi-fixed encoders under autoregressive amplification.

Experiment $\mathsf{Exp}^{\mathsf{IND\text{-}CSMA}}_{\Pi,\mathcal{A}}(b)$**.** A hidden bit $b \in \{0,1\}$ selects between the cover world ($b = 0$) and

the stego world ($b = 1$). The adversary $\mathcal{A}$ makes up to q adaptive queries $x_j \in \mathcal{X}$ to an oracle $\mathcal{O}_b$, which returns either $c_j \leftarrow \mathcal{C}$ (if $b = 0$) or $s_j \leftarrow \mathsf{Emb}_k(x_j, c_j)$ for fresh $c_j \leftarrow \mathcal{C}$ (if $b = 1$). Finally, $\mathcal{A}$ outputs b'.

Advantage.

$$\mathsf{Adv}^{\mathsf{IND\text{-}CSMA}}_{\Pi,\mathcal{A}}(\kappa) = \left|\Pr[b' = b] - \frac{1}{2}\right|. \quad (1)$$

We next present a salt-randomized encoder and show that its induced masking property suffices to bound this advantage.

3 Salt-Randomized Encoding

3.1 Breaking Deterministic Coupling

Bit-wise consumption schemes deterministically couple secret-bit windows to token selection, allowing local bit patterns to steer generation and accumulate into stable, time-evolving biases—especially under adaptive secret choices.

To break this coupling, we inject a fresh random salt at each step. Under *ideal* salting,

$$r_t \overset{\text{i.i.d.}}{\sim} \mathsf{Unif}(\mathcal{R}_t), \quad s_t \sim \mathbf{P}^t_{\mathcal{S}}(\cdot \mid C_t, x_t, r_t), \quad (2)$$

which weakens the step-wise correspondence between secret bits and generation trajectories.

3.2 S-Stega: Bit-Side Salting

We implement S-Stega with a shared key and domain separation to prevent salt reuse across texts/sessions. The payload is a z-bit string, partitioned into T windows $X = (x_1, \ldots, x_T)$ with $x_t \in \{0,1\}^{w_t}$ and $\sum_{t=1}^{T} w_t = z$. At step t, the salt space is $\mathcal{R}_t = \{0,1\}^{w_t}$, and the joint salt space is $\mathcal{R} = \prod_{t=1}^{T} \mathcal{R}_t$.

Entropy-driven windowing. We set the per-step bit budget using the min-entropy of the next-token distribution. Let $p_{\max}(t) = \max_v P(v \mid C_t)$ and $H_\infty(t) = -\log_2 p_{\max}(t)$. We choose

$$w_t = \min\Big\{w_{\max},\ \max\big\{0,\ \lfloor H_\infty(t) - \delta \rfloor\big\}\Big\}. \quad (3)$$

where $\delta \geq 0$ is a safety margin and $w_{\max}$ caps the rate. This reduces embedding on peaked (low-entropy) steps and increases it when the distribution is flatter, while remaining synchronizable given a shared LM and fixed decoding settings.[1]

[1]For fixed-rate encoders (e.g., ANS-style), we use a constant w chosen from a conservative min-entropy budget.

Salt generation. Ideally, $R \sim \mathsf{Unif}(\mathcal{R})$. In practice, both parties derive salts from a long-term key K and a public per-text nonce ν: define $K_\nu = \mathsf{PRF}(K, \nu)$ and generate

$$r_t = \mathsf{PRG}(K_\nu, t)[1\!:\!w_t], \quad (4)$$

i.e., counter-mode expansion domain-separated by t and truncated to w_t bits. This ensures independence across texts/sessions and avoids two-time-pad style correlations.

Encoding and decoding. At step t, we salt via $z_t = x_t \oplus r_t$ and sample

$$s_t \sim \mathbf{P}^t_{\mathcal{S}}(\cdot \mid C_t, z_t), \quad (5)$$

yielding $S = (s_1, \ldots, s_T)$. Since XOR is bijective, a decoder with the same (K, ν) recomputes r_t and inverts z_t to recover x_t, preserving lossless decodability.

3.3 IND-CSMA via Masking

We state a sufficient condition under which salting bounds IND-CSMA distinguishability. Intuitively, if salting makes the per-step control variables nearly input-independent, the generator cannot reliably amplify secret-bit differences into prefix-aligned trajectory divergence. The condition below is sufficient (not necessary).

Let $\sigma : \{0,1\}^z \times \mathcal{R} \to \{0,1\}^z$ be a salting function and $R \sim \mathsf{Unif}(\mathcal{R})$. We say that σ satisfies ε_σ-masking if for all $x, x' \in \{0,1\}^z$,

$$\big\|\mathsf{Law}(\sigma(x, R)) - \mathsf{Law}(\sigma(x', R))\big\|_{\mathrm{TV}} \leq \varepsilon_\sigma, \quad (6)$$

where $\|\cdot\|_{\mathrm{TV}}$ denotes total variation distance. In particular, $\varepsilon_\sigma = 0$ means the salted control distribution is input-independent.

In S-Stega, σ is window-wise XOR: $\sigma_t(x_t, r_t) = x_t \oplus r_t$. When window sizes vary, let $W = (w_1, \ldots, w_T)$ be the (context-dependent) schedule determined by the prefix C_t and fixed decoding settings (hence independent of the payload). With $\mathcal{R}_t = \{0,1\}^{w_t}$ and $R_t \sim \mathsf{Unif}(\mathcal{R}_t)$, conditioning on W yields perfect masking: for any x_t, x'_t, $\sigma_t(x_t, R_t) \equiv \sigma_t(x'_t, R_t) \equiv \mathsf{Unif}(\{0,1\}^{w_t})$, so $\varepsilon_\sigma = 0$. With PRG-generated salts (Sec. 3.2), the same holds computationally. A standard hybrid argument (Goldreich, 2004) then bounds the IND-CSMA advantage by the PRG distinguishing advantage (and by ε_σ in the information-theoretic case), so ε_σ-masking suffices for IND-CSMA security.

3.4 Consequences of ε_σ-Masking

We summarize two observable consequences of ε_σ-masking from two complementary perspectives. (i) *Implication I* is a **horizontal** (cross-secret) comparison: for two secrets that share the same prefix and differ only in suffix bits, it asks whether the early generated prefix remains distinguishable, capturing prefix leakage. (ii) *Implication II* is a **vertical** (within-generation) comparison: within a single generation process, it asks whether the generation dynamics exhibit abnormal step-wise drift over time, capturing temporal stability.

Implication I: Suppressing Prefix Leakage.

Corollary 1 (Prefix Leakage Suppression)**.** *Let $X, X' \in \{0,1\}^z$ be two secret messages that may share identical prefix bits. Let $S_{\le\tau}$ and $S'_{\le\tau}$ denote the length-τ prefixes of the stego texts generated by $\mathsf{Emb}(X,k,R)$ and $\mathsf{Emb}(X',k,R')$, respectively, where R and R' are sampled independently from $\mathsf{Unif}(\mathcal{R})$. If σ satisfies ε_σ-masking, then*

$$\left\|\mathbf{P}_{S_{\le\tau}|X} - \mathbf{P}_{S'_{\le\tau}|X'}\right\|_{\mathrm{TV}} \le \tau\,\varepsilon_\sigma. \tag{7}$$

(Proof in Appendix A.)

This shows that salting suppresses the dependence between stego prefixes and secret-bit prefixes, bounding prefix leakage by $\mathcal{O}(\tau\varepsilon_\sigma)$.

Implication II: Restoring Temporal Stability. Let $\mathbf{P}^t_{\mathcal{S}}(\cdot \mid C_t, X, R)$ denote the stego next-token distribution at step t, and let $\mathbf{P}^t_{\mathcal{C}}(\cdot) \triangleq \mathbf{P}_{\mathcal{C}}(\cdot \mid C_t)$ denote the cover distribution. Define the salt-averaged stego distribution $\widetilde{\mathbf{P}}^t_{\mathcal{S}}(\cdot) \triangleq \mathbb{E}_R[\mathbf{P}^t_{\mathcal{S}}(\cdot \mid C_t, X, R)]$ and the temporal drift

$$\Delta\mathbf{P}^t_{\mathcal{S}} \triangleq \left\|\widetilde{\mathbf{P}}^t_{\mathcal{S}} - \widetilde{\mathbf{P}}^{t-1}_{\mathcal{S}}\right\|_{\mathrm{TV}}, \tag{8}$$

with $\bar{\Delta}^t \triangleq \mathbb{E}[\Delta\mathbf{P}^t_{\mathcal{S}}]$.

Corollary 2 (Temporal Stability Restoration)**.** *Assume that for all t,*

$$\mathbb{E}_R\left[\left\|\mathbf{P}^t_{\mathcal{S}}(\cdot \mid C_t, X, R) - \mathbf{P}^t_{\mathcal{C}}(\cdot)\right\|_{\mathrm{TV}}\right] \le \varepsilon_\pi. \tag{9}$$

Then

$$\bar{\Delta}^t \le \left\|\mathbf{P}^t_{\mathcal{C}} - \mathbf{P}^{t-1}_{\mathcal{C}}\right\|_{\mathrm{TV}} + 2\varepsilon_\pi. \tag{10}$$

(Proof in Appendix B.)

Interpreting (10), salting compresses the stego drift to the cover model's intrinsic step-to-step variation up to an additive $2\varepsilon_\pi$, restoring temporal stability in a statistical sense. Quantitative validation is provided in Sec. 4.4.

4 Experimental Results and Analysis

4.1 Experimental Setup

1) Dataset Setup We evaluate S-Stega on four benchmark text datasets spanning diverse domains and writing styles: **Twitter** (Go et al., 2009), **IMDB** (Maas et al., 2011), **News** (Hermann et al., 2015), and **Covid-19** (Wang et al., 2020).

2) Baselines To assess salting, we integrate S-Stega into representative neural LS encoders and compare each salted variant with its unsalted counterpart under identical settings. We consider three widely used baselines: Huffman-coding HC (Yang et al., 2018a), arithmetic-coding AC (Ziegler et al., 2019), and asymmetric numeral systems ANS (Liu et al., 2025); we also include ADG (Zhang et al., 2021) as a token-side randomization baseline.

S-Stega is encoder-agnostic; for each baseline, we compare the original encoder to its salted variant under matched settings.

3) Parameter Settings We use pretrained LMs (GPT-2 medium, LLaMA-3.1-8B, Qwen-3-8B) with no fine-tuning, implemented in PyTorch/-Transformers, and set the candidate pool size to $K = 900$ for all encoding algorithms.

4.2 Prefix Leakage Evaluation

We assess prefix leakage by comparing stego texts generated from message pairs (X, X') that differ only in their final ν bits. Prefix-level similarity is measured by the *prefix agreement rate (PAR)*, defined as the probability that two stego texts share an identical prefix of length τ:

$$\mathrm{PAR}(\tau) = \Pr\left[S_{1:\tau} = S'_{1:\tau}\right]. \tag{11}$$

We estimate $\mathrm{PAR}(\tau)$ via Monte Carlo averaging over N independent trials.

Fig. 2 reports PAR for three representative encoders across bit-window lengths w and suffix perturbation sizes ν. Without salting, all encoders exhibit non-negligible prefix agreement, but with distinct regimes. For HC, PAR drops sharply as ν approaches w, indicating strong sensitivity to suffix perturbations. AC shows a smoother decay, consistent with greater LM dominance in early steps. ANS exhibits a hard-threshold pattern: prefixes match when $\nu < w$ and diverge immediately when $\nu \ge w$, aligning with its finite-state update structure.

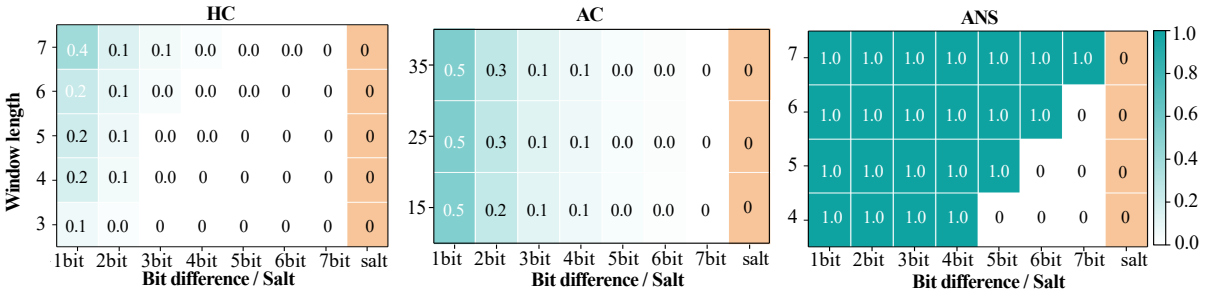

Figure 2: Estimated prefix agreement rate PAR of HC, AC, and ANS under suffix perturbations of ν bits on the Twitter dataset. Results on IMDB, News, and Covid-19 are reported in Appendix E.

Table 2: Coefficient of Variation (CV) of KL divergence across four datasets (higher is better).

Model	Method	Twitter	IMDB	News	Covid-19
GPT-2	$\text{HC}_{\text{(no-salt)}}$	0.308	0.313	0.335	0.326
	$\text{AC}_{\text{(no-salt)}}$	0.962	0.945	1.047	0.981
	$\text{ANS}_{\text{(no-salt)}}$	2.108	2.259	2.433	2.259
	$\text{HC}_{\text{(salted)}}$	**0.333**	**0.339**	**0.364**	**0.354**
	$\text{AC}_{\text{(salted)}}$	**1.239**	**1.263**	**1.338**	**1.276**
	$\text{ANS}_{\text{(salted)}}$	**2.259**	**2.433**	**2.635**	**2.433**
LLaMA-3	$\text{HC}_{\text{(no-salt)}}$	0.254	0.260	0.275	0.265
	$\text{AC}_{\text{(no-salt)}}$	0.812	0.805	0.840	0.825
	$\text{ANS}_{\text{(no-salt)}}$	1.626	1.751	1.924	1.852
	$\text{HC}_{\text{(salted)}}$	**0.419**	**0.425**	**0.438**	**0.430**
	$\text{AC}_{\text{(salted)}}$	**1.330**	**1.355**	**1.453**	**1.401**
	$\text{ANS}_{\text{(salted)}}$	**2.410**	**2.553**	**2.759**	**2.633**
Qwen-3	$\text{HC}_{\text{(no-salt)}}$	0.194	0.200	0.215	0.208
	$\text{AC}_{\text{(no-salt)}}$	0.342	0.335	0.380	0.365
	$\text{ANS}_{\text{(no-salt)}}$	1.213	1.323	1.487	1.402
	$\text{HC}_{\text{(salted)}}$	**0.332**	**0.338**	**0.360**	**0.350**
	$\text{AC}_{\text{(salted)}}$	**1.562**	**1.593**	**1.704**	**1.658**
	$\text{ANS}_{\text{(salted)}}$	**2.192**	**2.050**	**2.382**	**2.257**

Table 3: Lag-1 autocorrelation $\rho(1)$ of step-wise KL divergence across four datasets (lower is better).

Model	Method	Twitter	IMDB	News	Covid-19
GPT-2	$\text{HC}_{\text{(no-salt)}}$	0.989	0.957	0.912	0.978
	$\text{AC}_{\text{(no-salt)}}$	0.735	0.789	0.672	0.751
	$\text{ANS}_{\text{(no-salt)}}$	0.865	0.898	0.803	0.932
	$\text{HC}_{\text{(salted)}}$	**0.119**	**0.128**	**0.111**	**0.108**
	$\text{AC}_{\text{(salted)}}$	**0.451**	**0.485**	**0.412**	**0.467**
	$\text{ANS}_{\text{(salted)}}$	**0.085**	**0.091**	**0.078**	**0.082**
LLaMA-3	$\text{HC}_{\text{(no-salt)}}$	0.977	0.991	0.885	0.934
	$\text{AC}_{\text{(no-salt)}}$	0.717	0.759	0.651	0.738
	$\text{ANS}_{\text{(no-salt)}}$	0.794	0.841	0.725	0.829
	$\text{HC}_{\text{(salted)}}$	**0.238**	**0.255**	**0.221**	**0.247**
	$\text{AC}_{\text{(salted)}}$	**0.392**	**0.421**	**0.360**	**0.385**
	$\text{ANS}_{\text{(salted)}}$	**0.072**	**0.076**	**0.066**	**0.070**
Qwen-3	$\text{HC}_{\text{(no-salt)}}$	0.989	0.963	0.901	0.975
	$\text{AC}_{\text{(no-salt)}}$	0.921	0.945	0.835	0.888
	$\text{ANS}_{\text{(no-salt)}}$	0.981	0.941	0.895	0.966
	$\text{HC}_{\text{(salted)}}$	**0.008**	**0.009**	**0.007**	**0.008**
	$\text{AC}_{\text{(salted)}}$	**0.002**	**0.002**	**0.002**	**0.001**
	$\text{ANS}_{\text{(salted)}}$	**0.004**	**0.004**	**0.003**	**0.004**

With salting enabled, PAR is near zero throughout, suggesting that salting substantially reduces the coupling between secret bits and early-generation trajectories.

4.3 Temporal Dynamics

We measure distributional deviation during steganographic generation using the step-wise KL divergence D_{KL}^t between the steganographic distribution and the underlying LM. To characterize its temporal behavior, we use two complementary statistics: (i) *the coefficient of variation (CV)*, capturing relative fluctuation magnitude, and (ii) *the lag-1 autocorrelation* $\rho(1)$, capturing short-range temporal dependence:

$$\text{CV} = \frac{\sqrt{\text{Var}_t[D_{\text{KL}}^t]}}{\overline{D}_{\text{KL}}}, \quad \rho(1) = \text{corr}\big(D_{\text{KL}}^t, D_{\text{KL}}^{t-1}\big). \tag{12}$$

Larger CV means stronger relative fluctuations, while smaller $\rho(1)$ means weaker adjacent-step dependence.

Tab. 2–3 report CV and $\rho(1)$ for HC, AC, and ANS across four datasets. Without salting, HC and AC typically exhibit lower CV together with high $\rho(1)$, indicating deviations that are relatively stable in magnitude yet strongly persistent over time; ANS often shows higher CV due to its smaller mean deviation. After salting, CV generally increases while $\rho(1)$ drops markedly, especially for HC and AC, suggesting that salting both amplifies relative step-to-step fluctuations and reduces short-range temporal dependence. This is consistent with salting randomizing the effective bit-side control at each step, thereby perturbing the otherwise deterministic update patterns induced by fixed encoding rules (e.g., Huffman trees or interval partitioning).

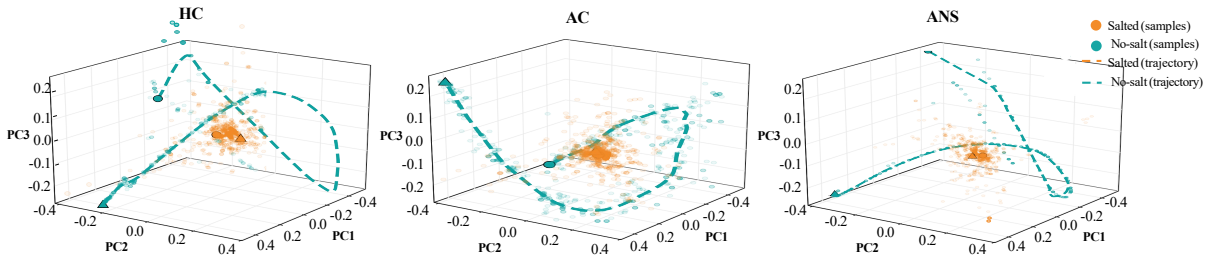

Figure 3: PCA of sentence-level trajectories for three encoders, with vs. without salting.

4.4 Embedding-Space Geometry

We analyze temporal behavior in representation space at two levels: sentence-level representations and token-level hidden states.

Sentence-level trajectories. We incrementally embed secret bits (8 bits/step) and encode each intermediate sentence as a TF–IDF vector; the resulting sequence forms a trajectory, which we visualize via 3D PCA. Fig. 3 shows that unsalted trajectories are more compact and smooth, whereas salting yields more dispersed, natural-text-like geometry.

Token-level hidden-state trajectories. We further analyze token-level dynamics using hidden states. Let $u_t(\gamma)$ be the (normalized) representation at step t for sample γ. We quantify cross-sample alignment via *trajectory variance*:

$$\mathrm{TrajVar}(\mathcal{C}) = \frac{1}{T_{\max}} \sum_{t=1}^{T_{\max}} \frac{1}{|\mathcal{C}_t|} \sum_{\gamma \in \mathcal{C}_t} \|u_t(\gamma) - \bar{u}_t\|_2^2, \tag{13}$$

where $\bar{u}_t$ is the mean at step t and $\mathcal{C}_t$ contains samples with length at least t. Larger TrajVar indicates weaker alignment. To measure geometric separability between two sets, we compute the average inter-class distance:

$$d_{\mathrm{traj}}(\mathcal{C}_i, \mathcal{C}_{\mathrm{cover}}) = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{|\mathcal{C}_i||\mathcal{C}_{\mathrm{cover}}|} \sum_{\gamma \in \mathcal{C}_i} \sum_{\gamma' \in \mathcal{C}_{\mathrm{cover}}} \|u_t(\gamma) - u_t(\gamma')\|_2, \tag{14}$$

here, $\mathcal{C}_{\mathrm{cover}}$ denotes trajectories from cover (non-embedded) texts.

Hidden-state results are consistent with the sentence-level trends. Fig. 4 shows that unsalted encoders exhibit more coherent and repeatable drift, whereas salting reduces such structure and increases step-wise dispersion (higher TrajVar). Tab. 4 further indicates reduced separability, with d_{traj} decreasing by 40%–70% across datasets.

Table 4: Trajectory separation (d_{traj}) across four datasets (lower is better).

Model	Method	Twitter	IMDB	News	Covid-19
GPT-2	$\mathrm{HC}_{(\text{no-salt})}$	0.461	0.452	0.439	0.468
	$\mathrm{AC}_{(\text{no-salt})}$	0.367	0.359	0.348	0.377
	$\mathrm{ANS}_{(\text{no-salt})}$	0.537	0.524	0.511	0.548
	$\mathrm{HC}_{(\text{salted})}$	**0.179**	**0.185**	**0.173**	**0.191**
	$\mathrm{AC}_{(\text{salted})}$	**0.162**	**0.168**	**0.159**	**0.171**
	$\mathrm{ANS}_{(\text{salted})}$	**0.172**	**0.178**	**0.167**	**0.183**
LLaMA-3	$\mathrm{HC}_{(\text{no-salt})}$	0.421	0.435	0.410	0.452
	$\mathrm{AC}_{(\text{no-salt})}$	0.353	0.365	0.342	0.385
	$\mathrm{ANS}_{(\text{no-salt})}$	0.289	0.305	0.275	0.315
	$\mathrm{HC}_{(\text{salted})}$	**0.131**	**0.139**	**0.125**	**0.143**
	$\mathrm{AC}_{(\text{salted})}$	**0.137**	**0.145**	**0.130**	**0.149**
	$\mathrm{ANS}_{(\text{salted})}$	**0.150**	**0.160**	**0.142**	**0.163**
Qwen-3	$\mathrm{HC}_{(\text{no-salt})}$	0.380	0.395	0.360	0.410
	$\mathrm{AC}_{(\text{no-salt})}$	0.299	0.315	0.282	0.324
	$\mathrm{ANS}_{(\text{no-salt})}$	0.250	0.265	0.237	0.270
	$\mathrm{HC}_{(\text{salted})}$	**0.129**	**0.138**	**0.122**	**0.140**
	$\mathrm{AC}_{(\text{salted})}$	**0.124**	**0.135**	**0.118**	**0.133**
	$\mathrm{ANS}_{(\text{salted})}$	**0.133**	**0.142**	**0.125**	**0.145**

4.5 Quality and Capacity

To assess the impact of random salting on both text quality and embedding efficiency, we evaluate stego texts using *perplexity (PPL)* under a reference LM and *the embedding rate (ER)*.

Fig. 5 (left) shows that salting has method-dependent effects on text quality. HC exhibits a slight decrease in PPL after salting, suggesting no degradationand potentially a mild improvementin fluency. In contrast, AC shifts to higher PPL values under salting, indicating a modest loss in naturalness along with increased variability. ANS remains largely unchanged, with near-overlapping PPL distributions across the two settings.

Fig. 5 (right) reports the corresponding embedding efficiency. HC shows a slight decrease in ER after salting, while AC shifts to higher ER values, indicating increased average embedding capacity under salting (with some additional variability). ANS remains essentially unchanged, with tightly

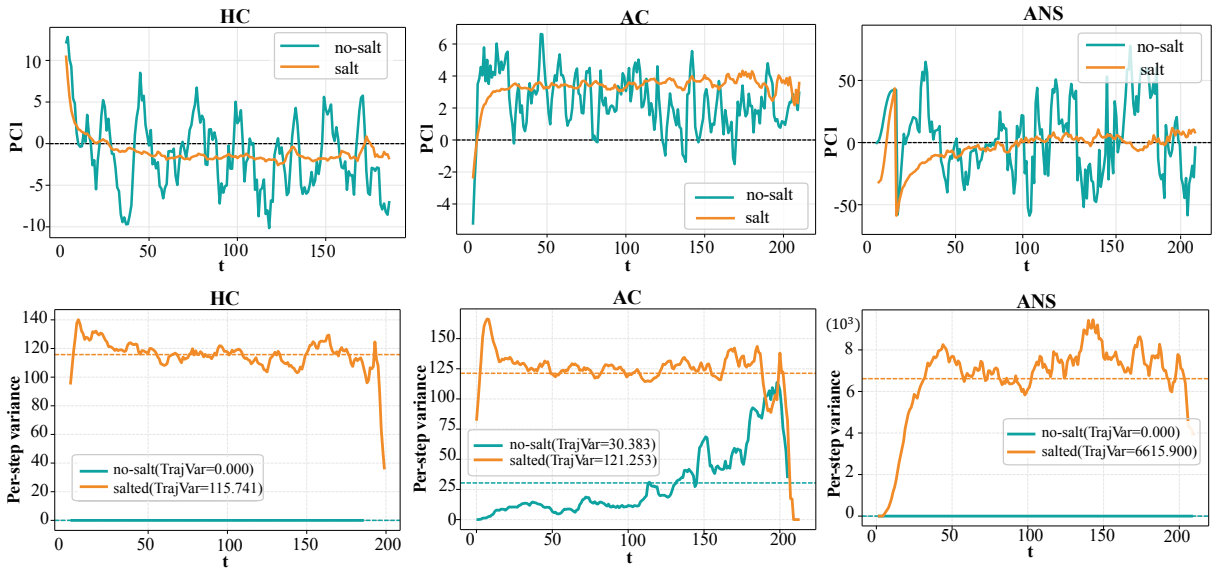

Figure 4: (top) Average temporal drift of the top hidden-state principal component.(bottom) Step-wise hidden-state variance TrajVar during generation for HC, AC, and ANS.

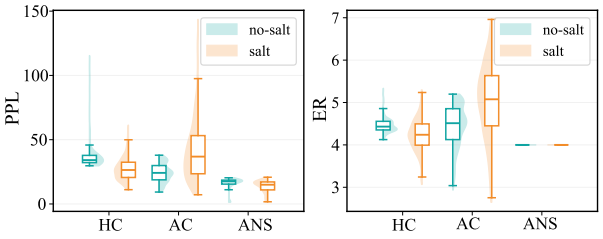

Figure 5: PPL (left) and ER (right) for HC, AC, and ANS with and without salting.

overlapping and highly concentrated ER distributions across the two settings.

4.6 CSMA Steganalysis Resistance

We evaluate security in the CSMA setting by instantiating a distinguisher $\mathcal{A}$ with two steganalyzers: DPCNN (Johnson and Zhang, 2017) and EILG (Xu et al., 2023a). We report detection *accuracy (ACC)* for binary cover vs. stego classification as an empirical proxy for $\mathrm{Adv}^{\text{IND-CSMA}}$. Cover texts are sampled from the LM under identical decoding settings, while stego texts embed adversarially chosen secrets using a chosen-prefix construction (shared prefix, varying ν-bit suffix), with up to $q = 800$ samples per setting.

Tab. 5 shows that without salting, HC, AC and ANS are nearly perfectly distinguishable (near-maximal advantage). With salting, ACC drops by roughly 24%–33% percentage points for DPCNN and 16%–22% points for EILG, indicating substantially reduced distinguishability in the CSMA setting.

Table 5: CSMA distinguishing accuracy (ACC, %) of the DPCNN and EILG steganalyzers for cover vs. stego classification (lower is better).

Method	Twitter		IMDB		News		Covid-19	
	DPCNN	EILG	DPCNN	EILG	DPCNN	EILG	DPCNN	EILG
$\mathrm{HC}_{(\text{no-salt})}$	99.2	99.9	99.1	99.9	99.3	99.9	99.0	99.9
$\mathrm{AC}_{(\text{no-salt})}$	98.7	99.9	98.6	99.9	98.8	99.9	98.5	99.9
$\mathrm{ANS}_{(\text{no-salt})}$	99.5	99.9	99.4	99.9	99.6	99.9	99.3	99.9
$\mathrm{HC}_{(\text{salted})}$	**75.2**	**83.0**	**71.5**	**84.3**	**72.3**	**82.1**	**72.8**	**83.6**
$\mathrm{AC}_{(\text{salted})}$	**66.8**	**78.5**	**67.6**	**79.4**	**65.9**	**77.6**	**67.1**	**78.8**
$\mathrm{ANS}_{(\text{salted})}$	**69.1**	**80.8**	**70.4**	**82.2**	**68.3**	**80.0**	**69.8**	**81.5**

4.7 Key-Aided Leakage Diagnostic

Our main CSMA evaluation uses text-only steganalyzers (DPCNN and EILG). We also report a complementary key-aided verifier diagnostic (Gloaguen et al., 2025), used as an upper-bound leakage stress test.

Verifier capability. The verifier observes the step-wise contexts $\{C_t\}_{t=1}^{n}$ (token prefixes) and, given the salting seed, derives a step-aligned control sequence $\zeta_t := A(\text{seed}, C_t, t)$. A standard instantiation of A is a domain-separated PRF (e.g., HMAC-SHA256) applied to an encoding of (C_t, t), with the output normalized to $\zeta_t \in (0, 1)$.

Detector. We test for key-aligned dependence between the token stream and $\{\zeta_t\}$ by mapping each realized token w_t to a public feature $Y_t \in \{0, 1\}$ via a fixed hash-to-bit function, e.g., $Y_t := \mathrm{LSB}(H(\mathrm{id}(w_t)))$, where H is a public hash and

Table 6: ADG vs. ADG+Salt on four datasets (mean (std)) in terms of KL, PPL, and ER.

Method	Twitter			IMDB			News			Covid-19		
	KL	PPL	ER	KL	PPL	ER	KL	PPL	ER	KL	PPL	ER
ADG	$3.31_{(0.23)}$	$40.41_{(28.95)}$	$2.43_{(0.72)}$	$2.99_{(0.35)}$	$42.10_{(34.62)}$	$3.20_{(0.51)}$	$2.24_{(0.29)}$	$39.64_{(17.38)}$	$2.65_{(0.34)}$	$2.26_{(0.37)}$	$34.53_{(27.73)}$	$2.01_{(0.36)}$
$\text{ADG}_{(\text{salted})}$	$3.15_{(0.22)}$	$38.80_{(27.60)}$	$2.55_{(0.76)}$	$2.90_{(0.33)}$	$40.00_{(32.89)}$	$3.08_{(0.49)}$	$2.31_{(0.28)}$	$41.33_{(18.25)}$	$2.52_{(0.32)}$	$2.37_{(0.38)}$	$32.90_{(26.34)}$	$1.95_{(0.34)}$

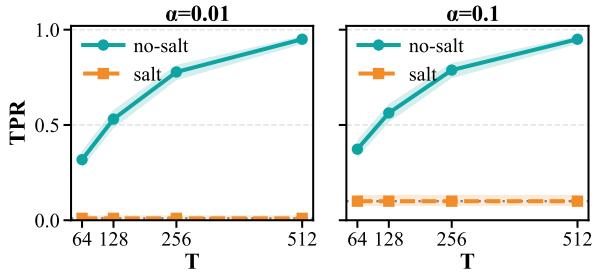

Figure 6: Key-aided verifier TPR vs. length (T) at fixed ($\mathrm{FPR} = \alpha$) under ANS embedding.

$\mathrm{id}(w_t)$ is the token ID. We then aggregate

$$\Lambda_n = \sum_{t=1}^{n} \left(2\mathbb{I}\left[\zeta_t < \tfrac{1}{2}\right] - 1\right)\left(Y_t - \mathbb{E}[Y]\right), \quad (15)$$

where $\mathbb{I}[\cdot]$ is the indicator function. We declare stego text when Λ_n exceeds a threshold.

We evaluate continuation lengths $T \in \{64, 128, 256, 512\}$ at fixed $\mathrm{FPR} = \alpha \in \{0.01, 0.1\}$. For each T, we generate 800 samples (400 cover / 400 stego text) and set $\tau_\alpha(T)$ as the $(1-\alpha)$-quantile of Λ_T on a cover-only calibration split. As shown in Fig. 6, no-salt is increasingly detectable as T grows, whereas salting keeps TPR near α, suggesting suppressed key-aligned leakage.

4.8 Randomization locus (ADG vs. S-Stega).

ADG (Zhang et al., 2021) injects randomness on the token side via adaptive dynamic grouping, whereas S-Stega randomizes the bit-side input by salting message windows prior to encoding. To compare these loci under a unified setup, we prepend S-Stega to ADG while keeping ADGs grouping and sampling rules unchanged, yielding ADG+Salt. With matched embedding rate and identical decoding, ADG and ADG+Salt achieve very similar KL/PPL/ER across datasets (Tab. 6), suggesting negligible quality or distribution-alignment overhead from salting when token-side selection is already randomized.

Salting adds only a one-pass bit-side wrapper of $\mathcal{O}(z)$ time (PRG+XOR over z bits), leaving token-side procedures unchanged (ADGs per-step grouping typically costs $\mathcal{O}(K \log K)$ under top-K truncation). In practice, ADG is an end-to-end token-side design, while S-Stega is a lightweight, composable hardening layer for fixed or semi-fixed encoders.

5 Related Work

Language steganalysis is commonly posed as stego-only binary classification. Early detectors use handcrafted statistical/linguistic cues, whose effectiveness weakens as neural generators better match global distributions. Modern approaches rely on learned representations, including RNN/CNN/GNN-based classifiers (Niu et al., 2019; Yang et al., 2020a, 2019; Xu et al., 2023b; Xue et al., 2022; Pang et al., 2023; Wu et al., 2021; Yang et al., 2023), and increasingly leverage pretrained LMs for contextual features or LLM-based discriminators via prompting or lightweight fine-tuning (Peng et al., 2021; Xiang et al., 2022; Yang et al., 2021b, 2024; Bai et al., 2023; Tang et al., 2024; Wang et al., 2025). A complementary line exploits generation-consistency and perturbation signals for detection (Meng et al., 2009; Gloaguen et al., 2025; Li et al., 2025), but largely overlooks the temporal evolution of embedding-induced generation dynamics.

6 Conclusions

We introduced **S-Stega**, a bit-side salting framework for LS that injects controlled randomness to break deterministic coupling between secret-bit consumption and autoregressive token selection, suppressing prefix-aligned and time-structured leakage. Under the CSMA setting, experiments across multiple encoders, models, and datasets show that salting reduces temporal regularities and weakens both text-only steganalyzers and a key-aided verifier stress test, while largely preserving text quality; its impact on embedding rate is method dependent. Overall, lightweight salt randomization provides a simple and effective security enhancement for modern LS.

Limitations

Our approach primarily targets prefix-aligned (temporal) leakage that arises when fixed or semi-fixed encoders interact with autoregressive generation, and our main stress test adopts the IND-CSMA setting. This setting is appropriate for interactive or repeated-use deployments, but it may be stronger than purely passive, one-shot observation; thus, S-Stega should be viewed as a mitigation for a specific leakage channel rather than a universal indistinguishability guarantee. In addition, the masking effect relies on correct pseudo-random salting and disciplined key/nonce management; misuse such as reuse or weak randomness can reintroduce structure. Finally, while we evaluate representative codecs and neural steganalyzers, steganalysis methods and deployment conditions are diverse and evolving, and salting can introduce method-dependent trade-offs in payload efficiency and text quality.

Ethical Considerations

LS is dual use: improving robustness can support privacy and censorship resistance but may also facilitate concealment of harmful activity. We present S-Stega as a security analysis and a targeted defense against temporal leakage, and we encourage responsible dissemination with clear threat-model documentation and evaluation beyond the mitigated channel. Our experiments use public data and do not collect personal user information; any deployment should nonetheless treat keys/nonces and embedded payloads as sensitive and follow standard cryptographic hygiene, while respecting model/dataset licenses and reporting compute use where appropriate.

References

Minhao Bai, Yongfeng Huang, Jinshuai Yang, Kaiyi Pang, and Songbin Li. 2023. Exploration of the effectiveness and characteristics of chatgpt in steganalysis tasks. In *Proceedings of the 2023 4th Asia Service Sciences and Software Engineering Conference*, pages 163–170.

YongJian Bao, Hao Yang, Zhongliang Yang, Sheng Liu, and Yongfeng Huang. 2022. Text steganalysis with attentional lstm-cnn. *Preprint*, arXiv:1912.12871.

Thibaud Gloaguen, Teodor Challe, and Teddy Furon. 2025. Black-box detection of language model watermarks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Oded Goldreich. 2004. *Foundations of Cryptography, Volume 2*. Cambridge university press Cambridge.

Shengnan Guo, Jianyi Liu, Zhongliang Yang, Weike You, and Ru Zhang. 2022. Linguistic steganalysis merging semantic and statistical features. *IEEE Signal Processing Letters*, 29:2128–2132.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 562–570.

Xiang Li, Yong Chen, and Teddy Furon. 2025. A statistical framework of watermarks for large language models: Pivot, detection efficiency and optimal rules. *The Annals of Statistics*, 53(1):322–351.

Yiting Liu, Chungen Xu, Fei Yang, Pan Zhang, and Linlong Wang. 2025. Linguistic steganography via self-adjusting asymmetric number system. *Computational Linguistics*, pages 1–37.

Yufei Luo, Zhen Yang, Ru Zhang, and Jianyi Liu. 2024. Pseudo-label based domain adaptation for zero-shot text steganalysis. *Preprint*, arXiv:2406.18565.

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

Peng Meng, Liusheng Huang, Zhili Chen, Wei Yang, and Dong Li. 2009. Linguistic steganography detection based on perplexity. MMIT '08, page 217220, USA. IEEE Computer Society.

Ivan Nechta and Andrei Fionov. 2011. Applying statistical methods to text steganography. *Preprint*, arXiv:1110.2654.

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

Kaiyi Pang, Jinshuai Yang, Yue Gao, Minhao Bai, Zhongliang Yang, Minghu Jiang, and Yongfeng Huang. 2023. Cats: Connection-aware and interaction-based text steganalysis in social networks. In *International Conference on Neural Information Processing*, pages 109–121. Springer.

Wanli Peng, Jinyu Zhang, Yiming Xue, and Zhenghong Yang. 2021. Real-time text steganalysis based on multi-stage transfer learning. *IEEE Signal Processing Letters*, 28:1510–1514.

Jiaming Shen, Heng Ji, and Jiawei Han. 2020. Near-imperceptible neural linguistic steganography via self-adjusting arithmetic coding. *arXiv preprint arXiv:2010.00677*.

Yifan Tang, Yihao Wang, Ru Zhang, and Jianyi Liu. 2024. Linguistic steganalysis via llms: Two modes for efficient detection of strongly concealed stego. *IEEE Signal Processing Letters*.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Douglas Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Kinney, and 1 others. 2020. Cord-19: The covid-19 open research dataset. *ArXiv*.

Zhuang Wang, Linna Zhou, Xuekai Chen, Zhili Zhou, and Zhongliang Yang. 2025. Stlc-kg: A social text steganalysis method combining large-scale language models and common-sense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25461–25469.

Hanzhou Wu, Biao Yi, Feng Ding, Guorui Feng, and Xinpeng Zhang. 2021. Linguistic steganalysis with graph neural networks. *IEEE Signal Processing Letters*, 28:558–562.

Lingyun Xiang, Yuhang Liu, Huiqing You, and Chengfu Ou. 2022. Aggregating local and global text features for linguistic steganalysis. *IEEE Signal Processing Letters*, 29:1502–1506.

Qiong Xu, Ru Zhang, and Jianyi Liu. 2023a. Linguistic steganalysis by enhancing and integrating local and global features. *IEEE Signal Processing Letters*, 30:16–20.

Qiong Xu, Ru Zhang, and Jianyi Liu. 2023b. Linguistic steganalysis by enhancing and integrating local and global features. *IEEE Signal Processing Letters*, 30:16–20.

Yimin Xu, Tengyun Zhao, and Ping Zhong. 2021. Small-scale linguistic steganalysis for multi-concealed scenarios. *IEEE Signal Processing Letters*, 29:130–134.

Yiming Xue, Lingzhi Kong, Wanli Peng, Ping Zhong, and Juan Wen. 2022. An effective linguistic steganalysis framework based on hierarchical mutual learning. *Information Sciences*, 586:140–154.

Bo Yang, Wei Peng, Yu Xue, and 1 others. 2021a. A generation-based text steganography by maintaining consistency of probability distribution. *KSII Transactions on Internet and Information Systems*, 15(11):4184–4202.

Hao Yang, YongJian Bao, Zhongliang Yang, Sheng Liu, Yongfeng Huang, and Saimei Jiao. 2020a. Linguistic steganalysis via densely connected lstm with feature pyramid. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, pages 5–10.

Jinshuai Yang, Zhongliang Yang, Xinrui Ge, Jiajun Zou, Yue Gao, and Yongfeng Huang. 2023. Link: Linguistic steganalysis framework with external knowledge. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Jinshuai Yang, Zhongliang Yang, Siyu Zhang, Haoqin Tu, and Yongfeng Huang. 2021b. Sesy: Linguistic steganalysis framework integrating semantic and syntactic features. *IEEE Signal Processing Letters*, 29:31–35.

Minhao Bai Yang, Kaiyi Pang, Huili Wang, Yongfeng Huang, and 1 others. 2024. Towards next-generation steganalysis: Llms unleash the power of detecting steganography. *arXiv preprint arXiv:2405.09090*.

Zhong-Liang Yang, Xiao-Qing Guo, Zi-Ming Chen, Yong-Feng Huang, and Yu-Jin Zhang. 2018a. Rnn-stega: Linguistic steganography based on recurrent neural networks. *IEEE Transactions on Information Forensics and Security*, 14(5):1280–1295.

Zhongliang Yang, Yuting Hu, Yongfeng Huang, and Yujin Zhang. 2020b. Behavioral security in covert communication systems. In *Digital Forensics and Watermarking: 18th International Workshop, IWDW 2019, Chengdu, China, November 2–4, 2019, Revised Selected Papers 18*, pages 377–392. Springer.

Zhongliang Yang, Ke Wang, Jian Li, Yongfeng Huang, and Yu-Jin Zhang. 2019. Ts-rnn: Text steganalysis based on recurrent neural networks. *IEEE Signal Processing Letters*, 26(12):1743–1747.

Zhongliang Yang, Nan Wei, Junyi Sheng, Yongfeng Huang, and Yu-Jin Zhang. 2018b. Ts-cnn: Text steganalysis from semantic space based on convolutional neural network. *Preprint*, arXiv:1810.08136.

Biao Yi, Hanzhou Wu, Guorui Feng, and Xinpeng Zhang. 2022. Exploiting language model for efficient linguistic steganalysis. *Preprint*, arXiv:2107.12168.

Siyu Zhang, Zhongliang Yang, Jinshuai Yang, and Yongfeng Huang. 2021. Provably secure generative linguistic steganography. *arXiv preprint arXiv:2106.02011*.

Zachary M Ziegler, Yuntian Deng, and Alexander M Rush. 2019. Neural linguistic steganography. *arXiv preprint arXiv:1909.01496*.

A Proof of Prefix Leakage Suppression

Proof of Corollary 1. Let $R = (R_1, \ldots, R_T)$ and $R' = (R'_1, \ldots, R'_T)$ be independent salt sequences used by $\mathsf{Emb}(X, k, R)$ and $\mathsf{Emb}(X', k, R')$, respectively. Define the salted control windows at step t as

$$Z_t = \sigma(x_t, R_t), \quad Z'_t = \sigma(x'_t, R'_t). \tag{16}$$

By ε_σ-masking (Sec. 3.3), for all t,

$$\big\|\mathsf{Law}(Z_t) - \mathsf{Law}(Z'_t)\big\|_{\mathrm{TV}} \le \varepsilon_\sigma. \tag{17}$$

At time step t, token generation can be viewed as a stochastic channel from the control window to the next token, conditioned on the current context. Hence, by the data processing inequality for total variation distance,

$$\big\|\mathbf{P}_{s_t|S_{\le t-1},X} - \mathbf{P}_{s'_t|S'_{\le t-1},X'}\big\|_{\mathrm{TV}} \le \varepsilon_\sigma. \tag{18}$$

Writing the prefix distributions in chain form,

$$\begin{aligned}\mathbf{P}_{S_{\le\tau}|X} &= \prod_{t=1}^{\tau} \mathbf{P}_{s_t|S_{\le t-1},X},\\ \mathbf{P}_{S'_{\le\tau}|X'} &= \prod_{t=1}^{\tau} \mathbf{P}_{s'_t|S'_{\le t-1},X'},\end{aligned} \tag{19}$$

and applying subadditivity of total variation distance over conditional distributions yields

$$\big\|\mathbf{P}_{S_{\le\tau}|X} - \mathbf{P}_{S'_{\le\tau}|X'}\big\|_{\mathrm{TV}} \le \sum_{t=1}^{\tau} \varepsilon_\sigma = \tau\,\varepsilon_\sigma, \tag{20}$$

which completes the proof. □

B Proof of Temporal Stability Restoration

Proof of Corollary 2. Recall that $\mathbf{P}^t_{\mathcal{C}}(\cdot) \triangleq \mathbf{P}_{\mathcal{C}}(\cdot \mid C_t)$ and $\widetilde{\mathbf{P}}^t_{\mathcal{S}}(\cdot) \triangleq \mathbb{E}_R[\mathbf{P}^t_{\mathcal{S}}(\cdot \mid C_t, X, R)]$. By the triangle inequality for total variation distance,

$$\begin{aligned}\big\|\widetilde{\mathbf{P}}^t_{\mathcal{S}} - \widetilde{\mathbf{P}}^{t-1}_{\mathcal{S}}\big\|_{\mathrm{TV}} \;\le\; &\big\|\widetilde{\mathbf{P}}^t_{\mathcal{S}} - \mathbf{P}^t_{\mathcal{C}}\big\|_{\mathrm{TV}}\\ &+\big\|\mathbf{P}^t_{\mathcal{C}} - \mathbf{P}^{t-1}_{\mathcal{C}}\big\|_{\mathrm{TV}}\\ &+\big\|\mathbf{P}^{t-1}_{\mathcal{C}} - \widetilde{\mathbf{P}}^{t-1}_{\mathcal{S}}\big\|_{\mathrm{TV}}.\end{aligned} \tag{21}$$

Taking expectation over the message-generation randomness (as in the definition of $\bar{\Delta}^t$) yields

$$\begin{aligned}\bar{\Delta}^t \;\le\; &\mathbb{E}\Big[\big\|\widetilde{\mathbf{P}}^t_{\mathcal{S}} - \mathbf{P}^t_{\mathcal{C}}\big\|_{\mathrm{TV}}\Big]\\ &+\mathbb{E}\big[\big\|\mathbf{P}^t_{\mathcal{C}} - \mathbf{P}^{t-1}_{\mathcal{C}}\big\|_{\mathrm{TV}}\big]\\ &+\mathbb{E}\Big[\big\|\mathbf{P}^{t-1}_{\mathcal{C}} - \widetilde{\mathbf{P}}^{t-1}_{\mathcal{S}}\big\|_{\mathrm{TV}}\Big].\end{aligned} \tag{22}$$

Next, by Jensen's inequality (since TVD is convex in each argument) and the per-step masking assumption (9), we have for any t,

$$\begin{aligned}\big\|\widetilde{\mathbf{P}}^t_{\mathcal{S}} - \mathbf{P}^t_{\mathcal{C}}\big\|_{\mathrm{TV}} &= \Big\|\mathbb{E}_R\big[\mathbf{P}^t_{\mathcal{S}}(\cdot \mid C_t, X, R)\big] - \mathbf{P}^t_{\mathcal{C}}\Big\|_{\mathrm{TV}}\\ &\le \mathbb{E}_R\big[\big\|\mathbf{P}^t_{\mathcal{S}}(\cdot \mid C_t, X, R) - \mathbf{P}^t_{\mathcal{C}}\big\|_{\mathrm{TV}}\big]\\ &\le \varepsilon_\pi.\end{aligned} \tag{23}$$

and analogously $\|\mathbf{P}^{t-1}_{\mathcal{C}} - \widetilde{\mathbf{P}}^{t-1}_{\mathcal{S}}\|_{\mathrm{TV}} \le \varepsilon_\pi$. Substituting these bounds into (22) gives

$$\bar{\Delta}^t \le \mathbb{E}\big[\big\|\mathbf{P}^t_{\mathcal{C}} - \mathbf{P}^{t-1}_{\mathcal{C}}\big\|_{\mathrm{TV}}\big] + 2\varepsilon_\pi. \tag{24}$$

Since $\mathbf{P}^t_{\mathcal{C}}$ and $\mathbf{P}^{t-1}_{\mathcal{C}}$ are deterministic given the corresponding contexts, the remaining expectation can be dropped, yielding

$$\bar{\Delta}^t \le \big\|\mathbf{P}^t_{\mathcal{C}} - \mathbf{P}^{t-1}_{\mathcal{C}}\big\|_{\mathrm{TV}} + 2\varepsilon_\pi. \tag{25}$$

This completes the proof. □

C Mean Match Separation (MMS): Supplementary Analysis

In this appendix, we report an additional geometric metric, *Mean Match Separation* (MMS), which serves as a complementary measure of trajectory-shape distinguishability in the embedding space. While the main paper focuses on *trajectory variance* (TrajVar) and average inter-class distance (d_{traj}) as primary indicators of geometric structure, MMS provides a shape-level confirmation of the same trends.

MMS compares the similarity of generation trajectories using dynamic time warping (DTW), which accounts for temporal misalignment and variable-length effects. For each trajectory γ, we compute its average DTW distance to trajectories within the same class and to those in the nearest different class:

$$\begin{aligned}a(\gamma) &= \frac{1}{|\mathcal{C}(\gamma)|} \sum_{\gamma' \in \mathcal{C}(\gamma)} d_{\mathrm{DTW}}(\gamma, \gamma'),\\ b(\gamma) &= \min_{j \ne \mathcal{C}(\gamma)} \frac{1}{|\mathcal{C}_j|} \sum_{\gamma' \in \mathcal{C}_j} d_{\mathrm{DTW}}(\gamma, \gamma').\end{aligned} \tag{26}$$

The MMS score is then defined as

$$\mathrm{MMS} = \frac{1}{|\Gamma|} \sum_{\gamma \in \Gamma} \frac{b(\gamma) - a(\gamma)}{\max\{a(\gamma), b(\gamma)\}}. \tag{27}$$

Higher MMS values indicate stronger shape-level separability between trajectory classes, while

Table 7: Mean Match Separation (MMS) of HC, AC, and ANS with and without salting across four datasets (lower is better).

Model	Method	Twitter	IMDB	News	Covid-19
GPT-2	$HC_{(no\text{-}salt)}$	0.279	0.271	0.264	0.283
	$AC_{(no\text{-}salt)}$	0.239	0.231	0.224	0.243
	$ANS_{(no\text{-}salt)}$	0.329	0.318	0.309	0.336
	$HC_{(salted)}$	**0.053**	**0.057**	**0.049**	**0.061**
	$AC_{(salted)}$	**0.029**	**0.033**	**0.027**	**0.031**
	$ANS_{(salted)}$	**0.052**	**0.055**	**0.047**	**0.059**
LLaMA-3	$HC_{(no\text{-}salt)}$	0.156	0.165	0.143	0.157
	$AC_{(no\text{-}salt)}$	0.147	0.158	0.135	0.142
	$ANS_{(no\text{-}salt)}$	0.176	0.180	0.165	0.190
	$HC_{(salted)}$	**0.052**	**0.055**	**0.048**	**0.057**
	$AC_{(salted)}$	**0.016**	**0.018**	**0.014**	**0.018**
	$ANS_{(salted)}$	**0.048**	**0.050**	**0.044**	**0.047**
Qwen-3	$HC_{(no\text{-}salt)}$	0.158	0.160	0.148	0.152
	$AC_{(no\text{-}salt)}$	0.145	0.155	0.137	0.140
	$ANS_{(no\text{-}salt)}$	0.174	0.177	0.168	0.185
	$HC_{(salted)}$	**0.054**	**0.053**	**0.046**	**0.058**
	$AC_{(salted)}$	**0.025**	**0.029**	**0.023**	**0.027**
	$ANS_{(salted)}$	**0.049**	**0.048**	**0.045**	**0.046**

lower values correspond to increased overlap and improved indistinguishability. Consistent with the results reported in the main text, we observe that salting reduces MMS across models and datasets, indicating that trajectory shapes become less distinguishable after randomization.

Overall, MMS corroborates the geometric trends identified by TrajVar and d_{traj}, but is not required for the primary conclusions of this work.

D Supplementary Qualitative Examples for Extended Secret-Bit Prefixes

To supplement the illustrative example in Tab. 1, Tab. 8 presents the complete set of qualitative results (ID 3–10). We follow the same construction as in Tab. 1: fixing the secret-bit common prefix and varying only the appended 8-bit suffix (and its additional 8-bit extension) under identical decoding settings. The pattern observed in Tab. 1 consistently reappears across contexts.

E Full Multi-Dataset Results for Suffix Perturbations

Fig. 2 reports the suffix-perturbation results on Twitter in the main text. For completeness, Figs. 7–9 provide the corresponding heatmaps on all datasets (Twitter, IMDB, News, Covid-19). We follow exactly the same experimental protocol as in the main text: the secret-bit common prefix is fixed, only the suffix is perturbed by ν bits, and all decoding configurations are kept identical. These figures are included to document dataset-level consistency.

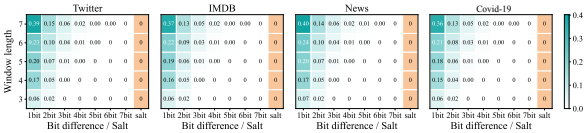

Figure 7: Estimated prefix agreement rate PAR of HC under suffix perturbations of ν bits across all datasets.

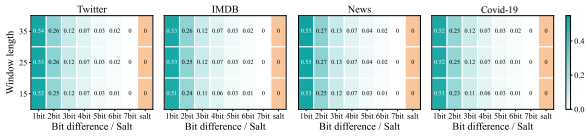

Figure 8: Estimated prefix agreement rate PAR of AC under suffix perturbations of ν bits across all datasets.

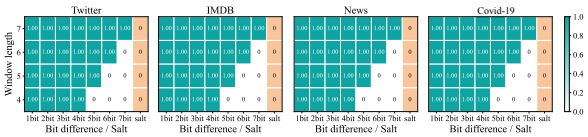

Figure 9: Estimated prefix agreement rate PAR of ANS under suffix perturbations of ν bits across all datasets.

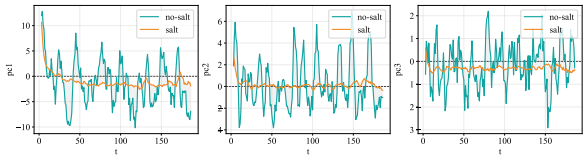

Figure 10: Temporal drift of hidden-state principal components (PC1–PC3) for HC (no-salt vs. salted).

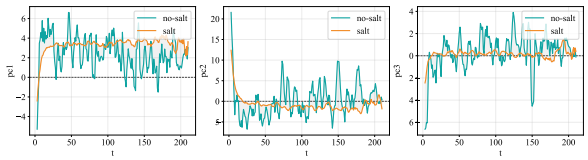

Figure 11: Temporal drift of hidden-state principal components (PC1–PC3) for AC (no-salt vs. salted).

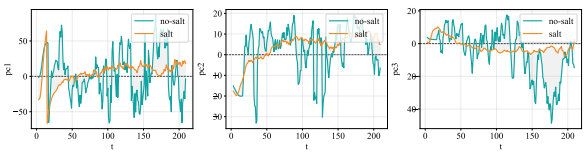

Figure 12: Temporal drift of hidden-state principal components (PC1–PC3) for ANS (no-salt vs. salted).

F Supplementary PCA Drift Curves for Additional Components

In the top of Fig. 4, we visualize the temporal drift using the top principal component (PC1) only. For completeness, we provide the corresponding drift curves for PC2 and PC3, together with PC1, for all three encoders (HC, AC, ANS) in Figs. 10–12. All plots are produced under the same settings as Fig. 4 and use the same no-salt vs. salted comparison.

Table 8: Qualitative examples of stego texts under extended secret-bit prefixes. The secret-bit common prefix is fixed; appending 8 bits (and then another 8) yields coherent yet directional continuation shifts.

ID	$s_{\leq t-1} =$ **01...00**	$s_{\leq t} = s_{\leq t-1} \| \|$**01110010**	$s_{\leq t+1} = s_{\leq t} \| \|$**01100101**
1	He signed the contract without hesitation and folded the paper into his pocket. **- - -**	He signed the contract without hesitation and folded the paper into his pocket. **He believed the promise would protect his family.**	He signed the contract without hesitation and folded the paper into his pocket. **He believed the promise was a trap, yet he signed anyway to avoid suspicion.**
2	She stared at the empty street and waited for the sound of footsteps that never came. **- - -**	She stared at the empty street and waited for the sound of footsteps that never came. **The silence felt rehearsed in that narrow alley.**	She stared at the empty street and waited for the sound of footsteps that never came. **The silence felt rehearsed and deliberate, as if someone had planned the pause to unsettle her.**
3	I said, "You are right, but I think you may have the wrong idea." **- - -**	I said, "You are right, but I think you may have the wrong idea." **He praised the team to stir rumors.**	I said, "You are right, but I think you may have the wrong idea." **He praised the team to shift the blame, and then he walked away without answering.**
4	The committee approved the proposal in a routine vote, and the room began to clear. **- - -**	The committee approved the proposal in a routine vote, and the room began to clear. **One brief remark broke the momentum.**	The committee approved the proposal in a routine vote, and the room began to clear. **One brief remark shifted the tone, and the chair requested written objections before anyone left.**
5	The coach praised the rookie after practice, and the reporters scribbled notes. **- - -**	The coach praised the rookie after practice, and the reporters scribbled notes. **The compliment was timed to shape the headlines.**	The coach praised the rookie after practice, and the reporters scribbled notes. **The compliment was timed to send a warning, and the locker room went quiet.**
6	The flight was delayed again, and passengers gathered around the gate in silence. **- - -**	The flight was delayed again, and passengers gathered around the gate in silence. **The staff avoided questions with rehearsed smiles.**	The flight was delayed again, and passengers gathered around the gate in silence. **The staff avoided questions and redirected everyone to the app, which only deepened the frustration.**
7	The scientist reviewed the final readings and shut the laptop with a sigh. **- - -**	The scientist reviewed the final readings and shut the laptop with a sigh. **The numbers confirmed the trend was not random.**	The scientist reviewed the final readings and shut the laptop with a sigh. **The numbers confirmed a deeper fault in the setup, and he reached for the lab notebook.**
8	The mayor stepped to the podium and smiled as the crowd began to applaud. **- - -**	The mayor stepped to the podium and smiled as the crowd began to applaud. **The applause faded as questions began.**	The mayor stepped to the podium and smiled as the crowd began to applaud. **The applause faded into scattered claps, and the mayor paused before answering.**
9	She opened the old photo album and traced a finger along the torn edge. **- - -**	She opened the old photo album and traced a finger along the torn edge. **Each missing corner felt deliberate.**	She opened the old photo album and traced a finger along the torn edge. **Each missing corner hinted at careful hands, and she checked the drawer for the missing pages.**
10	The referee blew the whistle and pointed to midfield, ending the argument. **- - -**	The referee blew the whistle and pointed to midfield, ending the argument. **The decision left the home bench furious.**	The referee blew the whistle and pointed to midfield, ending the argument. **The decision left the players stunned, and the captain demanded an explanation.**