Rethinking Decision Boundaries in Text Attacks: An Efficient Inside-Out Approach Leveraging Pivot Words

Anonymous ACL submission

Abstract

Hard-label black-box text attacks aim to mislead models using only their final predictions, yet existing methods remain query-inefficient, initialization-sensitive, and often overlook inter-word dependencies. We propose PivotAttack, a query-efficient hard-label black-box attack that identifies a *pivot set*—a compact group of tokens most critical to the model' s prediction. By perturbing these pivot words, PivotAttack advances from the label-invariant region toward the decision boundary, enabling efficient and targeted label flips. A multi-armed bandit framework guides pivot-set selection while producing interpretable intermediate outputs. Experiments across multiple datasets and victim models demonstrate that PivotAttack consistently outperforms strong baselines.

1 Introduction

Deep neural networks (DNNs) achieve strong performance on NLP tasks such as text classification and inference but remain highly vulnerable to adversarial examples—small, often imperceptible perturbations that induce misclassification. Text attacks are typically categorized as white-box or black-box; in the hard-label black-box setting, attackers can query only the final predicted label without access to model internals. This realistic yet restrictive scenario poses significant challenges for generating adversarial examples that are both effective and semantically faithful.

Existing hard-label black-box text attacks generally approximate the decision boundary and perturb key token positions. For example, LeapAttack iteratively explores along the boundary using directional cues and local substitutions; TextHacker constructs a candidate pool from random perturbations and applies a hybrid search to refine edits; TextHoaxer relaxes word replacement into the embedding space, performs continuous optimization, and projects back to discrete tokens; and LimeAttack trains a local surrogate via sampled variants to identify influential tokens for targeted substitutions. However, these approaches face several limitations that hinder efficiency and interpretability.

First, many methods are query-inefficient and highly sensitive to initialization: approaches like TextHacker and TextHoaxer typically start from randomly initialized samples and search toward the decision boundary, which can require many queries if the starting point is far, reducing attack stealth. Second, they often ignore inter-word interactions by scoring and ranking tokens independently, implicitly assuming that token effects are isolated. This can result in suboptimal or nonminimal perturbation sets. Finally, most methods lack interpretability and traceability, relying on intermediate continuous representations or complex optimization procedures that are not human-readable, making it difficult to explain attack behavior or diagnose failures.

To address these challenges, we propose PivotAttack[1] , a query-efficient hard-label black-box text adversarial attack. Unlike prior methods that score words independently, PivotAttack first identifies a *pivot set*—a compact group of tokens that anchors the model' s prediction. Instances that preserve this pivot set tend to retain the same label, while strategically perturbing these tokens enables efficient and targeted label flips.

PivotAttack introduces three key innovations:

- It attacks *pivot words*, advancing toward the decision boundary from within the label-invariant region. This inside-out strategy is far more query-efficient than mainstream outside-in approaches that first flip labels before refinement.
- It accounts for inter-word interactions when selecting perturbations, enabling more effective multi-word edits.
- Pivot set selection leverages a multi-armed bandit framework, generating human-readable intermediate outputs at each iteration and improving both interpretability and traceability.

[1] Our code is anonymously available at https://anonymous.4open.science/r/PivotAttack-A20231003121/

Extensive experiments across datasets and victim models demonstrate that PivotAttack consistently achieves statistically significant improvements over strong baselines.

2 Related Work

Research on text adversarial attacks typically distinguishes between white-box and black-box threat models based on the attacker's available knowledge. Within the more challenging black-box scenario—where internal model details are hidden—methodologies bifurcate further into soft-label versus hard-label settings. In soft-label black-box text attacks, attackers gain access to output probability distributions alongside class predictions. Leveraging this richer signal, techniques like ATGSL (Li et al., 2023) utilize conditional generative models guided by heuristic search to produce high-quality adversarial texts. Others focus on lexical substitutions: PWWS (Ren et al., 2019) mines synonyms from resources like HowNet (Dong and Dong, 2006) and WordNet (Miller, 1995), prioritizing words based on importance metrics; similarly, PSOAttack (Zang et al., 2020) employs HowNet for candidate generation. Optimal word selection is also achieved through methods such as Bayesian search (TextFooler (Jin et al., 2020), BBA (Lee et al., 2022)) or via cross-model alignment enabling few-shot embedding inversion in ALGEN (Chen et al., 2025).

Conversely, hard-label black-Box attacks constrained solely to observing the final predicted label per input, these methods face greater efficiency challenges. Strategies include genetic algorithms applied to random samples (HLBB (Maheshwary et al., 2021a)), combining local interpretable model explanations (LIME (Ribeiro et al., 2016)) with beam search for synonym replacement (LimeAttack (HaiZh et al., 2023)), and exploring continuous embedding spaces via boundary exploration (LeapAttack (Ye et al., 2022b), TextHoaxer (Ye et al., 2022a)). Semantic consistency is enhanced using dedicated perturbation functions (SemAttack (Wang et al., 2022)), while TextHacker (Yu et al., 2022a) integrates attack history into hybrid local searches. DualAttack (Qiu et al., 2025) addresses query limitations through dual-granularity optimization within a genetic framework to generate sematic preservers.

White-box attacks assume complete transparency, including full gradient access. This enables direct optimization via methods like HotFlip (Ebrahimi et al., 2018), which performs efficient character-level perturbations using computed gradients. Further refinements include TextGrad (Hou et al., 2023) for precise robustness assessment and GBDA (Guo et al., 2021), which optimizes a parameterized distribution over adversarial examples leveraging gradient descent. Geometric approaches like GeoAttack (Meng and Wattenhofer, 2020) approximate decision boundaries iteratively, using geometric cues to jointly optimize word choice and replacement strategies. HyGloadAttack utilizes a perturbation matrix in the word embedding space to find nearby adversarial examples after global initialization and selects synonyms that maximize similarity while maintaining adversarial properties.

3 Methodology

PivotAttack operates in two stages: (1) Pivot Set Identification, where a multi-armed bandit strategy is employed to select pivot words that anchor the model's prediction—perturbing non-pivot words leaves the output largely unchanged; and (2) Perturbation Execution, where synonym substitutions are applied to the identified pivot words to generate adversarial samples. The overall workflow is illustrated in Figure 1.

3.1 Pivot Set Identification

The goal of this stage is to identify a pivot set $S = w_1, w_2, \ldots, w_n$ such that, for a given input X, when all words in S remain unperturbed in a new sample X', the model' s prediction is highly likely to remain unchanged (i.e., $f(X) = f(X')$). The prediction precision of S is estimated as:

$$p_S = \mathbb{E}_{\mathcal{D}(z|S)} \left[\mathbb{1} \left(f(x) = f(z) \right) \right] \tag{1}$$

where z denotes samples drawn from $\mathcal{D}(z \mid S)$, the distribution of perturbed texts that preserve all pivot words in S. The final pivot set is required to satisfy $p_S > \tau$, where τ is a predefined threshold. A larger p_S indicates that preserving S makes it more likely for the predicted label to remain unchanged, implying that S is a stronger candidate for the pivot set.

It is generally intractable to compute p_S exactly. To address this, we allow an approximation with a tolerable error $\delta \in [0, 1]$ and adopt a probabilistic definition. Specifically, we regard S as a pivot word set if p_S satisfies the following condition:

$$P\left(p_S \geq \tau\right) \geq 1 - \delta \tag{2}$$

where δ is a predefined parameter controlling the acceptance threshold.

Multiple pivot word sets may satisfy the criterion. In such cases, we prioritize those with fewer constituent words. This preference arises because manipulating fewer tokens yields adversarial examples that closely resemble the original sentence, enhancing their stealthiness. In summary, the search for a pivot word set can be formulated as the following optimization problem:

$$\min_{S \; s.t. \; P(p_S \geq \tau) \geq 1-\delta} |S| \tag{3}$$

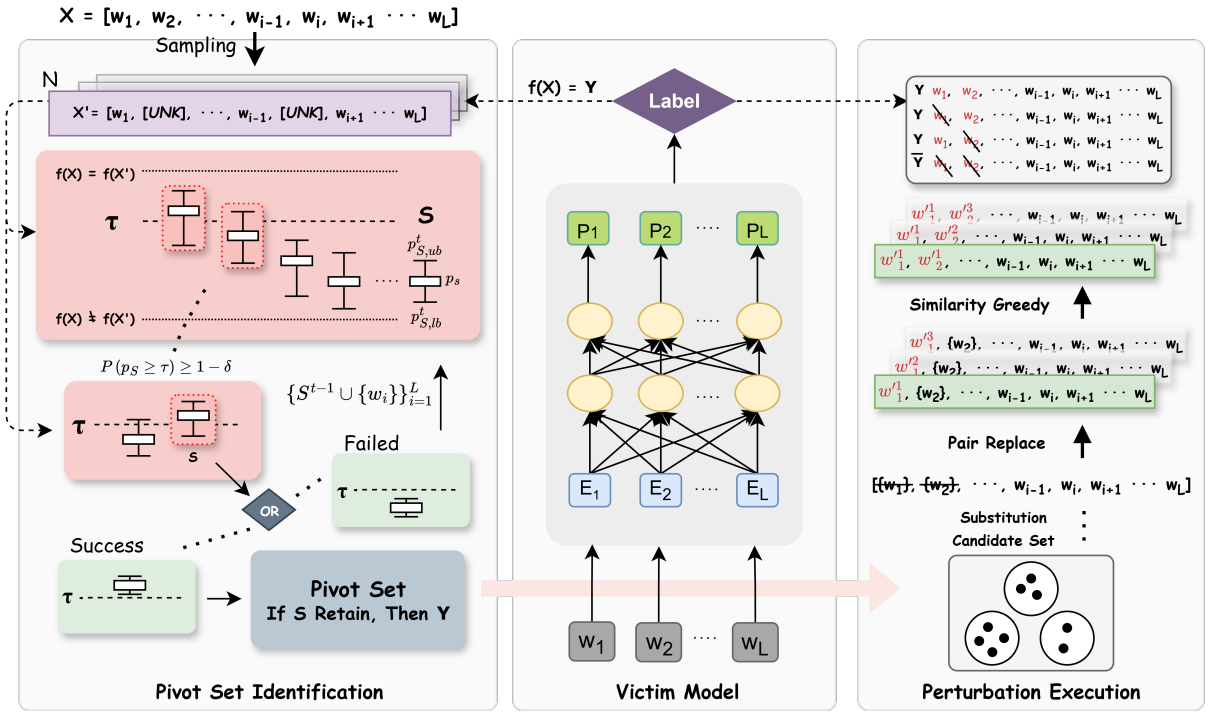


Figure 1: **Overview of PivotAttack.** The framework first performs Pivot Set Identification to locate words anchoring the model' s prediction, selecting high-precision pivot sets and refining their confidence intervals via a multi-armed bandit for accuracy. In the Perturbation Execution stage, substitution candidates are generated for pivot words, and the variant most similar to the original sentence is chosen as the final adversarial example.

3.1.1 Non-Actionable Attack Culling

To improve query efficiency, we first discard non-actionable instances whose labels are unlikely to flip within the perturbation bounds. For a sample $X = [w_1, \ldots, w_L]$, we generate N masked variants by replacing each token w_i with "[UNK]" at a fixed probability and compute the retention score p^0, estimating the likelihood that the model's prediction $f(X)$ remains unchanged:

$$p^0 = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(f(X) = f(X_i)) \tag{4}$$

Subsequently, we compute the lower bound of p^0 using the subsequent equation:

$$p^0_{lb} = \min \left\{ q \in [0, p^0] : d(p^0, q) < \frac{\beta_0}{N} \right\} \tag{5}$$

where $\beta_0 = -\log \delta$ is an exploration parameter, $d(p, q)$ represents the Kullback-Leibler (KL) divergence between two Bernoulli distributions, with its definition provided by Kaufmann and Kalyanakrishnan (2013):

$$d(p, q) = p \log \frac{p}{q} + (1 - p) \log \frac{1 - p}{1 - q} \tag{6}$$

If p^0_{lb} exceeds a predefined threshold, modifying the sample' s label is unlikely to alter the output, rendering it ineffective (a "dud") for adversarial attacks. Such instances are pruned, returning an empty set.

3.1.2 Construction of Pivot Set

Our objective is to identify a pivot set S that satisfies Equation 3. To keep $|S|$ minimal, we employ an incremental construction strategy. Starting from an empty set ($S^0 = \emptyset$), each iteration generates candidate sets by adding one word to the current set: $\mathcal{S}^t = \{S^{t-1} \cup \{w_i\}\}_{i=1}^{L}$. The candidate with the highest estimate precision is selected as the updated pivot set S^t. The process terminates once either S^t satisfies Equation 3 or a predefined budget limit is reached, returning S^t as the final pivot set.

To select the optimal pivot set at each iteration, we estimate the precision of each candidate set. For a candidate S, this is computed using samples from $\mathcal{D}(\cdot \mid S)$—the distribution of perturbed texts that preserve all words in S. To conserve query budget, we aim to minimize the number of samples required. Inspired by Ribeiro et al. (2018), we formulate this as a pure-exploration multi-armed bandit problem: each candidate pivot set S is an arm, its true precision under $\mathcal{D}(\cdot \mid S)$ is the latent reward, and pulling an arm corresponds to querying the model on a sampled instance $z \sim \mathcal{D}(z \mid S)$ and checking if the label changes (i.e., evaluating

$\mathbb{1}(f(x) = f(z)))$. Under this formulation, we can employ the KL-LUCB algorithm (Kaufmann and Kalyanakrishnan, 2013) to identify the best pivot set S.

According to KL-LUCB, at iteration t, the upper and lower confidence bounds of the estimated precision p_S for pivot set S are defined as:

$$p^t_{S,ub} = max\left\{q \in [p_S, 1] : d(p^t_S, q) < \frac{\beta(K,t)}{N^t_S}\right\} \tag{7}$$

$$p^t_{S,lb} = min\left\{q \in [0, p_S] : d(p^t_S, q) < \frac{\beta(K,t)}{N^t_S}\right\} \tag{8}$$

where $d(p, q)$ represents the Kullback-Leibler (KL) divergence between two Bernoulli distributions as defined in Equation 6, K is the number of arms; in the Pivot Set Selection stage, $K = L - |S|$, where L is the text length. N^t_S represents the number of times arm S has been pulled before iteration t. The exploration parameter $\beta(K, t)$ controls the confidence radius and increases logarithmically with t:

$$\beta(K,t) = \log\left(\frac{\lambda K t^\alpha}{\delta}\right) + \log\left(\log\left(\frac{\lambda K t^\alpha}{\delta}\right)\right) \tag{9}$$

where $\lambda > 0$ is a scaling constant, α determines the growth rate, and δ is the confidence parameter.

We initialize each rule's estimated precision and confidence interval using the sampling procedure in Section 3.1.1. Pivot sets are ranked by their estimated precision and partitioned into $\mathcal{S}^+$ and $\mathcal{S}^-$ based on whether their estimated precision exceeds the target threshold τ. To improve estimation accuracy, we iteratively tighten each pivot set's confidence interval. At each iteration, we select the pivot set $S \in \mathcal{S}^+$ with the smallest estimated precision lower bound and $S' \in \mathcal{S}^-$ with the largest estimated precision upper bound, and update both by pulling their corresponding arms.

The sampling process stops when $p_{S,lb}$ exceeds $p_{S',ub}$ within tolerance $\epsilon \in [0, 1]$. If S^* is the pivot set with the highest true precision, the following guarantee holds (Kaufmann and Kalyanakrishnan, 2013):

$$P(p_S \geq p_{S^*} - \epsilon) \geq 1 - \delta \tag{10}$$

For the pivot set S with the highest estimated precision in $\mathcal{S}^+$, we further verify whether it meets the precision criterion. If $p_{S,ub} \geq \tau$ but $p_{S,lb} < \tau$, its arm continues to be pulled until we can confidently determine $p_{S,lb} \geq \tau$ (valid pivot set) or $p_{S,ub} < \tau$ (invalid pivot set). The process of pivot set selection is outlined in Algorithm 1.

In addition, we use a real example on the MR dataset to illustrate the pivot set selection process (Figure 2). As shown in the figure, we demonstrate the pivot set identification process using the MR example "shaping one great character interaction throughline." Firstly, the sentence is randomly masked with "[UNK]" tokens to estimate the retention precision p_S. Using the KL–LUCB procedure, candidate arms are iteratively pulled to tighten their confidence bounds (u^t_i, l^t_i), ensuring that promising tokens are not overlooked. The token "great" initially achieves the highest precision, but since $p_S < \tau$, an additional token is incorporated. When the pair "great + character" satisfies $p_S > \tau$, it is finalized as the pivot set.

Algorithm 1 Workflow of Pivot Set Identification

```
function FindPivot(X, D, τ)
    hyperparameters: ε, δ
    S^0 ← ∅
    loop
        S_c^t ← GenerateCands(S^{t-1}, X)
        S^t ← BestCand(S_c^t, D, ε, δ)
        if S^t = ∅ then
            break
        if p_{S^t} ≥ τ then
            return S^t
function GenerateCands(S, X)
    for all S ∈ S, w_i ∈ X \ S do
        S ← S ∪ {S ∪ {w_i}}
    return S
function BestCand(S, D, ε, δ)
    Initialize estimates p^0 for all S ∈ S
    Partition S into S^+ and S^- based on p^0
    S ← arg max_{S∈S^+} p_S
    S' ← arg max_{S'∈S^-} u^t_{S'}
    while p^t_{S,ub} − p^t_{S,lb} > ε do
        Sample z ~ D(z | S) and z' ~ D(z' | S')
        Update p, p^t_{ub}, p^t_{lb} for S and S'
        S ← arg max_{S∈S^+} p_S
        S' ← arg max_{S'∈S^-} p^t_{S,ub}
    return S
```

In PivotAttack, the KL-LUCB component is relatively query-intensive. To prevent excessive budget consumption, PivotAttack allocates it a quota of γB, where $\gamma \in [0, 1]$ is a user-defined parameter and B is the total query budget. Once this quota is reached, KL-LUCB stops arm-pulling and returns the set S with the highest current estimate precision p as the pivot set. If all words in the pivot set have been attacked but budget remains, PivotAttack ranks non-pivot words by the precision estimates of their corresponding pivot set candidates and continues attacking the highest-ranked ones.

It is worth noting that during the construction of the pivot set, PivotAttack expands only the candidate set with the highest estimated precision, effectively following a greedy strategy. To obtain higher-quality pivot sets, one could instead employ beam search, which retains the top-k candidate sets at each step for further expansion—albeit at the cost of a higher query budget.

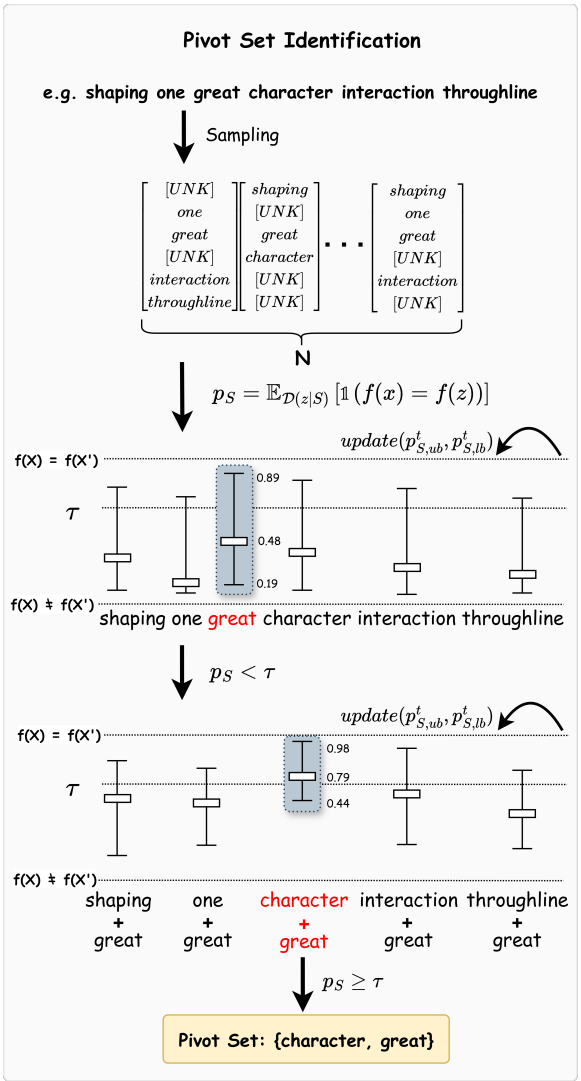


Figure 2: Pivot Set Identification Process on an MR Sample

3.1.3 Perturbation Execution

Given an input $X = [w_1, \ldots, w_i, \ldots, w_L]$ and its pivot set S, the perturbation stage of PivotAttack consists of three steps: generating substitution candidate sets, selecting adversarial samples, and skipping conspicuous samples.

Generating Substitution Candidate Sets. For each pivot token $w_i \in S$, we construct a substitution candidate set $C(w_i)$ by retrieving the M nearest vocabulary words to w_i in a pre-trained embedding space. Concretely, $C(w_i) = \{c_{i,1}, \ldots, c_{i,M}\}$, where each $c_{i,j}$ is among the M closest words to w_i under cosine similarity. We use counter-fitted word vectors (Mrkšić et al., 2016) to build this space, as they better preserve synonymy and antonymy for lexical substitution.

Selecting Adversarial Samples. Replacing w_i with $c_j \in C(w_i)$ yields a candidate adversarial sample $X_j^{'} = [w_1, \ldots, c_j, \ldots, w_L]$ For each pivot token we obtain M candidates. We select the candidate X'_j that maximizes cosine similarity to the original input X, thereby minimizing semantic drift:

$$X' = \arg\max_{X'_j \in X'_j} \cos\left(\text{embed}(X),\ \text{embed}(X'_j)\right) \tag{11}$$

where embed$(\cdot)$ denotes a sentence-level embedding used to measure semantic closeness.

Skipping Conspicuous Samples. To preserve stealth, we enforce a perturbation constraint based on the Perturbation Rate:

$$\text{Pert}(X, X') = \frac{1}{L}\sum_{i=1}^{L} \mathbb{1}(w_i \neq w'_i) \tag{12}$$

where L is the sentence length. Candidates exceeding a threshold h are skipped. We use a dynamic threshold that adapts to the remaining query budget:

$$h = \min\left(h_{\max},\ h_{\text{base}} + \frac{B_{\text{rm}}}{L}\right) \tag{13}$$

with user-defined $h_{\text{base}} < h_{\max}$ and B_{rm} denoting remaining queries. This adaptive rule balances stealth and flexibility as budget availability changes.

4 Experiment

We aim to investigate the following research questions through our experimental evaluation:

- RQ1: How does PivotAttack compare with existing black-box hard-label methods under a limited query budget?
- RQ2: How does the performance of PivotAttack vary across different query budgets?
- RQ3: How do different NLP tasks influence the effectiveness of PivotAttack?
- RQ4: What is the contribution of each component of PivotAttack to its overall performance?
- RQ5: How interpretable is PivotAttack relative to other attack models?

4.1 Experimental Setup

Datasets. We conducted our experiments on five publicly available text classification datasets: Yelp (Zhang et al., 2015), Yahoo (Zhang et al., 2015), MR (Pang and Lee, 2005), Amazon (Zhang et al., 2015), and SST-2 (Socher et al., 2013). To address RQ3, which examines performance across different NLP tasks, we additionally evaluated the attack methods on textual entailment using two datasets: SNLI (Conneau et al., 2017) and MultiNLI (Williams et al., 2018), the latter of which contains both the matched (MNLI-m)

Task	Dataset	Train	Test	Class	Length
Classification	Yelp	560K	38K	2	133
	Yahoo	1400K	60K	10	151
	MR	9K	1K	2	18
	Amazon	3600K	400K	2	79
	SST-2	70K	2K	2	8
Entailment	SNLI	570K	3K	3	20
	MNLI	433K	10K	3	11

Table 1: Dataset Statistics

and mismatched (MNLI-mm) splits. The dataset statistics are summarized in Table 1, with additional details provided in Appendix A.

Victim Models. For the text classification task, we employ three widely used deep learning models —WordCNN (Kim, 2014), WordLSTM (Hochreiter and Schmidhuber, 1997), and BERT (Devlin et al., 2019)—along with a large language model, Qwen2.5-1.5B (Yang et al., 2025), as the victim models. For the textual entailment task, we use BERT as the victim model. Further details about the victim models are provided in Appendix B.

Baselines. We have chosen the following existing hard-label attack algorithms as our baselines: HLBB (Maheshwary et al., 2021b), TextHoaxer (Ye et al., 2022c), LeapAttack (Ye et al., 2022a), TextHacker (Yu et al., 2022b), and LimeAttack (HaiZh et al., 2023). More details of the baselines are listed in the Appendix C.

Evaluation Metrics. The evaluation metrics used are Attack Success Rate (ASR), Perturbation Rate (Pert), and Semantic Similarity (Sim). ASR measures the effectiveness of the adversarial method (the proportion of examples on which the attack succeeds). Pert and Sim assess adversarial text quality—Pert quantifies the extent of modification (e.g., fraction of tokens changed) while Sim captures semantic preservation between original and adversarial texts.

Implementation Details. We set the query budget to $B = 100$, the KL-LUCB quota parameter to $\gamma = 0.8$, the precision threshold to $\tau = 0.85$, the maximum confidence error to $\epsilon = 0.9$, the tolerable prediction error to $\delta = 0.85$, the sampling count to $N = 5$, the substitution candidate set size to $M = 50$, and the base perturbation rate to $\rho = 0.1$. All experiments were conducted on an NVIDIA RTX 3080 Ti GPU (13 GB), and the results are averaged over three independent runs.

4.2 Overall Result (RQ1)

Performance Comparison. Table 2 reports results of textual adversarial attacks using PivotAttack and baselines across multiple datasets and victim models under a query budget of 100. PivotAttack consistently outperforms alternatives, particularly on long-text datasets (i.e., Yelp, Yahoo, Amazon), achieving both higher ASR and lower Pert. For example, on Yelp with an LSTM target, PivotAttack attains 16.8% ASR vs. 14.5% for TextHacker, with a Pert of 1.4% vs. 2.3% for TextHoaxer; against BERT, it achieves 9.7% ASR with Pert 1.0%, while all baselines remain below 7.4% ASR and above 2.0% Pert.

PivotAttack' s superiority stems from two factors: **First**, unlike other methods that first flip labels and then move toward the decision boundary from outside—often requiring excessive queries if the starting point lies far from the boundary (e.g., through many word modifications)—PivotAttack instead identifies pivot words that preserve the original label and advances from within the label-invariant region toward the boundary. This inherently reduces query consumption, enabling stronger performance under budget constraints. **Second**, by accounting for inter-word interactions during perturbation selection, PivotAttack captures the combinatorial effects of multiple edits—an especially important property for long-text datasets where several modifications may be needed to flip labels.

In contrast, methods like TextHoaxer and LeapAttack sometimes achieve lower Pert values but at the expense of reduced ASR.

Adversary Quality. To balance effectiveness and stealth, adversarial samples must preserve semantic similarity with the original text. We further evaluate attacks on the Yelp dataset with BERT as the victim model, using additional metrics: semantic similarity (Sim.) via USE (Bowman et al., 2015) and grammatical error rate (Gram.) via LanguageTool[2]. As shown in Table 3, PivotAttack maintains superior performance across these metrics while achieving the highest attack success.

4.3 Query Budget (RQ2)

We evaluate different attack methods on the MR and SST-2 datasets under varying query budgets (B = 50, 100, 150, 200, 300, 500). Due to space limitations, only CNN-MR and BERT–MR results are shown in Figure 3, with full results in Appendix D. As shown in the figure, PivotAttack consistently outperforms other methods in ASR as the query budget increases, with its advantage even becoming more pronounced. This is because a larger query budget allows the KL-LUCB component to perform more arm pulls, yielding more accurate precision estimates and thus identifying a better pivot set.

4.4 Transferability (RQ3)

To evaluate the performance of different text adversarial attack algorithms across NLP tasks, we conducted experiments on the textual entailment

[2] https://languagetool.org

Model	Attack	Yelp		Yahoo		MR		Amazon		SST-2	
		ASR.↑	Pert.↓	ASR.↑	Pert.↓	ASR.↑	Pert.↓	ASR.↑	Pert.↓	ASR.↑	Pert.↓
CNN	**PivotAttack**	**12.9**	**1.6**	**44.5**	**2.4**	**51.5**	6.0	**19.0**	**2.1**	**46.4**	6.1
	LimeAttack	11.1	2.3	43.2	2.9	48.8	5.1	17.0	2.4	43.6	5.8
	TextHacker	**12.9**	3.0	43.0	4.6	48.6	6.1	**19.0**	2.9	37.2	6.6
	LeapAttack	10.0	2.6	38.1	2.7	46.7	4.9	16.1	2.3	41.7	5.5
	TextHoaxer	9.8	2.6	38.0	2.8	42.3	**4.7**	16.0	2.5	37.5	**5.3**
	HLBB	11.9	2.9	39.1	3.0	41.3	5.3	16.8	2.9	37.3	5.7
LSTM	**PivotAttack**	**16.8**	**1.4**	**42.3**	**1.5**	**50.6**	5.1	**18.6**	**1.8**	**37.8**	6.1
	LimeAttack	11.9	2.5	39.3	2.9	48.3	5.0	18.2	2.4	36.5	5.8
	TextHacker	14.5	5.9	38.8	4.3	44.9	5.9	18.5	3.5	33.6	6.6
	LeapAttack	10.9	2.6	36.4	3.0	45.2	4.9	15.0	2.2	33.4	5.6
	TextHoaxer	9.5	2.3	34.4	3.8	42.8	**4.8**	14.4	2.5	29.2	**5.5**
	HLBB	11.1	2.7	37.9	3.0	41.7	5.4	17.6	3.1	27.0	5.9
BERT	**PivotAttack**	**9.7**	**1.0**	**39.7**	**1.9**	**47.5**	5.9	**15.5**	**2.2**	**30.6**	6.4
	LimeAttack	7.2	2.6	36.6	3.0	40.1	5.8	13.4	2.5	27.2	6.1
	TextHacker	7.4	2.5	34.6	3.9	38.5	6.2	14.8	3.6	19.2	6.9
	LeapAttack	6.1	2.6	32.2	2.9	37.6	5.0	12.0	2.6	22.5	6.2
	TextHoaxer	5.8	3.4	29.6	2.9	37.0	**4.9**	11.1	2.7	18.7	**5.7**
	HLBB	6.4	2.0	35.0	2.7	36.6	5.3	13.7	3.1	17.8	5.9
Qwen2.5	**PivotAttack**	**16.1**	**1.0**	**93.5**	**1.1**	**44.1**	6.4	21.8	**1.7**	**52.6**	6.1
	LimeAttack	13.5	2.2	92.7	2.5	43.1	5.3	**23.7**	3.0	49.6	**5.5**
	TextHacker	14.1	3.7	93.1	4.0	42.8	6.9	22.8	4.3	45.2	6.2
	LeapAttack	13.9	2.9	92.8	3.0	42.0	**5.1**	20.8	3.9	41.4	5.7
	TextHoaxer	12.5	3.1	91.3	3.1	41.1	6.4	21.2	3.7	42.5	6.0
	HLBB	13.9	2.7	82.7	2.8	35.9	5.3	18.6	3.2	37.0	5.9

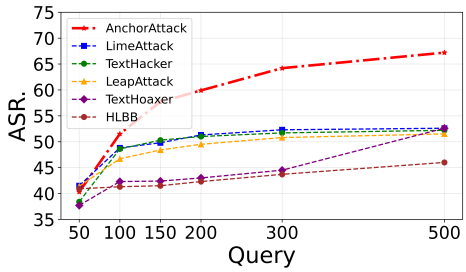
Table 2: Comparison of Attack Success Rate (ASR, %) and Perturbation Rate (Pert, %) under a 100-Query Budget

Attack	ASR.↑	Pert.↓	Sim.↑	Gram.↓
PivotAttack	**9.7**	**1**	**99.5**	0.31
LimeAttack	7.2	2.6	99.3	0.34
TextHacker	7.4	2.5	99.4	**0.27**
LeapAttack	6.1	2.6	99.3	0.55
TextHoaxer	5.8	3.4	99.4	0.73
HLBB	6.4	2	99.4	0.41

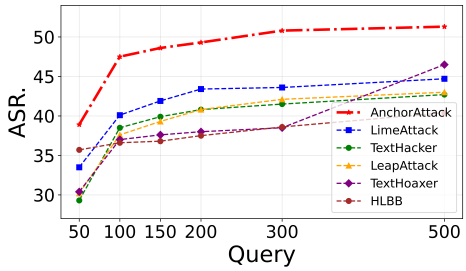
Table 3: Adversarial Sample Quality Comparison on BERT (Yelp, 100-Query Budget)

Attack	SNLI		MNLI-m		MNLI-mm	
	ASR.↑	Pert.↓	ASR.↑	Pert.↓	ASR.↑	Pert.↓
PivotAttack	**25.8**	7.9	**47.7**	7.7	**52.7**	7.6
LimeAttack	25.3	8.0	46.4	7.2	52.3	7.2
TextHacker	22.1	8.3	38.2	7.7	44.1	7.5
LeapAttack	24.4	**7.6**	43.1	**7.6**	45.9	**7.6**
TextHoaxer	23.6	7.7	42.2	7.2	46.8	**7.0**
HLBB	23.6	7.9	43.6	**7.1**	46.3	**7.0**

Table 4: Textual Entailment Attack Performance on BERT (100-Query Budget)



(a) CNN-MR



(b) BERT-MR

Figure 3: ASR Comparison Across Different Query Budgets on CNN-MR and BERT-MR

datasets SNLI and MNLI. The results, shown in Table 4, indicate that PivotAttack achieves the highest ASR while maintaining perturbation levels comparable to other methods.

4.5 Ablation Study (RQ4)

Ablation Study of PivotAttack. To evaluate the contribution of each component in PivotAttack, we conduct an ablation study on the long-text dataset Yelp and the short-text dataset MR, using an LSTM victim model with a query budget of 100. We test three variants: (1) replacing pivot set words with random words (*–Pivot set*); (2) replacing the dynamic perturbation threshold with a fixed value of 0.1 (*–Dynamic constraints*); and (3) ranking non-pivot words randomly instead

of reusing the intermediate KL-LUCB output (*−Reuse*).

Results in Table 5 show that while the perturbation rate remains nearly unchanged, ASR varies significantly. Removing the pivot set reduces ASR from 16.8% to 13.7% on Yelp and from 50.6% to 46.0% on MR, demonstrating that targeting pivot words more effectively flips labels. Disabling dynamic constraints also lowers ASR, highlighting the benefit of adaptive perturbation control. In contrast, randomly ranking non-pivot words leads to only a minor decrease, suggesting that attacking pivot words alone is typically sufficient.

Dataset	Method	ASR.↑	Pert.↓
Yelp	**PivotAttack**	**16.8**	**1.4**
	−Pivot set	13.7	1.4
	−Dynamic constrains	14.4	1.4
	−Reuse	16.6	1.4
MR	**PivotAttack**	**50.6**	5.1
	−Pivot set	46	5.2
	−Dynamic constrains	47.2	**4.8**
	−Reuse	50.2	5.2

Table 5: Ablation Study Results for PivotAttack

Abalation Study of Pviot Set Identification. To assess the contribution of each component in pivot set identification, we conduct an ablation study under a fixed query budget of 300. We evaluate three variants: (1) skipping non-actionable attack culling (*−Culling*), (2) omitting the multi-armed bandit refinement and selecting the pivot set directly from initial precision estimates (*−MAB*), and (3) disabling the tightening step when precision bounds do not meet their thresholds (*−Tighten*).

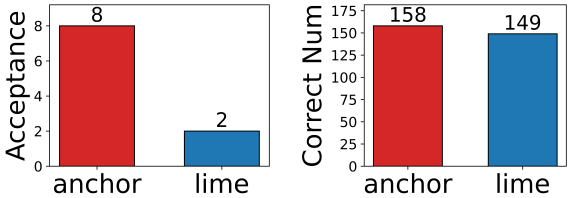
As shown in Table 6, omitting the multi-armed bandit refinement results in the largest performance drop, confirming that precision refinement is crucial for accurate pivot selection. Excluding culling also reduces performance, as it conserves queries that can be better allocated to pivot identification and word replacement. The effect of disabling the tightening step is minor, since it is rarely triggered in practice.

4.6 Human Evaluation (RQ5)

To assess the interpretability of PivotAttack, we conducted a human study with 10 participants, comparing it against the best-performing baseline, LimeAttack. We examined which algorithm' s selected perturbation words were more predictive and more reasonable to humans. For each method, participants first reviewed 10 sentences with the top-2 important words identified by the algorithm, then answered 20 multiple-choice questions based on SST-2 sentences. Each question included the top-2 words from both algorithms among five options, allowing us to evaluate participants' understanding of each model' s word-importance rationale. Finally, participants selected which method produced more reasonable results. As shown in Figure 4.6, PivotAttack achieved slightly higher predictability and was generally judged more reasonable. Participants noted that LimeAttack often emphasized functional words such as conjunctions or prepositions (e.g., "of"), whereas PivotAttack focused on semantically meaningful tokens better aligned with human intuition.

Dataset	Method	ASR.	Pert.
Yelp	**PivotAttack**	**15.2**	**1.2**
	−Culling	13.8	1.4
	−MAB	13.6	1.4
	−Tighten	14.8	1.3
MR	**PivotAttack**	**49.8**	**5.0**
	−Culling	47.9	**5.0**
	−MAB	46.3	5.1
	−Tighten	49.5	5.1

Table 6: Ablation Study Results for Pivot Set Identification



(a) Reasonableness (b) Prediction

Figure 4: Human evaluation of LimeAttack and PivotAttack on perturbed word selection predictability and reasonableness.

5 Conclusion

We presented PivotAttack, a query-efficient hard-label black-box text attack that identifies a pivot set—a small group of tokens critical to the model' s prediction. By perturbing these pivot words, it efficiently drives inputs toward the decision boundary. Experiments show that PivotAttack consistently outperforms baselines across datasets and victim models under limited query budgets, demonstrating the effectiveness of targeting pivot words and modeling inter-word dependencies. With moderately increased budgets, the multi-armed bandit component further refines pivot-set selection, maintaining its advantage, while evaluations on diverse NLP tasks and interpretability analyses confirm its robustness and practical utility.

Limitation

Although PivotAttack outperforms baselines under limited query budgets, the KL-LUCB component of the multi-armed bandit used for pivot set

identification is relatively query-intensive. As a result, we currently rely on a greedy search to select the best pivot set under constrained budgets, which prevents the use of more advanced strategies, such as beam search, for potentially better pivot selection. In future work, we plan to investigate methods to reduce the query cost of the multi-armed bandit component.

References

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Yiyi Chen, Qiongkai Xu, and Johannes Bjerva. 2025. ALGEN: few-shot inversion attacks on textual embeddings via cross-model alignment and generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 24330–24348. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 670–680. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. In *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics. [link].

Z. Dong and Q. Dong. 2006. *Hownet And The Computation Of Meaning (With Cd-rom)*. World Scientific Publishing Company.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 31–36. Association for Computational Linguistics.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5747–5757. Association for Computational Linguistics.

HaiZh, Qingyang Zhao, Weiwei Shang, Yuren Wu, and Kai Liu. 2023. Limeattack: Local explainable method for textual hard-label adversarial attack. *arXiv preprint arXiv:2308.00319*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Bairu Hou, Jinghan Jia, Yihua Zhang, Guanhua Zhang, Yang Zhang, Sijia Liu, and Shiyu Chang. 2023. Textgrad: Advancing robustness evaluation in NLP by gradient-driven optimization. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

Emilie Kaufmann and Shivaram Kalyanakrishnan. 2013. Information complexity in bandit subset selection. In *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 228–251, Princeton, NJ, USA. PMLR.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Deokjae Lee, Seungyong Moon, Junhyeok Lee, and Hyun Oh Song. 2022. Query-efficient and scalable black-box adversarial attacks on discrete sequential data via bayesian optimization. *CoRR*, abs/2206.08575.

Guoyi Li, Bingkang Shi, Zongzhen Liu, Dehan Kong, Yulei Wu, Xiaodan Zhang, Longtao Huang, and Honglei Lyu. 2023. Adversarial text generation by search and learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 15722–15738. Association for Computational Linguistics.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021a. Generating natural language attacks in a hard label black box setting. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium*

on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, pages 13525–13533. AAAI Press.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021b. Generating natural language attacks in a hard label black box setting. In *AAAI*.

Zhao Meng and Roger Wattenhofer. 2020. A geometry-inspired attack for generating natural language adversarial examples. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6679–6689. International Committee on Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *NAACL-HLT*, pages 142–148.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*.

Shilin Qiu, Qihe Liu, Shijie Zhou, Min Gou, Zhun Zhang, and Zhewei Wu. 2025. Hard-label adversarial attack with dual-granularity optimization on texts. *Neurocomputing*, 634:129839.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California. Association for Computational Linguistics.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1527–1535. AAAI Press.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Boxin Wang, Chejian Xu, Xiangyu Liu, Yu Cheng, and Bo Li. 2022. Semattack: Natural textual attacks via different semantic spaces. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 176–205. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 24 others. 2025. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Muchao Ye, Jiazhuo Chen, Chenglin Miao, Ting Wang, and Fenglong Ma. 2022a. Leapattack: Hard-label adversarial attack on text via gradient-based optimization. In *KDD*, pages 2307–2315.

Muchao Ye, Jinghui Chen, Chenglin Miao, Ting Wang, and Fenglong Ma. 2022b. Leapattack: Hard-label adversarial attack on text via gradient-based optimization. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 2307–2315. ACM.

Muchao Ye, Chenglin Miao, Ting Wang, and Fenglong Ma. 2022c. Texthoaxer: Budgeted hard-label adversarial attacks on text. In *AAAI*.

Zhen Yu, Xiaosen Wang, Wanxiang Che, and Kun He. 2022a. Texthacker: Learning based hybrid local search algorithm for text hard-label adversarial attack. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 622–637. Association for Computational Linguistics.

Zhengyan Yu, Xiaosen Wang, Wanxiang Che, and Kun He. 2022b. Learning-based hybrid local search for the hard-label textual attack. *arXiv preprint arXiv:2201.08193*.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6066–6080. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NeurIPS*.

A Dataset Details

Our experiments utilized five text classification datasets: Yelp (Zhang et al., 2015), Yahoo (Zhang et al., 2015), MR (Pang and Lee, 2005), Amazon (Zhang et al., 2015), and SST-2 (Socher et al., 2013). Consistent with prior work on text adversarial attacks (HaiZh et al., 2023; Maheshwary et al., 2021b), we constructed our test sets by sampling 1000 examples from each corpus. Specifically, for Yelp, Yahoo, and MR, we utilized the publicly available datasets released by HLBB. For Amazon and SST-2, our samples were drawn randomly from their respective corpora. Notably, certain entries in the SST-2 dataset contained very few tokens, making them unsuitable for attack within our Pert budget; consequently, we restricted our sampling to sentences exceeding 10 tokens in length. Details regarding these datasets are provided below:

- **Yelp**: User reviews annotated with binary sentiment labels (positive/negative).
- **Yahoo**: Community QA dataset for topic classification across 10 categories: Society & Culture, Science & Mathematics, Health, Education & Reference, Computers & Internet, Sports, Business & Finance, Entertainment & Music, Family & Relationships, Politics & Government.
- **MR**: Binary sentiment classification task on movie reviews using the standard train/test split.
- **Amazon**: E-commerce review dataset; we use the binary polarity version (positive/negative).
- **SST-2**: Stanford Sentiment Treebank (SST) version adapted for binary classification at the sentence level.

Our evaluation on textual entailment utilizes two benchmark datasets: SNLI (Conneau et al., 2017) and MultiNLI (Williams et al., 2018), with the latter encompassing both matched (MNLI-m) and mismatched (MNLI-mm) partitions. The experimental data for both datasets were sourced from the public repository accompanying the HLBB implementation. Details of each dataset are provided below:

- **SNLI:** Consists of human-annotated premise-hypothesis pairs categorized into three classes: entailment, contradiction, and neutrality.
- **MultiNLI:** Represents a comprehensive NLI resource offering two distinct evaluation settings—a matched domain (*MNLI-m*) and a challenging mismatched domain (*MNLI-mm*).

Dataset	CNN	LSTM	BERT	Qwen2.5-1b
Yelp	93.6	94.6	96.5	89.0
Yahoo	71.1	73.7	79.1	23.7
MR	76.5	78	85	79.7
Amazon	91.1	92.2	94.2	88.5
SST-2	86.4	87.8	98.6	85.3
SNLI	–	–	89.1	–
MNLI-m	–	–	85.1	–
MNLI-mm	–	–	82.2	–

Table 7: Original Accuracy of the Victim Models

B Victim Model Details

We evaluated four model families: CNN, LSTM, BERT, and the large language model Qwen. Architectural details are as follows:

- **CNN**: WordCNN with three convolutional kernels (sizes 3, 4, 5) and 100 filters each, followed by Dropout (p=0.3) and 200-dimensional GloVe embeddings (trained on the 6B token dataset).
- **LSTM**: Bidirectional WordLSTM with 150 hidden units, using the same initialization and regularization as CNN.
- **BERT**: Full BERT transformer stack (12 layers, 768 hidden size, 12 attention heads) with fixed sequence length of 256 tokens (via padding/truncation).
- **Qwen**: Qwen2.5-1.5B model deployed in a zero-shot setting. The prompts used are illustrated in Figure 5.

The original accuracy of the victim models on various classification and entailment datasets is reported in Table 7.

C Baseline Details

We compare PivotAttack against five representative hard-label attack algorithms:

- **HLBB (Maheshwary et al., 2021b)**: A population-based (genetic) search over sentences that favors candidates with higher semantic similarity and fewer edits under a strict query budget.
- **TextHoaxer (Ye et al., 2022c)**: A greedy heuristic that prioritizes token positions, using a few probing queries to determine whether a substitution is worthwhile before committing.
- **LeapAttack (Ye et al., 2022a)**: Employs finite-difference boundary exploration with gradient-like directions to jointly select positions and synonyms, pushing inputs into the misclassification region.

2 classes prompt

```
Help me determine whether the following
sentence is negative or positive.
Answer with 0 for negative and 1 for
positive, and only respond with 0 or 1.
```

(a) Yelp (2 classes)

10 classes prompt

```
You are a strict Yahoo Answers topic classifier.
Return ONLY one digit 0-9 with NO other text.
Mapping: 0=Society & Culture, 1=Science &
Mathematics, 2=Health...
Examples:
Q: "Would my girlfriend break up ..." -> 0
Q: "Whats an easy way to visualize..." -> 1
Q: "when I'm trying too lose..." -> 2
...
Answer:
```

(b) Yahoo (10 classes)

Figure 5: Zero-shot LLM prompts used in hard-label evaluation on different classes. It shows the exact prompts we use during evaluation.

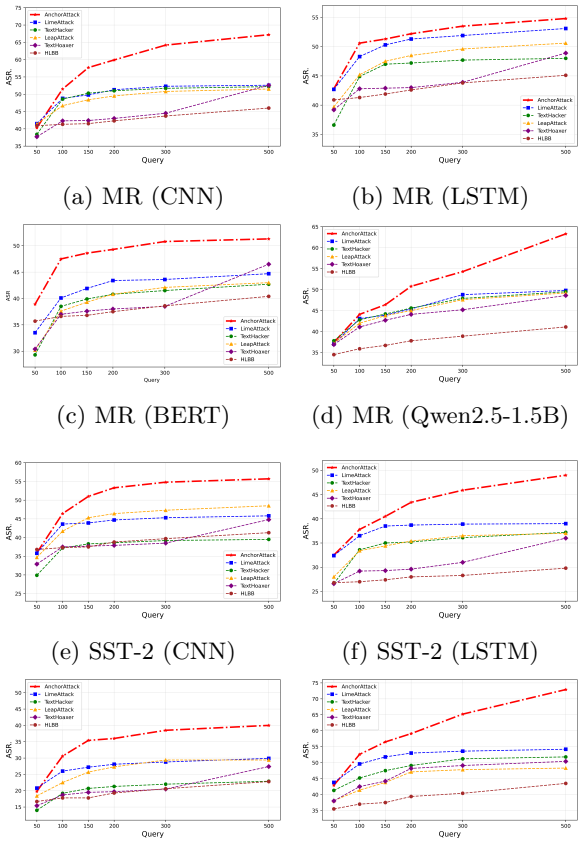
- **TextHacker (Yu et al., 2022b)**: Maintains an online-updated word-importance table derived from edit-flip history, which guides a hybrid local search.
- **LimeAttack (HaiZh et al., 2023)**: Uses LIME-style local perturbation sampling around the current input to estimate marginal word importance, followed by rank-and-replace under a query budget.

D Query Budget Results

Additional results on the MR and SST-2 datasets under varying query budgets (i.e., (B = 50, 100, 150, 200, 300, 500)) are presented in Figure 6. As shown, PivotAttack consistently surpasses all baseline methods across different datasets and victim models.

E Human Evaluation Details

To assess the interpretability of PivotAttack, we conducted a survey with 10 computer science students, comparing it against LimeAttack. The algorithm names were masked. As shown in Figure 7a, participants were first asked to review 10 prediction examples, each consisting of a sentence along with the two most important keywords selected by the algorithm. Subsequently, in the testing phase (Figure 7a, participants were required to complete 20 multiple-choice questions. Each question provided five candidate keywords, from which participants had to select the two words they believed the algorithm would identify as most important. In this experiment, selecting at least one correct keyword was considered a correct response.



(a) MR (CNN) (b) MR (LSTM)

(c) MR (BERT) (d) MR (Qwen2.5-1.5B)

(e) SST-2 (CNN) (f) SST-2 (LSTM)

(g) SST-2 (BERT) (h) SST-2 (Qwen2.5-1.5B)

Figure 6: ASR of different models under different query budgets on MR and SST-2.

it 's hard to resist his enthusiasm

`Method1 selection: resist, hard`

greatly impressed by the skill of the actors involved in the enterprise

A greatly
B impressed
C skill
D actors
E enterprise
`(Answer: B, C)`

(a) Example of PivotAttack

it 's hard to resist his enthusiasm

`Method2 selection: even, it`

greatly impressed by the skill of the actors involved in the enterprise

A greatly
B impressed
C skill
D actors
E enterprise
`(Answer: A, D)`

(b) Example of LimeAttack

Figure 7: Example of the Human Evaluation Survey