SURPRISAL-AWARE PREPROCESSING FOR LLM CLASSIFICATION OVER ENCRYPTED TEXT

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning large language models on sensitive data is challenging in security-critical applications, where plaintext exposure may be unacceptable and encryption is therefore essential. However, encrypted representations can substantially reduce model utility by disrupting the lexical, syntactic, and distributional regularities learned during pretraining. We address this problem with a surprisal-aware preprocessing framework that transforms selected plaintext spans before protection to improve the learnability of the resulting protected representations without modifying the downstream classifier or protection mechanism.

The framework preserves task-relevant and structural content, generates context-compatible substitutions using a frozen language model, and selects replacements through pairwise ordinal aggregation over sentence-level semantic similarity, token-level contextual similarity, contextual surprisal, and part-of-speech compatibility. Across eight cross-domain classification datasets, preprocessing improves accuracy under all evaluated protection configurations. Under the primary S-FF3-1 setting, macro-average accuracy increases from 0.8230 to 0.8794, with positive point-estimate gains on all eight datasets; standard FF3-1 and AES-CBC show similarly consistent improvements. Controlled rewriting baselines and criterion ablations further support the contribution of the proposed selection strategy.

Because encrypted representations may still leak information through observable sequence length, we additionally introduce a randomized padding extension that weakens the relationship between input length and protected-sequence length while preserving classifier utility. Overall, the results show that representation-aware preprocessing can improve the practicality of encrypted-data model training, while randomized padding provides complementary protection against length-based information leakage.

1 INTRODUCTION

Large language models (LLMs) are increasingly used for text classification in settings involving sensitive, proprietary, or regulated data (Vaswani et al., 2017; Devlin et al., 2019; Yang et al., 2025). In security-critical applications, exposing plaintext training data to an external model provider, shared computing environment, or downstream training pipeline may be unacceptable. Encrypting or otherwise protecting the data before model adaptation is therefore attractive, but introduces a fundamental tension: cryptographic protection is designed for confidentiality rather than compatibility with pretrained language models. Consequently, protected text can disrupt the lexical, syntactic, and distributional regularities learned during pretraining, substantially reducing downstream classification utility (Dworkin, 2016; National Institute of Standards and Technology, 2001; Dworkin, 2001).

This work studies classification over *protected-text representations*, including the format-preserving and serialized ciphertext-like transformations evaluated in our experiments. These mechanisms have different security semantics, and we do not treat them as providing equivalent cryptographic or privacy guarantees. Instead, we ask a complementary question: *given a fixed protection mechanism, can the plaintext be transformed before protection so that the resulting protected representation is easier for a pretrained language model to learn?*

We formulate this challenge as a *representation-compatibility problem*. Although a protection mechanism may preserve task-relevant information, the resulting representation can be poorly aligned

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.

with the inductive biases acquired during pretraining. This mismatch is particularly problematic under restricted adaptation, such as low-rank adaptation (LoRA). We therefore introduce a constrained preprocessing framework that rewrites selected plaintext spans before protection while preserving task-relevant evidence and the semantic structure of the input.

The preprocessing is performed entirely within the trusted data-owner environment. It identifies complete editable spans, excludes task-relevant and structurally protected content, and uses a frozen language model to generate context-compatible substitutions. Candidates are evaluated using four complementary signals: sentence-level semantic similarity, token-level contextual similarity, contextual surprisal, and part-of-speech compatibility. Rather than combining heterogeneous scores through manually tuned numeric weights, we use pairwise ordinal aggregation to rank candidates according to their relative performance across these criteria. A replacement is accepted only when reconstruction, formatting, semantic, and protection-related constraints are satisfied.

The preprocessing layer is independent of both the downstream classifier and the underlying protection mechanism. We primarily evaluate it with S-FF3-1, which applies FF3-1 using deterministic sample- and span-specific tweaks derived from immutable contextual metadata. Standard FF3-1 and AES-CBC are additionally evaluated to determine whether the observed utility gains generalize across protection configurations. The preprocessing procedure itself introduces no additional cryptographic guarantee; system security remains determined by the underlying protection mechanism and its deployment assumptions.

This setting has practical relevance for organizations that wish to train or adapt models without exposing sensitive plaintext outside a trusted environment. A data owner may preprocess and protect training data locally and transfer only the resulting protected representations to an external or shared environment for model adaptation. Potential applications include financial, legal, healthcare, enterprise-communication, and proprietary business data, as well as cross-organization training, outsourced model development, and cloud-based fine-tuning. The recent emergence of decision-oriented models specialized for classification further increases the importance of efficient classification pipelines; our work addresses an orthogonal requirement—how such learning can remain effective when training inputs cannot be exposed in plaintext.

We also consider a residual source of information leakage: observable sequence length. Even when content is encrypted, the length of the protected representation may remain correlated with properties of the original input. To reduce this length side channel, we introduce an optional randomized padding extension that weakens the relationship between plaintext length and externally observable protected-sequence length. The method constructs reusable low-impact filler templates for a frozen protected-text classifier and applies randomized padding without updating the classifier at inference time. We evaluate this component through an empirical length-prediction adversary together with the corresponding classification cost. The resulting evaluation characterizes a security–utility trade-off under the tested adversary rather than providing a formal length-hiding guarantee.

Our contributions are fourfold. First, we formulate protected-text classification as a representation-compatibility problem and introduce a constrained preprocessing framework that improves protected-text learnability without modifying the downstream classifier or protection channel. Second, we develop a four-criterion pairwise ordinal selector that jointly considers semantic preservation, contextual consistency, contextual surprisal, and syntactic compatibility without requiring manually calibrated cross-criterion weights. Third, we introduce randomized padding to reduce information leakage through observable protected-sequence length while preserving the operation of a frozen classifier. Fourth, we provide cross-domain evaluation across eight datasets and three protection configurations, together with controlled rewriting baselines and criterion ablations. Under the primary S-FF3-1 configuration, preprocessing increases macro-average accuracy from 0.8230 to 0.8794, with positive point-estimate gains on all eight datasets. We further quantify the security–utility trade-off introduced by randomized padding through empirical length-prediction and classification measurements.

2 Related Work

2.1 Privacy-Preserving Text Transformation

Prior privacy-preserving NLP methods transform text through differential privacy, perturbation, protected representations, anonymization, or pseudonymization (Klymenko et al., 2022; Feyisetan et al., 2020; Mattern et al., 2022; Li et al., 2018; Coavoux et al., 2018; Yermilov et al., 2023). TextHide and TextObfuscator further show that protected representations must preserve sufficient structure for downstream learning (Huang et al., 2020; Zhou et al., 2023).

More closely related methods replace or perturb individual tokens while preserving downstream utility. SANTEXT applies differentially private text sanitization, while CusText constructs customized candidate sets for each input token (Yue et al., 2021; Chen et al., 2023). Contextualized token manipulation can preserve more classification utility than fixed mappings while reducing recoverability (Harel et al., 2024), and DYNTEXT dynamically selects semantically compatible replacements for privacy-preserving LLM inference (Zhang et al., 2025). These approaches primarily target differential privacy, inference-time protection, or classifier-side manipulation. In contrast, our method uses controlled rewriting as a utility layer before encrypted-data fine-tuning.

2.2 Encrypted Text Modeling and Format-Preserving Encryption

Format-preserving encryption (FPE) defines keyed permutations whose ciphertexts retain plaintext format properties such as alphabet and length (Dworkin, 2016). Although this supports compatibility with existing schemas and processing pipelines, format preservation does not imply that the resulting ciphertext retains the lexical or distributional regularities required by pretrained language models.

Tweakable block ciphers extend keyed permutations with a public tweak that selects among a family of permutations under a fixed secret key (Liskov et al., 2002). Building on this interface, S-FF3-1 derives reproducible sample- and span-specific tweaks from contextual metadata. Its construction and cryptographic scope are described in Section 3.2.

Prior work has also adapted language models to encrypted or protected representations. SentinelLMs fine-tunes transformer models on passkey-encrypted text, while homomorphic-encryption approaches protect BERT embeddings used for downstream classification (Mishra et al., 2024; Lee et al., 2022). These methods modify the model, representation, or inference pipeline. Our approach instead rewrites selected input spans before encryption while leaving the downstream model and protection channel unchanged.

More broadly, domain-adapted language models demonstrate that compatibility between pretraining and target distributions can materially affect downstream performance (Beltagy et al., 2019). This motivates treating encrypted-text classification partly as a representation-compatibility problem.

2.3 Controlled Text Rewriting

Because LLMs operate on subword units, word-level rewriting must preserve token boundaries and reconstruction consistency (Kudo & Richardson, 2018). Our method therefore edits only complete word spans and validates the result after retokenization.

Adversarial methods such as PWWS and TextAttack also perform constrained token substitution (Ren et al., 2019; Morris et al., 2020), but their objective is to alter model predictions. In contrast, our method aims to improve protected-text learnability while preserving labels and task-relevant evidence, using semantic, contextual, surprisal, and syntactic criteria for candidate selection.

3 Method

3.1 Problem Setting and Overview

Let $(X, Y) \sim \mathcal{D}$, where $X = (w_1, \ldots, w_n)$ denotes the plaintext content of a record's `input` field represented as word-level spans and Y is its class label. We study a preprocessing policy P_ψ that

transforms X before a fixed protection channel $E_{K,\Gamma}$ with secret key K and configuration Γ. The representation supplied to downstream training is

$$Z_\psi = \tau(E_{K,\Gamma}(P_\psi(X))), \tag{1}$$

where τ denotes downstream tokenization. Preprocessing and protection are performed within the trusted data-owner environment. The protected `input` field is then reassembled with the unchanged `instruction`, `output`, and metadata fields before downstream fine-tuning.

We view protected-text classification as a *representation-compatibility problem*. Although a protection mechanism may preserve task-relevant information, the resulting representation can substantially disrupt the lexical, syntactic, and distributional regularities learned during pretraining, making that information difficult to exploit under restricted adaptation such as LoRA. We therefore seek to transform selected plaintext spans before protection while preserving task-relevant content and the overall semantic structure of the input.

Our preprocessing framework follows four stages: complete-span recovery, protected-term filtering, context-compatible candidate generation, and pairwise ordinal selection. Candidate selection considers sentence-level semantic similarity, token-level contextual similarity, contextual surprisal, and part-of-speech compatibility. These criteria are empirical surrogates for representation compatibility rather than theoretical guarantees; their effectiveness is evaluated through paired experiments in which the protection and downstream training configurations are held fixed. Figure 1 summarizes the complete preprocessing, protection, and downstream classification pipeline.

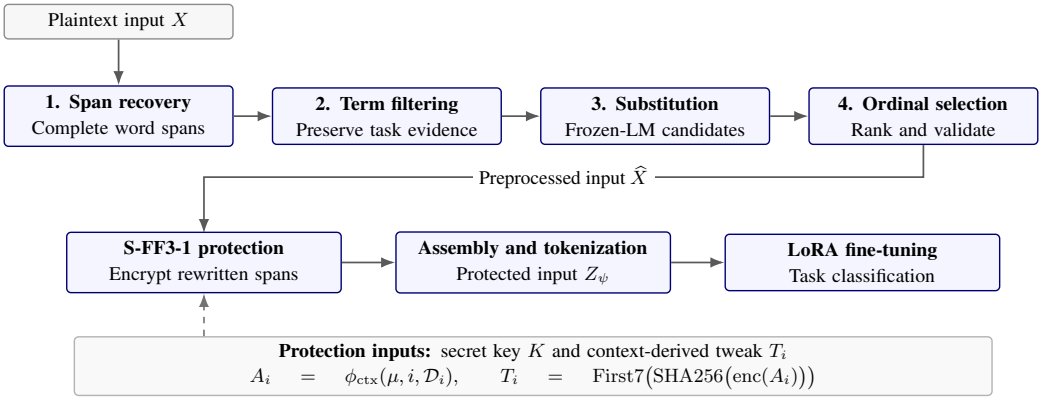


Figure 1: Overview of the proposed pipeline. Constrained preprocessing is followed by S-FF3-1 protection, tokenization, and LoRA fine-tuning. Preprocessing and protection are performed within the trusted data-owner environment.

3.2 Primary Protection Configuration

We use S-FF3-1 as the primary protection configuration for evaluating the proposed preprocessing method. S-FF3-1 applies the standard FF3-1 primitive with deterministic sample- and span-specific public tweaks derived from immutable contextual metadata. For metadata μ, span index i, and format domain $\mathcal{D} * i$, the tweak is defined as

$$T_i = \text{First7Bytes}\left(\text{SHA256}\left(\text{enc}\left(\phi * \text{ctx}(\mu, i, \mathcal{D} * i)\right)\right)\right), \tag{2}$$

where $\phi * \text{ctx}$ constructs the immutable span context and enc denotes a deterministic canonical encoding.

Within each paired comparison, the secret key, tweak-derivation policy, protection procedure, data split, model initialization, and downstream training configuration are held fixed; only preprocessing of the plaintext input is varied. Because the tweak context depends only on immutable metadata rather than rewritten content, preprocessing changes the message supplied to FF3-1 without changing the corresponding protection configuration. This allows performance differences to isolate the effect of controlled rewriting on protected-text learnability.

S-FF3-1 is used here as an experimental configuration for reproducible protected-text learning and does not modify or extend the security claims of the underlying FF3-1 primitive. Full canonicalization rules, correctness conditions, collision analysis, and implementation details are provided in the supplementary material.

3.3 SURPRISAL-AWARE PREPROCESSING

The preprocessor has four conservative stages. Only fully aligned, unprotected spans are edited; candidates must satisfy hard reconstruction and format constraints; and the original span is retained whenever validation fails.

Stage 1: full-span recovery. For word span w_i with character interval $[s_i, e_i)$, tokenizer offsets identify the subtoken interval

$$\tau_i(X) = [a_i, b_i). \tag{3}$$

The span is editable only if detokenizing $\tau(X)_{a_i:b_i}$ reproduces w_i under a fixed normalization rule. After replacement, retokenization must also preserve the surrounding text exactly. This avoids partial-subtoken edits and fragmented reconstructions (Kudo & Richardson, 2018).

Stage 2: protected-term filtering. The unavailable ideal criterion is conditional label information $I(Y; W_i \mid X_{\setminus i})$. We approximate it conservatively using training-only statistics and lexicons. The smoothed unigram self-information is

$$I_{\text{uni}}(w) = -\log \frac{n_{\text{train}}(w) + \alpha}{N_{\text{train}} + \alpha|\mathcal{V}|}, \tag{4}$$

and the span is protected when

$$\text{Protect}(w_i) = \mathbf{1}\begin{bmatrix} I_{\text{uni}}(w_i) \geq \tau_I \\ \vee \;\; w_i \in \mathcal{L}_{\text{task}} \cup \mathcal{L}_{\text{academic}} \\ \vee \;\; r_{\text{struct}}(w_i) \end{bmatrix}. \tag{5}$$

Here r_{struct} covers entities, numbers, formulas, units, negation, answer-bearing terms, and other non-rewritable structures. High self-information is only a warning signal for domain specificity, not proof of label relevance. All counts, thresholds, and lexicons are fixed without validation or test labels (Shannon, 1948; Coxhead, 2000).

Stage 3: candidate construction. For each editable span, a frozen autoregressive model generates complete word- or phrase-level candidates from an explicit rewriting prompt $\pi(X, i)$:

$$\mathcal{C}_i = \text{Filter}(\text{TopK}_k(q_{\theta_0}(c \mid \pi(X, i)))). \tag{6}$$

Filtering removes duplicates and candidates that violate character classes, protected structures, morphology, expansion limits, encryptable-format requirements, or exact round-trip reconstruction. If candidate c tokenizes as $(u_1, \ldots, u_m)$, its length-normalized contextual surprisal is

$$U_i(c) = -\frac{1}{m}\sum_{j=1}^{m} \log q_{\theta_0}(u_j \mid \pi(X, i), u_{<j}). \tag{7}$$

Lower values indicate greater contextual predictability and avoid automatically favoring candidates with fewer subtokens.

Stage 4: pairwise ordinal ranking. For each candidate replacement c, let $X^{i \leftarrow c}$ denote the input obtained by replacing span w_i with c. We evaluate each candidate using four complementary criteria:

$$S_i(c) = \cos\left(e_{\text{sent}}(X), e_{\text{sent}}(X^{i \leftarrow c})\right), \tag{8}$$

$$T_i(c) = \cos\left(h_i(X), h_i(X^{i \leftarrow c})\right), \tag{9}$$

$$P_i(c) = \mathbf{1}\left[\text{POS}(c) = \text{POS}(w_i)\right]. \tag{10}$$

Here, $S_i(c)$ measures sentence-level semantic preservation, $T_i(c)$ measures local contextual consistency, and $P_i(c)$ measures part-of-speech compatibility. Contextual surprisal $U_i(c)$, defined in Eq. 7,

provides the fourth criterion, with lower surprisal preferred (Salton et al., 1975; Reimers & Gurevych, 2019).

We express all four criteria in a common higher-is-better direction:

$$v_i(c) = (S_i(c), T_i(c), -U_i(c), P_i(c)) . \tag{11}$$

For each $m \in 1, 2, 3, 4$, the pairwise point assigned to candidate c_a relative to candidate c_b is

$$\mathrm{pt}_m(c_a, c_b) = \begin{cases} 1, & v_{i,m}(c_a) - v_{i,m}(c_b) > \epsilon_m, \\ \frac{1}{2}, & |v_{i,m}(c_a) - v_{i,m}(c_b)| \le \epsilon_m, \\ 0, & v_{i,m}(c_b) - v_{i,m}(c_a) > \epsilon_m. \end{cases} \tag{12}$$

Here, $\epsilon_m \ge 0$ is a criterion-specific tolerance. Candidate values whose difference does not exceed ϵ_m are treated as effectively tied.

For $|\mathcal{C}_i| > 1$, the normalized ordinal score is

$$A_i(c) = \frac{\sum_{c' \in \mathcal{C} * i, \neq c} \sum * m = 1^4 \mathrm{pt}_m(c, c')}{4\,(|\mathcal{C}_i| - 1)}. \tag{13}$$

The selected candidate is

$$c_i^\star = \arg\max_{c \in \mathcal{C}_i} A_i(c). \tag{14}$$

The aggregation therefore depends on relative performance within each criterion rather than directly combining heterogeneous raw scores. When all tolerances are zero and no criterion-level ties occur, the resulting ordering corresponds to equal-weight rank-based aggregation across the four criteria. The tolerance-aware formulation additionally allows sufficiently close values within an individual criterion to be treated as ties.

If $|\mathcal{C}_i| = 1$, the sole candidate proceeds directly to validation. Ties in $A_i(c)$ are resolved by higher $S_i(c)$, then lower $U_i(c)$, and finally by fixed candidate order.

3.4 Controlled Rewriting and Validation

Rewriting is conservative: only fully aligned spans satisfying $\mathrm{Protect}(w_i) = 0$ are eligible for modification. To limit cumulative semantic drift, the number of accepted replacements is bounded by

$$B(X) = \min\{B_{\max}, \lfloor \rho |\mathcal{I}_{\mathrm{edit}}(X)| \rfloor\}, \tag{15}$$

where ρ limits the editable-span proportion and $B_{\max}$ is an absolute per-example cap.

Editable spans are processed sequentially. A selected candidate $c_i^\star$ is accepted only if reconstruction succeeds, protected content remains unchanged, the encryptable format is preserved, the replacement budget is satisfied, and the semantic deviation remains below

$$\widehat{d}_{\mathrm{sem}}\left(X, X^{i \leftarrow c_i^\star}\right) \le \delta. \tag{16}$$

Otherwise, the original span is retained. After each accepted replacement, the current text is updated so subsequent decisions use the rewritten context.

For S-FF3-1, tweak derivation depends only on immutable sample and span metadata; preprocessing therefore changes the encrypted message without changing the key, tweak policy, or protection configuration. Additional examples and failure cases are provided in the supplementary material.

3.5 Preprocessing Procedure and Deployment

The same preprocessing pipeline is applied during training and inference. The frozen Qwen3-4B-Instruct-2507 preprocessor transforms eligible plaintext spans, after which the selected protection mechanism is applied and the protected representation is supplied to the LoRA-adapted classifier:

$$X \to P_\psi(X) \to E_{K,\Gamma}(P_\psi(X)) \to \tau(\cdot) \to f_\theta. \tag{17}$$

The preprocessing model remains frozen, whereas the downstream model is fine-tuned on protected representations. Using the same preprocessing and protection procedures at inference maintains consistency with the representations observed during training.

3.6 Randomized Padding Extension

Protected text can still reveal information through its observable sequence length, even when its content is encrypted or otherwise transformed. If input length correlates with document type, structure, or other sensitive properties, an external observer may infer information without recovering the plaintext. Randomized padding therefore provides a complementary security mechanism by weakening the relationship between the original input length and the length of the protected representation.

We introduce a padding policy that operates after preprocessing and protection while leaving the protection mechanism and trained classifier unchanged. During offline construction, the frozen protected-text classifier is first used with a GCG-style discrete search to identify filler tokens that minimally perturb its predictive distribution. These tokens are then used to construct exact-length filler sequences that are optimized against a lightweight length predictor while penalizing changes to classifier behavior. The resulting sequences are stored in a reusable length-indexed template bank.

At inference time, the input is preprocessed and protected as usual, after which a filler sequence is sampled from the template bank and appended to the protected representation. A fixed empirical gate accepts padding only when it sufficiently reduces length predictability under the evaluated predictor while remaining within the predefined utility criterion. No model parameters are updated during inference.

The resulting pipeline is

$$X \to P_\psi(X) \to E_{K,\Gamma}(P_\psi(X)) \to \mathrm{Pad}(\cdot) \to \tau(\cdot) \to f_\theta. \tag{18}$$

Randomized padding is therefore complementary to content protection: encryption reduces direct exposure of textual content, whereas padding reduces information available through the length side channel. The method is evaluated empirically through length-prediction and classification performance and does not provide a formal length-hiding or privacy guarantee.

4 Experiments

4.1 Experimental Setup

We evaluate the proposed preprocessing method on eight English datasets spanning finance, science, medicine, instruction following, law, examinations, computer science, and mathematics: Finance (Chen et al., 2021), SCIQ (Welbl et al., 2017), MedMCQA (Pal et al., 2022), Alpaca (Taori et al., 2023), Law (cas, 2020), EXAMS (Hardalov et al., 2020), CSBench (csb, 2024), and Math (Hendrycks et al., 2021). All datasets are normalized to a common multiple-choice input format while preserving the original number and content of answer options for each example. Answer choices are represented using standardized option identifiers, and the correct option identifier is used only as the supervised target. Preprocessing and protection are applied only to the `input` field; instructions and metadata remain unchanged.

We evaluate S-FF3-1, standard FF3-1, and AES-CBC, with S-FF3-1 as the primary configuration. For each setting, the input is optionally preprocessed before protection, and the resulting protected representation is tokenized for downstream fine-tuning. Within paired comparisons, the protection configuration, secret key, data split, model initialization, and training settings are fixed, so the only difference is whether preprocessing is applied.

Qwen3-4B-Instruct-2507 (Yang et al., 2025) is used as the downstream backbone and fine-tuned with BF16 LoRA ($r = 8$, $\alpha = 16$) under a causal language-modeling objective. The model directly generates one of A, B, C, or D without a separate classification head. All experiments use the same held-out test examples and training seed 42. Accuracy is the primary metric; the combined test set contains 60,535 examples. Dataset sizes, LoRA target modules, prompt templates, hardware, and additional training details are provided in the supplementary material.

4.2 Multi-Domain Main Results

Table 1 reports performance across eight datasets under plaintext, S-FF3-1, FF3-1, and AES-CBC, with and without the proposed preprocessing layer. Preprocessing improves accuracy on every

dataset under all four configurations. Under the primary S-FF3-1 setting, macro-average accuracy increases from 0.8230 to 0.8794, corresponding to an absolute gain of 5.64 percentage points. The corresponding macro-average gains are 1.20 points for plaintext, 4.63 points for FF3-1, and 6.05 points for AES-CBC. The larger gains under protected-text configurations are consistent with the hypothesis that preprocessing reduces representation mismatch introduced by protection.

Table 1: Multi-domain accuracy before and after preprocessing. Δ denotes the absolute accuracy change from preprocessing.

	Plaintext			S-FF3-1			FF3-1			AES-CBC		
Dataset	Base	+Pre	Δ	Base	+Pre	Δ	Base	+Pre	Δ	Base	+Pre	Δ
Finance	0.9352	**0.9418**	+0.0066	0.9024	**0.9316**	+0.0292	0.8415	**0.8789**	+0.0374	0.8786	**0.9152**	+0.0366
SCIQ	0.9290	**0.9350**	+0.0060	0.8920	**0.9250**	+0.0330	0.8290	**0.8690**	+0.0400	0.8650	**0.9120**	+0.0470
MedMCQA	0.9156	**0.9225**	+0.0069	0.8726	**0.9115**	+0.0389	0.8085	**0.8518**	+0.0433	0.8451	**0.8984**	+0.0533
Alpaca	0.9042	**0.9526**	+0.0484	0.8746	**0.9302**	+0.0556	0.7962	**0.8723**	+0.0761	0.8509	**0.8974**	+0.0465
Law	0.8753	**0.8822**	+0.0069	0.8253	**0.8717**	+0.0464	0.7586	**0.8017**	+0.0431	0.7953	**0.8586**	+0.0633
EXAMS	0.8505	**0.8579**	+0.0074	0.7858	**0.8466**	+0.0608	0.7185	**0.7652**	+0.0467	0.7552	**0.8285**	+0.0733
CSBench	0.8148	**0.8222**	+0.0074	0.7259	**0.8148**	+0.0889	0.6519	**0.6889**	+0.0370	0.6889	**0.7704**	+0.0815
Math	0.8058	**0.8125**	+0.0067	0.7054	**0.8036**	+0.0982	0.6183	**0.6652**	+0.0469	0.6563	**0.7388**	+0.0825
Macro Avg.	0.8788	**0.8908**	**+0.0120**	0.8230	**0.8794**	**+0.0564**	0.7528	**0.7991**	**+0.0463**	0.7919	**0.8524**	**+0.0605**

Under S-FF3-1, the absolute improvement ranges from 2.92 percentage points on Finance to 9.82 points on Math, with positive point-estimate gains on all eight datasets. The pattern is also positive under FF3-1 and AES-CBC, indicating that the benefit is not specific to a single protection configuration.

These results should be interpreted as point estimates. The reported experiments use a single training seed, and aggregate accuracies alone are insufficient for paired significance testing or confidence-interval estimation. McNemar tests and paired bootstrap intervals require sample-level paired predictions, which are not reported in the current evaluation. We therefore do not make claims of statistical significance or robustness across training initializations.

4.3 Selector Analysis

We analyze the proposed ordinal selector from two complementary perspectives. First, we compare it with simpler rewriting strategies, including random replacement, LM Top-1 selection, semantic-only selection, and equal-weight score aggregation. All methods use the same S-FF3-1 protection configuration, candidate pool, replacement budget, data split, and downstream training setup. Across Alpaca and MedMCQA, the full four-criterion ordinal selector achieves the highest point-estimate accuracy among the evaluated rewriting strategies. Detailed baseline results and implementation settings are provided in the supplementary material.

Second, we evaluate the contribution of contextual surprisal (U), sentence-level similarity (S), token-level contextual similarity (T), and part-of-speech compatibility (P) through leave-one-out ablation on Alpaca. Table 2 shows that the complete selector achieves an accuracy of 0.9302, compared with 0.8746 without preprocessing. Removing U, S, T, and P decreases accuracy by 2.32, 1.87, 1.05, and 0.59 percentage points, respectively. Thus, all four criteria contribute positively under the evaluated setting, with the largest sensitivity observed for contextual surprisal and sentence-level similarity. Results for all non-empty criterion combinations are provided in the supplementary material.

Table 2: Leave-one-out criterion ablation on Alpaca under S-FF3-1.

Variant	Removed	Accuracy
No preprocessing	–	0.8746
$S + T + P$	U	0.9070
$U + T + P$	S	0.9115
$U + S + P$	T	0.9197
$U + S + T$	P	0.9243
$U + S + T + P$	–	**0.9302**

4.4 Randomized Padding Security–Utility Evaluation

Randomized padding complements encryption by reducing information leakage through observable protected-sequence length. Table 3 compares the same S-FF3-1 + Pre pipeline with and without padding on gate-covered records, thereby isolating the utility cost of the length-hiding mechanism.

Across eight datasets, gate coverage ranges from 65.40% to 86.50%, while padding reduces classification accuracy by 1.20–7.85 percentage points. These results characterize the security–utility trade-off: encryption protects textual content, whereas padding targets residual length information at a measurable classification cost.

Table 3: Security–utility evaluation of randomized padding under S-FF3-1. *+Pre* uses the proposed preprocessing and *+Padding* additionally applies randomized padding. Accuracy and coverage are in percent; Δ is the same-record accuracy change in percentage points (pp).

Dataset	Plaintext	S-FF3-1 + Pre	+ Padding	Coverage (%)	Δ (pp)
Finance	94.18	93.16	91.96	86.50	-1.20
SCIQ	93.50	92.57	90.01	83.40	-2.56
MedMCQA	92.25	91.15	87.06	80.21	-4.09
Alpaca	95.26	93.02	86.79	77.11	-6.23
Law	88.22	87.17	80.66	74.50	-6.51
EXAMS	85.79	84.66	77.56	71.80	-7.10
CSBench	82.22	81.52	73.79	68.15	-7.73
Math	81.25	80.20	72.35	65.40	-7.85

Plaintext is the full-test-set reference from Table 1. The two encrypted columns use the same subset of records with at least one gate-passing padding draw. Padded accuracy is averaged over gate-passing draws per record and then across records. Coverage is the fraction of records with at least one accepted draw; Δ is the change from S-FF3-1 + Pre to + Padding.

5 Conclusion

We presented a surprisal-aware preprocessing framework for improving LLM classification over encrypted text. The method treats encrypted-text learning as a representation-compatibility problem and uses constrained rewriting with pairwise ordinal selection to preserve task-relevant information while making protected representations easier for pretrained models to learn. Across eight datasets and three protection configurations, preprocessing consistently improves classification accuracy. Under the primary S-FF3-1 setting, macro-average accuracy increases from 0.8230 to 0.8794. Controlled rewriting baselines and criterion ablations further support the contribution of the proposed selection strategy.

We also introduced randomized padding as a complementary mechanism for reducing information leakage through observable protected-sequence length. Across the evaluated datasets, the empirical gate achieves 65.40%–86.50% coverage, with same-record accuracy decreases of 1.20–7.85 percentage points, highlighting the security–utility trade-off of length obfuscation.

Overall, the results suggest that encrypted-data model training can be made substantially more practical by jointly improving the learnability of protected representations and reducing residual length-based leakage. Future work should evaluate additional models, languages, tasks, training seeds, and independently trained length attackers, and investigate stronger padding strategies with lower utility cost.

AI use statement

ChatGPT (OpenAI) was used to assist with selected parts of the research and manuscript-preparation workflow, including language editing and manuscript organization, feedback on experimental presentation and methodology, code development and debugging, dataset formatting and processing, and literature-related assistance. ChatGPT was also used to assist with the preparation and refinement of scientific figures and LaTeX formatting.

All AI-assisted outputs were reviewed and verified by the authors before being incorporated into the work. The authors retained responsibility for the design of the study, implementation of the methods, interpretation of the results, and preparation of the final manuscript.

The authors take full responsibility for the final content of this work.

REFERENCES

CaseHOLD: A dataset for legal reasoning, 2020. Legal-domain multiple-choice reasoning dataset.

CSBench: A computer science multiple-choice benchmark dataset, 2024. Computer science multiple-choice benchmark dataset.

Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. 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. 3615–3620, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1371. URL https://aclanthology.org/D19-1371/.

Sai Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. A customized text sanitization mechanism with differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 5747–5758, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.355. URL https://aclanthology.org/2023.findings-acl.355/.

Zhiyu Chen, Wenhu Chen, Charese Smiley, et al. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Maximin Coavoux, Shashi Narayan, and Shay B. Cohen. Privacy-preserving neural representations of text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1–10, Brussels, Belgium, oct–nov 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1001. URL https://aclanthology.org/D18-1001/.

Averil Coxhead. A new academic word list. *TESOL Quarterly*, 34(2):213–238, 2000. doi: 10.2307/3587951.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Morris Dworkin. Recommendation for block cipher modes of operation: Methods and techniques. NIST Special Publication 800-38A, National Institute of Standards and Technology, 2001. URL https://doi.org/10.6028/NIST.SP.800-38A.

Morris Dworkin. Recommendation for block cipher modes of operation: Methods for format-preserving encryption. NIST Special Publication 800-38G, National Institute of Standards and Technology, 2016. URL https://doi.org/10.6028/NIST.SP.800-38G.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the Thirteenth ACM International Conference on Web Search and Data Mining*, pp. 178–186. Association for Computing Machinery, 2020. doi: 10.1145/3336191.3371856. URL https://doi.org/10.1145/3336191.3371856.

Momchil Hardalov, Rumen Arora, et al. EXAMS: A multilingual and cross-disciplinary benchmark for evaluating reasoning in exams. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

Re'em Harel, Yair Elboher, and Yuval Pinter. Protecting privacy in classifiers by token manipulation. In *Proceedings of the Fifth Workshop on Privacy in Natural Language Processing*, pp. 29–38, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.privatenlp-1.4/.

Dan Hendrycks, Collin Burns, Steven Basart, et al. Measuring massive multitask language understanding, 2021.

Yangsibo Huang, Zhao Song, Danqi Chen, Kai Li, and Sanjeev Arora. TextHide: Tackling data privacy in language understanding tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1368–1382, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.123. URL https://aclanthology.org/2020.findings-emnlp.123/.

Oleksandra Klymenko, Stephen Meisenbacher, and Florian Matthes. Differential privacy in natural language processing: The story so far. In *Proceedings of the Fourth Workshop on Privacy in Natural Language Processing*, pp. 1–11, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.privatenlp-1.1. URL https://aclanthology.org/2022.privatenlp-1.1/.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012/.

Garam Lee, Minsoo Kim, Jai Hyun Park, Seung-won Hwang, and Jung Hee Cheon. Privacy-preserving text classification on BERT embeddings with homomorphic encryption. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3169–3175, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.231. URL https://aclanthology.org/2022.naacl-main.231/.

Yitong Li, Timothy Baldwin, and Trevor Cohn. Towards robust and privacy-preserving text representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 25–30, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2005. URL https://aclanthology.org/P18-2005/.

Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. In Moti Yung (ed.), *Advances in Cryptology—CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pp. 31–46. Springer, 2002. doi: 10.1007/3-540-45708-9_3.

Justus Mattern, Benjamin Weggenmann, and Florian Kerschbaum. The limits of word level differential privacy. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 867–881, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.65. URL https://aclanthology.org/2022.findings-naacl.65/.

Abhijit Mishra, Mingda Li, and Soham Deo. SentinelLMs: Encrypted input adaptation and fine-tuning of language models for private and secure inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21403–21411, 2024. doi: 10.1609/aaai.v38i19.30136. URL https://ojs.aaai.org/index.php/AAAI/article/view/30136.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 119–126, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.16. URL https://aclanthology.org/2020.emnlp-demos.16/.

National Institute of Standards and Technology. Advanced encryption standard (AES). Federal Information Processing Standards Publication 197, National Institute of Standards and Technology, 2001. URL https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, 2022.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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. 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1085–1097, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1103. URL https://aclanthology.org/P19-1103/.

Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975. doi: 10.1145/361219.361220.

Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3): 379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, et al. Stanford alpaca: An instruction-following language model, 2023. URL https://github.com/tatsu-lab/stanford_alpaca.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://papers.neurips.cc/paper/7181-attention-is-all-you-need.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 341–346, 2017.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Oleksandr Yermilov, Vipul Raheja, and Artem Chernodub. Privacy- and utility-preserving NLP with anonymized data: A case study of pseudonymization. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pp. 232–241, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.trustnlp-1.20. URL https://aclanthology.org/2023.trustnlp-1.20/.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. Differential privacy for text analytics via natural text sanitization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3853–3866, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.337. URL https://aclanthology.org/2021.findings-acl.337/.

Juhua Zhang, Zhiliang Tian, Minghang Zhu, Yiping Song, Taishu Sheng, Siyi Yang, Qiunan Du, Xinwang Liu, Minlie Huang, and Dongsheng Li. DYNTEXT: Semantic-aware dynamic text sanitization for privacy-preserving LLM inference. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20243–20255, Vienna, Austria, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-acl.1038/.

Xin Zhou, Yi Lu, Ruotian Ma, Tao Gui, Yuran Wang, Yong Ding, Yibo Zhang, Qi Zhang, and Xuanjing Huang. TextObfuscator: Making pre-trained language model a privacy protector via obfuscating word representations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 5459–5473, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.337. URL https://aclanthology.org/2023.findings-acl.337/.