PivotAttack: Rethinking the Search Trajectory in Hard-Label Text Attacks via Pivot Words

Yuzhi Liang[*†] **Shiliang Xiao**[*] **Jingsong Wei** **Qiliang Lin** **Xia Li**
School of Information Science and Technology
Guangdong University of Foreign Studies
{yzliang, xiali}@gdufs.edu.cn
{slxiao, jswei, qllin}@mail.gdufs.edu.cn

Abstract

Existing hard-label text attacks often rely on inefficient "outside-in" strategies that traverse vast search spaces. We propose PivotAttack, a query-efficient "inside-out" framework. It employs a Multi-Armed Bandit algorithm to identify *Pivot Sets*—combinatorial token groups acting as prediction anchors—and strategically perturbs them to induce label flips. This approach captures inter-word dependencies and minimizes query costs. Extensive experiments across traditional models and Large Language Models demonstrate that PivotAttack consistently outperforms state-of-the-art baselines in both Attack Success Rate and query efficiency.

1 Introduction

Deep neural networks have achieved remarkable performance across natural language processing tasks, yet they remain highly vulnerable to adversarial examples—small, often imperceptible perturbations that induce misclassification. Among various threat models, the hard-label black-box setting represents the most restrictive and realistic scenario: the attacker queries the target model and receives only a discrete class label, without access to gradients, confidence scores, or internal states. This constraint poses a significant challenge: *how to generate semantically faithful adversarial examples with minimal queries?*

Existing hard-label attacks typically rely on approximating the decision boundary, but they suffer from intrinsic inefficiencies. **First**, many state-of-the-art methods, such as HyGloadAttack (Liu et al., 2024) and TextHoaxer (Ye et al., 2022b), adopt an "outside-in" initialization strategy. They often start with a heavily perturbed text far from the original semantics and iteratively refine it to approach the decision boundary. This trajectory traverses a vast search space, inevitably consuming excessive queries and degrading textual quality. **Second**, methods like VIWHard (Zhang et al., 2025) and LimeAttack (Zhu et al., 2024) attempt to identify important words via local surrogates but typically score tokens independently. This independence assumption ignores the combinatorial nature of language, often highlighting functional words while missing multi-word semantic anchors, leading to suboptimal perturbation sets. **Finally**, most methods lack interpretability, relying on opaque continuous relaxations or complex heuristic searches that offer little insight into why specific substitutions trigger a label flip.

To address these challenges, we propose PivotAttack[1], a query-efficient attack that fundamentally shifts the paradigm from "approximating the boundary" to "breaking the load-bearing walls." Specifically, PivotAttack implements a novel "inside-out" strategy. It starts from the original text and identifies a *Pivot Set*—a compact group of tokens that anchors the models prediction. We observe that as long as this Pivot Set remains intact, the prediction is robust; however, strategically perturbing these tokens triggers a disproportionate collapse in model confidence, efficiently driving the instance across the decision boundary.

Methodologically, we formulate Pivot Set identification as a Multi-Armed Bandit (MAB) problem, employing the KL-LUCB algorithm to rigorously estimate the influence of token combinations under a limited budget. This rigorous formulation allows PivotAttack to distinguish true semantic anchors from statistical noise.

Our contributions are summarized as follows:

- We propose the novel "inside-out" strategy, which attacks pivot words to advance toward the decision boundary from within the label-

[*]Equal contribution.
[†]Corresponding author.

[1]The source code for PivotAttack is available at https://github.com/s1xiao/PivotAttack

Findings of the Association for Computational Linguistics: ACL 2026, pages 3190–3207

invariant region. This approach is significantly more query-efficient than mainstream "outside-in" methods that require expensive refinement steps.

- Unlike methods that rank tokens in isolation, PivotAttack explicitly accounts for inter-word interactions when selecting perturbations, enabling the identification of effective multi-word edits.
- We formulate Pivot Set selection via a multi-armed bandit framework, which generates human-readable intermediate outputs at each iteration, thereby improving both the interpretability and traceability of the attack behavior.

Extensive experiments verify that PivotAttack consistently outperforms baselines across varying architectures. Notably, on Large Language Models, PivotAttack demonstrates exceptional efficacy: it exposes the high vulnerability of zero-shot models and, more importantly, remains the most effective attacker against robust Fine-tuned LLMs, surpassing state-of-the-art methods in both success rate and query efficiency.

2 Related Work

Research on textual adversarial attacks is formally categorized based on the adversary's knowledge of the victim model.

White-box and Soft-label Attacks. White-box attacks assume full transparency, allowing direct optimization via gradients. Ebrahimi et al. (2018) proposed HotFlip for gradient-based character perturbations, while Guo et al. (2021) introduced GBDA to optimize adversarial distributions via Gumbel-Softmax. More recently, TextGrad (Hou et al., 2023) utilized gradients for precise robustness assessment. In the black-box setting, soft-label attacks rely on output probabilities. Early approaches focused on synonym replacement strategies guided by lexical resources like HowNet and WordNet (Ren et al., 2019; Zang et al., 2020). Others optimize word selection through importance ranking (TextFooler; Jin et al., 2020), Bayesian search (Lee et al., 2022), or conditional generative models (Li et al., 2023). Recently, Chen et al. (2025) proposed ALGEN, leveraging cross-model alignment for few-shot embedding inversion.

Hard-label Black-box Attacks. This setting is the most challenging, as attackers can access only the final discrete prediction. Existing methodologies generally adopt evolutionary or boundary-approximation strategies. Population-based methods, such as HLBB (Maheshwary et al., 2021a), utilize genetic algorithms to evolve candidates but are often query-intensive. Boundary-approximation methods aim to locate and traverse the decision boundary. LeapAttack (Ye et al., 2022a) exploits directional cues, while GeoAttack (Meng and Wattenhofer, 2020) and TextHoaxer (Ye et al., 2022b) optimize within geometric or continuous embedding spaces. To mitigate local optima, HyGloadAttack (Liu et al., 2024) introduces a hybrid optimization framework with perturbation matrices. Refinement-based methods often start with effective substitutions or noise. TextHacker (Yu et al., 2022a) combines hybrid local search with attack history, and LimeAttack (Zhu et al., 2024) employs local surrogate models (LIME) to estimate token importance. Most recently, VIWHard (Zhang et al., 2025) utilized masked language models to identify critical words.

3 Methodology

We defer the formal problem formulation to Appendix A. Before detailing the algorithm, we clarify the intuition behind PivotAttack. Unlike traditional text attack methods that focus on identifying tokens to flip the predicted label immediately, PivotAttack targets robustness anchors—a specific set of tokens whose preservation ensures label stability. Conceptually, these tokens function as the "load-bearing walls" of the prediction: even if the majority of the sentence remains semantically intact, perturbing the Pivot Set often triggers a disproportionate collapse in model confidence.

Specifically, PivotAttack operates in two stages: (1) Pivot Set Identification, where a multi-armed bandit strategy is employed to isolate pivot words that anchor the models prediction (implying that perturbing non-pivot words leaves the output largely unchanged); and (2) Perturbation Execution, where synonym substitutions are applied specifically to these 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

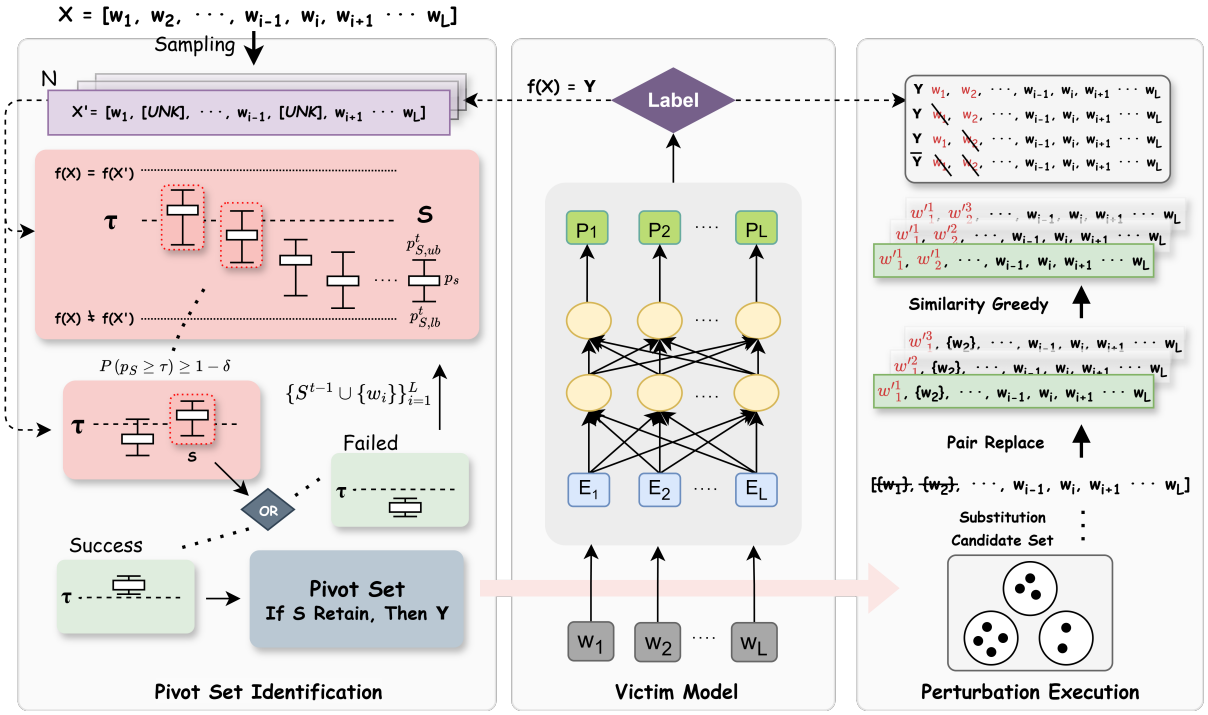


Figure 1: **The overall workflow of PivotAttack.** The framework employs an MAB algorithm to identify prediction-anchoring Pivot Sets, then crafts adversarial examples by performing similarity-constrained substitutions on these isolated pivot words..

sample X', the models prediction is very likely to remain unchanged (i.e., $f(X) = f(X')$).

We formulate pivot word selection as a MAB problem, treating each word as an arm. This bounds the probability of identifying the optimal K arms within a high confidence interval. Formally, for a given input x and a candidate set S, we estimate the retention precision p_S, which quantifies the probability that the victim model f maintains its original prediction when non-pivot words are perturbed:

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

where $\mathbb{1}(\cdot)$ is the indicator function, and $z \sim \mathcal{D}(z \mid S)$ denotes perturbed texts preserving S. The final Pivot Set must satisfy $p_S > \tau$ (a predefined threshold). Intuitively, a larger p_S indicates S more strongly anchors the original prediction.

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 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 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 Set can be formulated as the following optimization problem:

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

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 models 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):

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

If p^0_{lb} exceeds a predefined threshold, modifying the samples label is unlikely to alter the output, rendering it non-actionable 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 Eq. 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 estimated retention precision is selected as the updated Pivot Set S^t. The process terminates once either S^t satisfies Eq. 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 retention 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 retention 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 retention precision p_S for Pivot Set S are defined as:

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

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

where $d(p,q)$ represents the KL divergence between two Bernoulli distributions as defined in Eq. 6, and 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:

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

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

We initialize each rule's estimated retention precision and confidence interval using the sampling procedure in Section 3.1.1. Pivot Sets are ranked by their estimated retention precision and partitioned into $\mathcal{S}^+$ and $\mathcal{S}^-$ based on whether their estimated retention precision exceeds the target threshold τ. To improve estimation accuracy, we iteratively tighten each Pivot Sets confidence interval. At each iteration, we select the Pivot Set $S \in \mathcal{S}^+$ with the smallest estimated retention precision lower bound and $S' \in \mathcal{S}^-$ with the largest estimated retention 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 retention precision, the following guarantee holds (Kaufmann and Kalyanakrishnan, 2013):

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

For the Pivot Set S with the highest estimated retention precision in $\mathcal{S}^+$, we further verify whether it meets the retention 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.

Figure 2 illustrates the Pivot Set selection process using a real 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 retention precision, but since $p_S < \tau$, an additional

Algorithm 1 Workflow of Pivot Set Identification

function FINDPIVOT($X, \mathcal{D}, \tau$)
  hyperparameters: ϵ, δ
  $S^0 \leftarrow \emptyset$
  loop
    $\mathcal{S}_c^t \leftarrow$ GENERATECANDS($\mathcal{S}^{t-1}, X$)
    $S^t \leftarrow$ BESTCAND($\mathcal{S}_c^t, \mathcal{D}, \epsilon, \delta$)
    if $S^t = \emptyset$ **then**
      break
    if $p_{S^t} \geq \tau$ **then**
      return S^t

function GENERATECANDS($\mathcal{S}, X$)
  for all $S \in \mathcal{S}, w_i \in X \setminus S$ **do**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{S \cup \{w_i\}\}$
  return $\mathcal{S}$

function BESTCAND($\mathcal{S}, \mathcal{D}, \epsilon, \delta$)
  Initialize estimates p^0 for all $S \in \mathcal{S}$
  Partition $\mathcal{S}$ into $\mathcal{S}^+$ and $\mathcal{S}^-$ based on p^0
  $S \leftarrow \arg\max_{S \in \mathcal{S}^+} p_S$
  $S' \leftarrow \arg\max_{S' \in \mathcal{S}^-} u_{S'}^t$
  while $p_{S,ub}^t - p_{S,lb}^t > \epsilon$ **do**
    Sample $z \sim \mathcal{D}(z \mid S)$ and $z' \sim \mathcal{D}(z' \mid S')$
    Update p, p_{ub}^t, p_{lb}^t for S and S' according to Eq. 1, Eq. 7, and Eq. 8
    $S \leftarrow \arg\max_{S \in \mathcal{S}^+} p_S$
    $S' \leftarrow \arg\max_{S' \in \mathcal{S}^-} p_{S,ub}^t$
  return S

token is incorporated. When the pair great + character satisfies $p_S > \tau$, it is finalized as the Pivot Set.

To limit the query-intensive KL-LUCB component, PivotAttack allocates it a budget quota of γB ($\gamma \in [0, 1]$, where B is the total budget). Upon reaching this limit, it returns the set S with the highest estimated retention precision p as the Pivot Set. If budget remains after attacking all pivot words, PivotAttack sequentially attacks non-pivot words prioritized by their candidates' estimated p.

It is worth noting that during the construction of the Pivot Set, PivotAttack expands only the candidate set with the highest estimated retention 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.

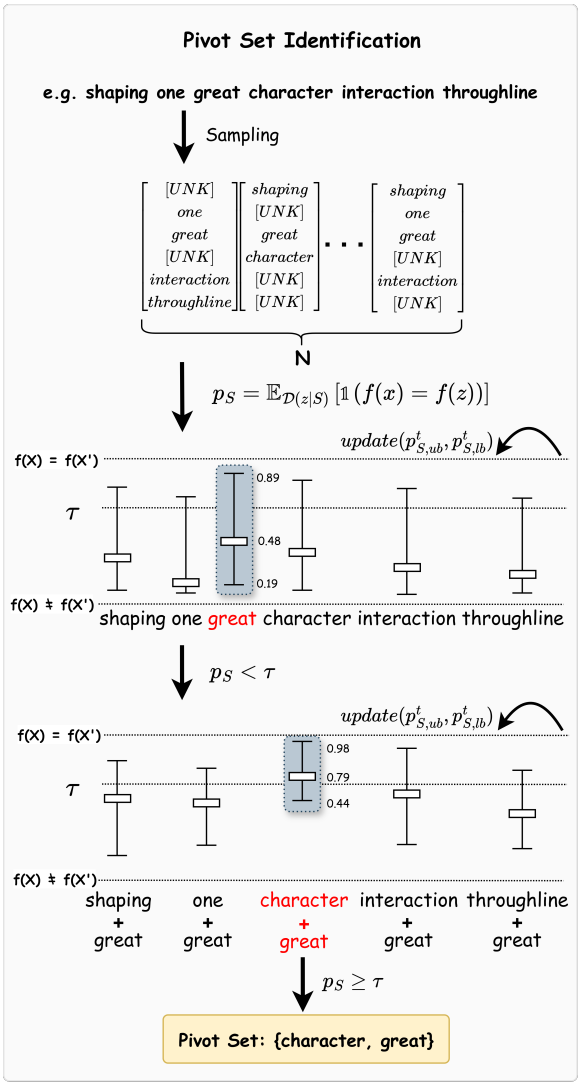


Figure 2: Pivot Set Identification on MR

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 pretrained 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 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, we evaluated the performance of PivotAttack on the textual entailment and question answering (QA) tasks. For textual entailment, we utilized two datasets: SNLI (Conneau et al., 2017) and MultiNLI (Williams et al., 2018). For the QA task, we employed the SQuAD dataset (Rajpurkar et al., 2016).

Victim Models. Our evaluation covers a broad range of model architectures. Following Zhu et al. (2024), we employ WordCNN (Kim, 2014), WordLSTM (Hochreiter and Schmidhuber, 1997), and BERT (Devlin et al., 2019) as representative classification models. We further include efficient encoder variants, namely ALBERT (Lan et al., 2020) and DistilBERT (Sanh et al., 2019), as well as recent large language models (LLMs): Qwen2.5-1.5B, evaluated in both zero-shot and fine-tuned settings (Yang et al., 2025), and Gemma 3 (Team, 2025), evaluated in the zero-shot setting . For textual entailment and QA, we adopt BERT. Additional model details are provided in Appendix C.

Baselines. We have chosen the following existing hard-label attack algorithms as our baselines: HyGloadAttack (Liu et al., 2024), VIWHard (Zhang et al., 2025), HLBB (Maheshwary et al., 2021b), TextHoaxer (Ye et al., 2022b), LeapAttack (Ye et al., 2022a), TextHacker (Yu et al., 2022b), and LimeAttack (Zhu et al., 2024). More details of the baselines are listed in the Appendix D.

Details of the evaluation metrics and implementation are provided in Appendix E and Appendix F, respectively.

4.2 Overall Result (RQ1)

Performance Comparison. Table 1 compares PivotAttack against seven baselines under a strict 100-query budget. PivotAttack consistently strikes a superior balance between high ASR and low perturbation across both traditional architectures and LLMs. For instance, on WordLSTM (Yelp), PivotAttack attains 16.8% ASR (1.4% Pert) compared to TextHacker's 14.5% (5.9% Pert). Against BERT, it achieves 9.7% ASR with only 1.0% perturbation, while baselines either lag below 8.2% ASR or incur higher costs. This dominance extends to LLMs: on Qwen2.5 (Zero-shot/Yahoo), PivotAttack reaches 93.5% ASR with a mere 1.1% perturbation, significantly outperforming TextHacker (4.0% Pert). Even against the robust fine-tuned Qwen2.5, PivotAttack remains the top performer on 4 out of 5 datasets. Additional results on WordCNN and ALBERT are reported

Model	Attack	Yelp		Yahoo		MR		Amazon		SST-2	
		ASR↑	Pert↓	ASR↑	Pert↓	ASR↑	Pert↓	ASR↑	Pert↓	ASR↑	Pert↓
WordLSTM	**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
	VIWHard	13.3	1.5	41.9	1.6	47.3	5.2	14.7	1.9	35.3	6.1
	HyGloadAttack	11.7	2.3	37.9	3.4	46.2	5.3	17.8	2.4	33.5	6.4
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
	VIWHard	8.2	1.2	39.0	2.0	36.3	5.5	11.4	**1.9**	26.1	6.8
	HyGloadAttack	**9.7**	2.1	36.0	3.4	41.3	5.7	**16.3**	3.3	25.0	6.7
DistilBERT	**PivotAttack**	**8.6**	**1.2**	**37.1**	1.8	**33.5**	6.2	**11.7**	**1.5**	**32.4**	6.5
	LimeAttack	8.0	2.4	36.1	2.3	28.7	5.4	10.8	1.6	27.8	**6.0**
	TextHacker	8.0	2.3	35.4	2.6	27.1	6.0	10.5	2.0	27.1	6.6
	LeapAttack	7.6	2.6	33.4	3.7	24.3	5.9	10.5	2.4	25.9	6.8
	TextHoaxer	7.0	2.1	32.2	3.5	21.1	**5.5**	9.5	2.5	20.0	6.6
	HLBB	6.8	2.1	31.2	3.7	17.0	6.0	10.9	3.2	16.2	6.8
	VIWHard	6.7	1.3	36.8	**1.7**	21.9	5.7	8.3	1.6	27.6	7.0
	HyGloadAttack	8.4	2.0	33.5	3.6	26.2	5.7	11.5	2.4	24.4	6.8
Gemma 3 (Zero-shot)	**PivotAttack**	**39.8**	**0.6**	**88.7**	1.0	**62.2**	4.8	**34.9**	**1.2**	**54.6**	5.9
	LimeAttack	38.6	0.8	84.1	**0.8**	51.2	4.9	34	1.3	53.4	**5.5**
	TextHacker	34.1	2.1	85.5	1.6	59.6	5.0	33.6	3.5	51.2	6.1
	LeapAttack	29.8	2.4	83.2	2.1	60.3	4.6	33.2	2.1	50.8	6.0
	TextHoaxer	25.5	2.3	82.8	1.8	53.5	4.3	32.1	2.4	50.3	6.2
	HLBB	23.3	8.2	83.1	2.5	52.6	5.0	32.6	4.5	49.2	6.4
	VIWHard	33.1	1.0	86.1	1.3	56.9	**4.2**	32.7	3.2	53.1	6.3
	HyGloadAttack	35.8	2.1	87.5	2.5	60.1	4.7	33.4	3.9	53.8	6.7
Qwen2.5 (Zero-shot)	**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
	VIWHard	14.1	1.3	89.2	1.3	36.9	6.0	19.2	2.1	48.2	6.6
	HyGloadAttack	13.6	2.0	83.7	1.9	35.1	5.4	17.6	2.5	45.9	6.7
Qwen2.5 (Fine-tuned)	**PivotAttack**	**3.9**	**1.6**	**46.2**	**1.2**	**32.1**	6.4	**6.5**	**2.0**	**29.6**	6.4
	LimeAttack	3.7	2	41.3	1.6	30.0	5.5	6.0	2.5	25.6	**6.1**
	TextHacker	3.7	2.3	45.5	1.8	29.6	6.0	6.0	2.2	26.4	6.2
	LeapAttack	3.4	2.2	42.7	2.1	28.3	5.7	6.3	2.5	25.1	6.3
	TextHoaxer	3.2	2.5	40.8	1.8	28.1	**5.4**	5.9	2.8	24.5	6.7
	HLBB	3.6	3.1	41.1	2.4	24.5	5.6	5.7	2.4	26.8	6.3
	VIWHard	2.9	2.3	44.3	1.4	26.4	6.0	5.8	2.2	24.8	6.4
	HyGloadAttack	3.6	2.4	45.8	2.9	29.1	5.8	6.1	2.9	25.3	6.9

Table 1: Comprehensive performance comparison (ASR % and Pert %) across all victim models and datasets under a 100-Query Budget.

in Appendix G, exhibiting trends consistent with those observed here.

PivotAttack's superiority stems from two key factors: **First**, unlike methods like HyGloadAttack or TextHacker that typically start outside from the decision boundary or perform random modifications—often consuming excessive queries to find a valid direction—PivotAttack identifies *pivot words* that dictate the model's prediction and optimizes within the label-invariant region. This inherently reduces query consumption, enabling stronger performance under budget constraints. **Second**, unlike approaches such as LimeAttack that assess token importance indepen-

dently (e.g., selecting top-K words in isolation), PivotAttack explicitly models inter-word interactions, thereby capturing the combinatorial effects of multiple edits. This is crucial for both long-text datasets and robust LLMs, where single-word modifications are often insufficient to flip the prediction.

Detailed Analysis. To provide a comprehensive analysis, we evaluate adversarial example quality, the robustness of PivotAttack under various prompting configurations, and the influence of text length, with detailed results provided in Appendices HJ. Our findings indicate that PivotAttack produces high-fidelity perturbations that preserve original semantics with low grammatical degradation. Furthermore, although implicit defenses such as CoT/Persona prompting and model fine-tuning generally dampen attack performance, PivotAttack remains robust, achieving peak ASR across all scenarios. Regarding text length, PivotAttack consistently surpasses baseline models; this performance gap widens on longer texts, as the increased perturbation budget allows our method to better exploit word-level combinatorial effects, thereby maximizing the impact of the generated adversarial sequences.

4.3 Query Budget (RQ2)

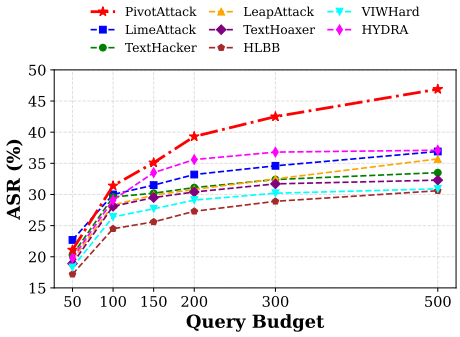


Figure 3: ASR vs. Query Budget: MR (Qwen2.5-FT)

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 MR (Qwen2.5-FT) results are shown in Figure 3, with full results in Appendix K. 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 retention precision estimates and thus identifying a better Pivot Set.

Attack	SNLI		MNLI-m		MNLI-mm	
	ASR.↑	Pert.↓	ASR.↑	Pert.↓	ASR.↑	Pert.↓
PivotAttack	**25.8**	7.9	**47.7**	7.7	**54.8**	8.0
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**
VIWHard	24.1	7.8	45.1	7.6	53.3	7.3
HyGloadAttack	25.5	8.8	46.5	7.7	54.0	7.5

Table 2: Textual Entailment Attack Performance on BERT (100-Query Budget)

Method	ASR.↑	Pert.↓	Query↓	EM↓	F1↓
PivotAttack	**48.3**	12.8	165	**35**	**45.2**
LimeAttack	43.2	13.1	126	40	49.2
TextHacker	42.8	14.2	222	41	49.4
LeapAttack	32.4	19.8	197	39	48.2
TextHoaxer	45.7	14.8	213	43	46.1
HLBB	22.3	14.2	195	48	58.8
VIWHard	21.6	**12.2**	**119**	55	68.3
HyGloadAttack	29.1	13.4	213	44	52.4

Table 3: Experimental results on the SQuAD dataset (QA task). The victim model achieves an original Exact Match (EM) of 58.0% and F1 score of 73.4%. Constraints: Perturbation rate < 0.2, Query budget < 300.

4.4 Transferability (RQ3)

To evaluate the effectiveness of different text adversarial attack algorithms, we conducted experiments on two fundamental NLP tasks: question answering (QA) and textual entailment. For the QA task, we applied our algorithm following the methodology described by Boreshban et al. (2023), utilizing the SQuAD dataset and the identical victim model employed in their study. As detailed in Table 3, PivotAttack outperforms all baseline methods by achieving the highest Attack Success Rate and inducing the most substantial degradation in model performance. Furthermore, we evaluated our approach on the textual entailment task using the SNLI and MNLI datasets. The results, presented in Table 2, demonstrate that PivotAttack attains the highest ASR while preserving perturbation levels comparable to existing methods.

4.5 Ablation Study (RQ4)

Ablation Study of PivotAttack. We conduct an ablation study on Yelp and MR (LSTM, 100 queries) with three variants: (1) randomizing the Pivot Set ($-$*Pivot Set*); (2) fixing the perturbation threshold h at 0.1 ($-$*Dynamic constraints*);

and (3) randomly ranking non-pivot words instead of reusing the intermediate KL-LUCB output (−*Reuse*). Table 4 shows that while perturbation rates remain stable, −*Pivot Set* causes the largest ASR drop (e.g., 16.8%→13.7% on Yelp), confirming the efficacy of pivot targeting. −*Dynamic constraints* also reduces performance, whereas −*Reuse* has minimal impact, 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 constraints	14.4	**1.4**
	–Reuse	16.6	**1.4**
MR	**PivotAttack**	**50.6**	5.1
	–Pivot Set	46	5.2
	–Dynamic constraints	47.2	**4.8**
	–Reuse	50.2	5.2

Table 4: Ablation study of PivotAttack components.

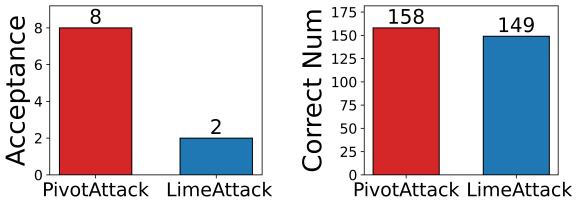
Ablation Study of Pivot Set Identification. We evaluate component contributions under a 300-query budget using three variants: (1) skipping attack culling (−*Culling*), (2) omitting multi-armed bandit refinement (−*MAB*), and (3) disabling retention precision tightening (−*Tighten*). As shown in Table 5, −*MAB* causes the largest performance drop, confirming the necessity of retention precision refinement. Furthermore, −*Culling* reduces efficacy by wasting queries, while −*Tighten* has negligible impact as it is rarely triggered.

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 5: Ablation study of the Pivot Set Identification module.

4.6 Human Evaluation (RQ5)

To assess the interpretability of PivotAttack, we conducted a human study with 10 participants, comparing it against the interpretable baseline, LimeAttack. We examined which algorithms 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 models word-importance rationale. Finally, participants selected which method produced more reasonable results. As shown in Figure 4, PivotAttack achieved slightly higher predictability and was generally judged more reasonable.



(a) Reasonableness (b) Prediction

Figure 4: Human Evaluation

Qualitatively, participants observed that LimeAttack often prioritized functional words (e.g., "of"), whereas PivotAttack targeted semantically meaningful tokens. For instance, in the example "It's hard to resist his enthusiasm" (Figure 8, Appendix L), PivotAttack identifies the Pivot Set {hard, resist}, correctly locating the semantic anchor where modification destroys the positive sentiment. In contrast, LimeAttack highlights trivial tokens such as {even, it}, which may trigger statistical fluctuations in the model but lack genuine semantic significance.

5 Conclusion

We presented PivotAttack, a query-efficient hard-label black-box attack that fundamentally shifts the paradigm from approximating decision boundaries to identifying Pivot Sets—combinatorial tokens that act as the "load-bearing walls" of a 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. Notably, it exposes the vulnerability of Large Language Models (e.g., Qwen2.5, Gemma 3) in both zero-shot and robust fine-tuned settings, achieving high success rates with minimal perturbation.

Limitations

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.

Ethical Considerations

This work presents PivotAttack, a novel adversarial attack framework designed to evaluate and expose the vulnerabilities of NLP models. We acknowledge the dual-use nature of this research: while our primary objective is to advance the community's understanding of model robustness and aid in the development of stronger defensive mechanisms, such attack methods could potentially be exploited by malicious actors to bypass safety guardrails or degrade system performance. To mitigate potential harm, all experiments were conducted exclusively on publicly available academic datasets (e.g., SST-2, SQuAD, SNLI) and open-weight models (e.g., BERT, Qwen2.5, Gemma 3), without targeting live, user-facing commercial applications. By open-sourcing our code, we aim to provide the research community with a transparent tool for red-teaming, ultimately contributing to the creation of safer and more reliable AI systems.

Acknowledgements

This research is supported by the Joint Project of Philosophy and Social Sciences Planning Discipline in Guangdong Province (Grant No.: GD23XZY07), and the Ordinary University Characteristic Innovation Project of Guangdong Province (Grant No.: 2023KTSCX031).

References

Yasaman Boreshban, Seyed Morteza Mirbostani, Seyedeh Fatemeh Ahmadi, Gita Shojaee, Fatemeh Kamani, Gholamreza Ghassem-Sani, and Seyed Abolghasem Mirroshandel. 2023. Robustqa: A framework for adversarial text generation analysis on question answering systems. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023 - System Demonstrations, Singapore, December 6-10, 2023*, pages 274–285. Association for Computational Linguistics.

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].

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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):17351780.

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.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Zhaorong Liu, Xi Xiong, Yuanyuan Li, Yan Yu, Jiazhong Lu, Shuai Zhang, and Fei Xiong. 2024. Hygloadattack: Hard-label black-box textual adversarial attacks via hybrid optimization. *Neural Networks*, 178:106461.

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.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

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 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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

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*.

Gemma Team. 2025. Gemma 3 technical report. *CoRR*, abs/2503.19786.

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, Jinghui Chen, Chenglin Miao, Ting Wang, and Fenglong Ma. 2022a. 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. 2022b. 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.

Zhen Yu, Xiaosen Wang, Wanxiang Che, and Kun He. 2022b. 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.

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.

Hua Zhang, Jiahui Wang, Haoran Gao, Xin Zhang, Huewei Wang, and Wenmin Li. 2025. Viwhard: Text adversarial attacks based on important-word discriminator in the hard-label black-box setting. *Neurocomputing*, 616:128917.

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

Hai Zhu, Qingyang Zhao, Weiwei Shang, Yuren Wu, and Kai Liu. 2024. Limeattack: Local explainable method for textual hard-label adversarial attack. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19759–19767. AAAI Press.

A Problem Formulation

Given a victim classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ and an input X, our goal is to generate an adversarial example X' that misleads the model (i.e., $f(X') \neq f(X)$) while preserving the semantic content of the original input. In the hard-label black-box setting, the attacker queries the victim model to obtain only the predicted label, without access to gradients or confidence scores. Accordingly, the objective is to find such a semantically consistent adversarial example X' within a limited query budget B.

B 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 (Zhu et al., 2024; 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.

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
QA	SQuAD	88K	11K	–	10

Table 6: Dataset Statistics

To evaluate the effectiveness of PivotAttack across a broader range of NLP tasks, we compared its performance against baseline models on textual entailment and QA. For the textual entailment task, we utilized the SNLI (Conneau et al., 2017) and MultiNLI (Williams et al., 2018) datasets, with the latter encompassing both matched (MNLI-m) and mismatched (MNLI-mm) partitions. The experimental data for these entailment datasets were sourced from the public repository accompanying the HLBB implementation. For the QA task, we employed the SQuAD dataset (Rajpurkar et al., 2016), obtaining our experimental data from the public repository associated with Boreshban et al. (2023).

Details of each dataset are provided below:

- **SNLI:** Consists of human-annotated premise-hypothesis pairs categorized into three classes: entailment, contradiction, and neutral.
- **MultiNLI:** A comprehensive NLI dataset providing two distinct evaluation settings: a matched domain (MNLI-m) and a more challenging mismatched domain (MNLI-mm).
- **SQuAD (v1.1):** SQuAD is a reading comprehension dataset comprising numerous question-answer pairs, where the answer to every question is a specific text span extracted from the corresponding Wikipedia article.

The dataset statistics are summarized in Table 6.

C Victim Model Details

We evaluate our attack against a diverse set of victim models spanning three architectural paradigms: traditional deep learning models, including WordCNN (Kim, 2014) and WordLSTM (Hochreiter and Schmidhuber, 1997); pre-trained language models (PLMs) such as

BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), and ALBERT (Lan et al., 2020); and large language models (LLMs), specifically Gemma 3 (Team, 2025) and Qwen 2.5 (Yang et al., 2025). Their specific architectural details are outlined below:

- **WordCNN**: A convolutional neural network utilizing 200-dimensional GloVe embeddings (trained on 6B tokens). It consists of three convolutional kernels (sizes 3, 4, and 5) with 100 filters each, followed by a dropout layer ($p = 0.3$).
- **WordLSTM**: A bidirectional LSTM with 150 hidden units, employing the same embedding initialization and regularization settings as WordCNN.
- **BERT**: For the text adversarial attack and textual entailment tasks, we employ the standard BERT architecture, comprising 12 layers, 768 hidden units, and 12 attention heads. Input sequences are fixed to a length of 256 tokens via padding or truncation. For the question answering task, we adopt the identical BERT model utilized by Boreshban et al. (2023).
- **DistilBERT**: A distilled version of BERT comprising six layers, each with 768 hidden units and 12 attention heads. We utilize the final-layer `[CLS]` token representation for classification, applying dropout for regularization.
- **ALBERT**: A parameter-efficient variant comprising 12 layers (768 hidden units, 12 attention heads) with factorized embeddings (projecting 128-dim to 768-dim) and cross-layer parameter sharing. We use a fixed sequence length of 256 and feed the final `[CLS]` representation into a linear classification head.
- **Gemma 3 Zero-shot (Gemma 3-ZS)**: We evaluate the Gemma 3-1B model in a zero-shot setting. The specific prompts used for classification are illustrated in Figure 5.
- **Qwen2.5 Zero-shot (Qwen2.5-ZS)**: The Qwen2.5-1.5B model evaluated in a zero-shot configuration, using the same prompt structure shown in Figure 5.

Model	Yelp	Yahoo	MR	Amazon	SST-2
CNN	93.6	71.1	76.5	91.1	86.4
LSTM	94.6	73.7	78.0	92.2	87.8
BERT	96.5	79.1	85.0	94.2	98.6
DistilBERT	96.3	78.9	97.6	96.1	98.6
ALBERT	93.2	78.3	93.0	93.2	95.3
Gemma 3-ZS	81.7	19.6	71.5	79.7	61.0
Qwen2.5-ZS	89.0	23.7	79.7	88.5	85.3
Qwen2.5-FT	98.3	64.0	97.0	97.1	95.8

(a) Text Classification Datasets

Model	SNLI	MNLI-m	MNLI-mm
BERT	89.1	85.1	82.2

(b) Natural Language Inference Datasets

Table 7: Original Accuracy of the Victim Models on Classification and NLI Tasks

- **Qwen2.5 Fine-tuned (Qwen2.5-FT)**: Task-specific variants of Qwen2.5-1.5B obtained via QLoRA fine-tuning. The base model parameters are frozen, and LoRA adapters ($r = 16, \alpha = 32,$ dropout $= 0.05$) are optimized on the query, key, value, and output projections. Classification is cast as a generation task outputting a single numeric label via instruction-style prompts (see Figure 5), with cross-entropy loss applied exclusively to the target token.

The original accuracy of the victim models on various classification and entailment datasets is reported in Table 7.[2]

D Baseline Details

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

- **HLBB (Maheshwary et al., 2021b)**: A population-based genetic algorithm that optimizes adversarial examples by iteratively selecting candidates with high semantic similarity and low perturbation under a strict query budget.
- **TextHoaxer (Ye et al., 2022b)**: A greedy heuristic that prioritizes token positions, using a few probing queries to determine whether a substitution is worthwhile before committing.

[2]Large language models (e.g., Qwen and Gemma) may refuse to generate responses for certain inputs. Consequently, instances triggering such refusals were excluded from the evaluation for these models.

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.

- **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.

- **TextHacker (Yu et al., 2022b)**: Maintains an online-updated word-importance table derived from edit-flip history, which guides a hybrid local search.

- **LimeAttack (Zhu et al., 2024)**: Leverages a local explainable method to approximate word importance ranking, and then adopts beam search to find the optimal solution.

- **VIWHard (Zhang et al., 2025)**: Introduces an important-word discriminator trained on a local surrogate model to identify vulnerable tokens without querying the target. It generates context-aware substitutions via a Masked Language Model (MLM) and optimizes the attack using a genetic algorithm.

- **HyGloadAttack (Liu et al., 2024)**: Integrates gradient-based search in the embedding space with exchange-based mechanisms via a hybrid optimization strategy. It employs global random initialization to escape local optima and accelerate the search process.

E Evaluation Metrics

Following established practices in prior work (e.g., LimeAttack, LeapAttack), we evaluate our method using Attack Success Rate (ASR), Perturbation Rate (Pert), and Semantic Similarity (Sim). ASR quantifies the attack's effectiveness as the proportion of successfully misclassified examples. To evaluate the stealthiness and quality of the adversarial examples, Pert measures the modification magnitude (the fraction of altered tokens), while Sim captures the degree of semantic preservation relative to the original input. For the question answering task, we additionally report Exact Match (EM) and token-level F1 scores. EM demonstrates the percentage of those responses that exactly match the ground truth answer, whereas F1 represents the average overlap between the ground truth and the prediction.

F Implementation Details

We set the query budget to $B = 100$, the KL-LUCB quota parameter to $\gamma = 0.8$, the retention 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.

G Additional Experimental Results on WordCNN and ALBERT

To verify generalization, we evaluate PivotAttack on WordCNN and ALBERT (Table 8). Consistent with main results, PivotAttack maintains dominance across diverse architectures. On WordCNN (Yelp), PivotAttack matches the top baseline's ASR (12.9%) but with nearly half the perturbation (1.6% vs. 3.0%). On ALBERT (SST-2), PivotAttack leads with 51.1% ASR compared to the runner-up's 47.0% (LimeAttack). These results confirm that PivotAttack's superiority is model-agnostic, effective against both convolutional networks and lightweight Transformers.

H Adversary Quality

To balance effectiveness and stealth, adversarial samples must preserve semantic similarity with the original text. We further evaluate attacks on the

Model	Attack	Yelp		Yahoo		MR		Amazon		SST-2	
		ASR↑	Pert↓	ASR↑	Pert↓	ASR↑	Pert↓	ASR↑	Pert↓	ASR↑	Pert↓
WordCNN	**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
	VIWHard	10.7	1.7	42.5	2.6	47.5	5.2	14.7	2.3	39.7	6.1
	HyGloadAttack	11.3	2.7	40.9	3.5	47.5	5.5	18.9	2.4	43.4	6.2
ALBERT	**PivotAttack**	**11.7**	**1.0**	**41.0**	**1.7**	**39.4**	6.1	**15.4**	**1.3**	**51.1**	6.2
	LimeAttack	11.4	2.4	40.3	2.5	38.6	**5.1**	15.0	2.0	47.0	5.9
	TextHacker	11.0	2.5	38.3	3.2	30.5	7.0	14.5	2.5	35.5	7.0
	LeapAttack	11.1	2.0	37.3	3.4	35.2	5.6	15.1	2.4	42.8	6.4
	TextHoaxer	11.0	2.2	34.5	3.4	30.1	5.3	14.7	2.3	34.8	6.5
	HLBB	11.5	2.8	34.2	3.8	30.1	6.0	15.3	2.7	34.1	6.5
	VIWHard	11.5	1.5	40.5	1.7	32.0	5.2	14.9	1.4	45.2	6.3
	HyGloadAttack	11.6	1.8	36.9	3.2	36.6	5.5	15.2	2.3	38.1	6.6

Table 8: Comprehensive performance comparison (ASR % and Pert %) across all victim models and datasets under a 100-Query Budget.

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[3]. As shown in Table 9, PivotAttack maintains superior performance across these metrics while achieving the highest attack success.

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
VIWHard	8.0	1.2	99.3	**0.24**
HyGloadAttack	**9.7**	2	99.4	0.44

Table 9: Adversarial Sample Quality Comparison on BERT (Yelp, 100-Query Budget)

I Robustness under Different Prompting Settings

To further investigate the vulnerability of LLMs under more complex operational contexts, we evaluate the effectiveness of our attack strategy across different prompting settings. As demonstrated in Table 10, although advanced prompting strategies (e.g., CoT, Persona) and model fine-tuning act as effective implicit defenses that generally degrade overall attack efficacy, PivotAttack consistently achieves the highest ASR across all evaluated configurations.

J Impact of Text Length

We conducted a breakdown analysis by sampling 100 instances for each length interval from the dataset. The results are shown in Figure 6. As shown in Figure 6, PivotAttack consistently maintains the highest ASR and lowest perturbation rate across all length intervals.

K 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 7. As shown, PivotAttack consistently surpasses all baseline methods across different datasets and victim models.

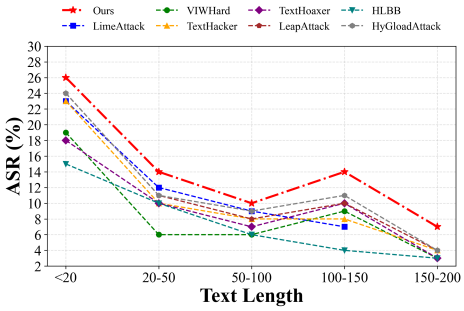
L Human Evaluation Details

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 8a, 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, participants were required to complete 20 multiple-choice questions. Each question pro-

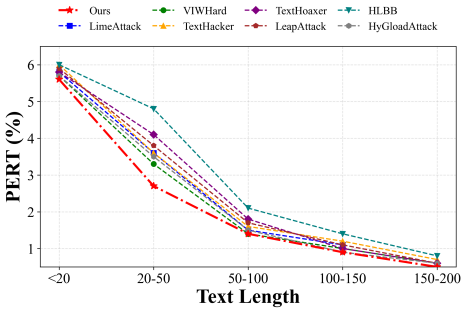
[3]https://languagetool.org

Method	ZS-Std	ZS-CoT	ZS-Identity	FT-Std	FT-CoT	FT-Identity
	ASR / Pert	ASR / Pert	ASR / Pert	ASR / Pert	ASR / Pert	ASR / Pert
PivotAttack	**53.2** / 6.1	**46.8** / 6.2	**45.1** / 6.2	**30.3** / 6.4	**25.7** / **5.9**	**24.2** / 6.2
LimeAttack	49.6 / **5.5**	44.3 / 6.0	41.2 / 6.2	26.1 / **6.1**	21.8 / 6.0	18.4 / 6.4
TextHacker	45.2 / 6.2	40.9 / 6.3	38.4 / 6.5	25.8 / 6.2	20.3 / **5.9**	16.2 / 6.3
LeapAttack	40.8 / 5.7	36.2 / 6.2	31.3 / **6.1**	24.9 / 6.3	19.2 / 6.2	18.1 / **6.0**
TextHoaxer	43.1 / 6.0	30.4 / 5.9	33.2 / 6.2	25.2 / 6.7	18.1 / 7.1	15.3 / 7.0
HLBB	37.3 / 5.9	30.8 / **5.8**	29.1 / 6.5	27.2 / 6.3	16.1 / 7.2	14.2 / 7.1
VIWHard	48.1 / 6.6	24.2 / 6.4	39.3 / 6.3	24.8 / 6.4	18.2 / 6.0	17.1 / 6.3
HyGloadAttack	46.4 / 6.7	41.7 / 6.7	37.2 / 6.6	25.3 / 6.9	22.6 / 7.2	20.1 / 7.0

Table 10: Robustness evaluation across different prompt templates on SST-2. We compare Standard (Std), Chain-of-Thought (CoT), and Identity-based prompts under both Zero-Shot (ZS) and Fine-Tuned (FT) settings.



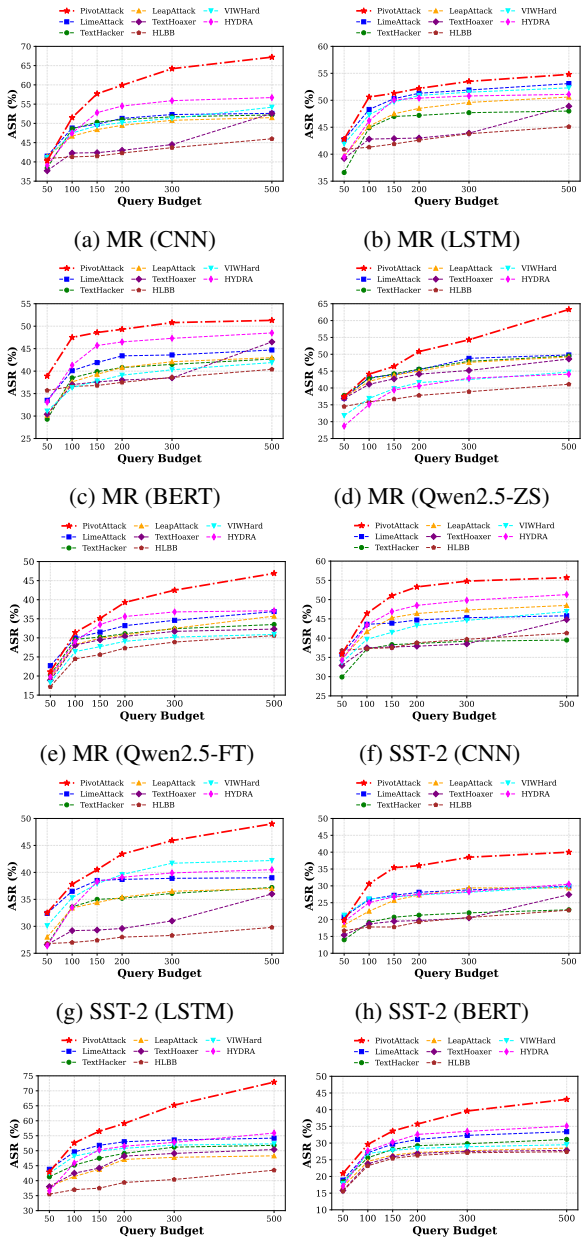
(a) ASR



(b) Perturbation rate

Figure 6: Impact of Text Length

vided 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-ZS)

(e) MR (Qwen2.5-FT) (f) SST-2 (CNN)

(g) SST-2 (LSTM) (h) SST-2 (BERT)

(i) SST-2 (Qwen2.5-ZS) (j) SST-2 (Qwen2.5-FT)

Figure 7: ASR of different models under different query budgets on MR and SST-2.

it 's hard to resist his enthusiasm , even if the filmmakers come up with nothing original in the way of slapstick sequences

`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 , even if the filmmakers come up with nothing original in the way of slapstick sequences

`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 8: Example of the Human Evaluation Survey